6

Scalable Test Problems for Evolutionary Multiobjective Optimization

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns and Eckart Zitzler

Summary. After adequately demonstrating the ability to solve different two-objective optimization problems, multiobjective evolutionary algorithms (MOEAs) must demonstrate their efficacy in handling problems having more than two objectives. In this study, 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 the shape and the 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.

6.1 Introduction

Most earlier studies on multi-objective evolutionary algorithms (MOEAs) introduced test problems which were either too simple or not scalable in terms of number of objectives and decision variables. Some test problems were too complicated to visualize the exact shape and location of the resulting Pareto-optimal front. Schaffer's [1] study introduced two single-variable test problems (SCH1 and SCH2), which have been widely used as test problems. Kursawe's test problem [2], 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 test problem [3], FON. Poloni et al.'s test problem [4], POL used only two decision variables. Although the mathematical formulation of the problem is non-linear, the resulting Pareto-optimal front corresponds to an almost linear relationship among decision variables. Viennet's test problem [5], 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 [6, 7].

However, in 1999, the first author, for the first time, introduced a systematic procedure of designing two-objective test problems which are simple to construct and are scalable to the number of decision variables [8]. In these problems, the exact shape and location of the Pareto-optimal front 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. The construction procedure adopted in that study is not the only alternative for the test problem design and certainly many other principles are possible. In the absence of any other systematic construction procedure, those test problems have been used by many researchers since then. However, they have also been somewhat criticized for the relative independence feature of the functionals in achieving both the tasks. Such critics have overlooked an important aspect of that study. The non-overlapping property of the two key functionals in the test problems was introduced for ease of the construction procedure. That study also suggested the use of a procedure to map the original variable vector (say $\mathbf{y}$) on which an MOEA works to a different decision variable vector (say $\mathbf{x}$) with a transformation matrix: $\mathbf{x} = \mathcal{M}\mathbf{y}$. This way, although test problems are constructed for two non-overlapping sets from $\mathbf{x}$, each dependent variable x_i 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 et al. [9] showed that six test problems designed based on an uncorrelated version of Deb's construction procedure were 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 by including three basic operators: (1) an elite-preserving operator, (2) a niche-preserving operator, and (3) a non-domination based selection operator. 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 a larger number of objectives. Besides testing an MOEA's ability to solve problems with a large number of objectives, the proposed test problems may also be used for systematically comparing two or more MOEAs. Since one such test problem can be used to test a particular aspect of multiobjective 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 study, we first describe the desired features needed in a test problem and then suggest three approaches for systematically

designing test problems for multiobjective optimization algorithms. Although most problems are illustrated for three objectives (for ease of illustration), the test problems are generic and scalable to an arbitrary number of objectives. Finally, we suggest a set of nine test problems based on the suggested construction procedures and show the difficulties faced by two MOEAs (NSGA-II and SPEA2) in solving these problems.

A short version of this study appeared elsewhere [10],but this study describes different test problem design procedures in a more systematic manner and presents more simulation results. Hopefully, the techniques suggested in this study would be useful in designing further test problems for multiobjective optimization.

6.2 Desired Features of Test Problems

The ultimate goal of developing any optimization algorithm is to solve real-world optimization problems reliably and efficiently. However, since in real-world optimization problems the nature of landscape and optimum solution(s) are not usually known beforehand, at the end of a simulation run, it becomes difficult to test how well an algorithm has performed. For this purpose, there is a need to develop test problems for testing optimization algorithms. Since the landscape and corresponding optimum solution(s) of such problems will be known, they allow to test an algorithm's ability to overcome the difficulties posed by the landscape and ability to converge near the optimum solution(s). Keeping in mind the ultimate goal of solving real-world problems, it then becomes important to construct test problems which are representative to real-world problems. However, since real-world problems can be very different from each other and since their landscapes may not be known a priori, it is essential to design a test suite, instead of a single test problem, for the task. This way, various complexities which may be present in real-world problems may be introduced systematically in a number of test problems.

Besides handling different landscape complexities, algorithms should be scalable to the problem size. An algorithm which is efficient in solving a problem with a few decision variables may not work well (or may work with exponentially more computations) in higher problem sizes. Therefore, it is desirable that a test suite, containing test problems each of which is capable of testing an algorithm's ability to handle different aspects of landscape complexity, is also scalable to the problem size. This way, a test problem not only allows to study a particular landscape complexity, but also allows to test how that landscape complexity scales with increased problem sizes. Based on this discussion, we suggest the following features that a test problem suite should have for adequately testing an MOEA:

1. 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. This is because convergence near to the Pareto-optimal front and maintenance of a diverse set of solutions are two basic goals in multiobjective optimization.
2. Test problems may be scalable to have any number of decision variables. This is because many real-world problems usually involve a large number of decision variables. For all practical purposes, test problems involving a few hundred variables may be included in the test suite.
3. Test problems should be scalable to have any number of objectives. Although in most real-world problems the number of objectives can be reduced to four or five, test problems involving as large as 15 to 20 objectives may be included in the test suite.
4. Test problems may be simple to construct. This may not be an essential matter for constructing test problems, but if desired features can be incorporated in test problems with a simple construction procedure, it is always desirable.
5. The resulting Pareto-optimal front (continuous or discrete) may be easy to comprehend, and its shape and location may be exactly known. The corresponding decision variable values may also be easy to find.
6. To make the test problems useful in practice, they should exhibit difficulties similar to those present in most real-world problems.

It is important to mention that the fifth feature described above may not be of much importance for comparing two or more MOEAs. But for evaluating an MOEA's performance in convergence and maintenance of diversity, knowledge of the exact Pareto-optimal set is essential.

Along with the test problems designed by keeping the above features in mind, it does not do any harm to keep a few additional real-world problems in the test suite. Such problems can be treated as a black-box with a clearly defined set of input decision variables (and their possible ranges) and corresponding objective values and constraints (if any). Many such real-world problems [6, 11] have been already solved using MOEAs and some representative of them can be included in the test suite. A network design application problem along with a number of MOEAs already exist on the PISA [12] web page (`http://www.tik.ee.ethz.ch/pisa`) for this purpose.

The popularity of two-objective test problems suggested earlier [8] in many research studies is partly because of the ease of constructing the test problems and the ease of illustrating the obtained non-dominated solutions against the true 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 as close to the Pareto-optimal front and finding as diverse a set of solutions on the entire Pareto-optimal front as possible. An MOEA therefore, should be tested for each of the two tasks. Thus, some test problems should test an MOEA's ability to negotiate artificial hurdles which hinder MOEA's 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. 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 non-convex, discrete, and having variable density of solutions along the front. Although these features of test problems were also suggested for two-objective problems earlier [8], the technique can be extended to more than two objectives. In any case, the increased dimensionality associated with a large number of objectives may cause added difficulties to MOEAs.

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

6.3 Different Methods of Test Problem Design

Although the main focus of the above discussion is based on solving real-world problems, there is a need to develop simple-to-understand scalable test problems for theoretical studies, such as studies for analyzing the running time complexity of MOEAs. Keeping in mind all the above issues which may be present in a test problem suite, a number of different construction procedures can be adopted for multiobjective 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 multiobjective 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 [1]:

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

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 members of the Pareto-optimal set. Similarly, the problem FON shown below

$$\begin{cases} \text{Minimize } f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2\right), \\ \text{Minimize } f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2\right), \\ \qquad -4 \le x_i \le 4 \quad i = 1, 2, \ldots, n. \end{cases} \tag{6.1}$$

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 [7] lists a number of other test problems, which follow this construction principle. It is interesting to note that such a construction procedure can be extended to problems having more than two objectives as well [13]. In a systematic procedure, each optimum may be assumed to lie on each of M (usually $< n$) coordinate directions. The main advantage of this procedure is its simplicity and ease of construction. 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 [1]), 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. For running time complexity analysis, a two-objective leading-ones-trailing-zeros (LOTZ) problem was designed recently [14].

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 to describing them.

6.4 Bottom-up Approach

In this approach, a mathematical function describing the Pareto-optimal front is assumed in the objective space and an overall objective space is constructed

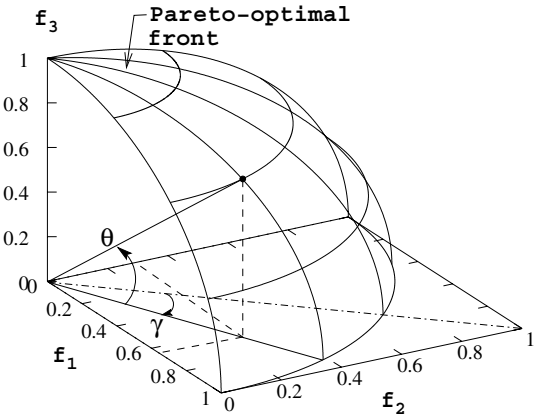

Figure 6.1. First quadrant of a unit sphere as a Pareto-optimal front.

from this front to define the test problem. For two objectives, one such construction was briefly suggested earlier [8] and was extended for higher objectives elsewhere [6]. Here we outline the generic procedure through a three-objective example problem:

Step 1: Choose a Pareto-optimal front. The first task in the bottom-up approach is to choose the Pareto-optimal front. For M objectives, this means designing a parametric surface involving $(M-1)$ variables. For illustration purposes, let us assume that we would like to have a Pareto-optimal front in a three-objective problem, where all objective functions take non-negative values and the desired front is the surface of an octant of a sphere of radius one (as shown in Figure 6.1). With the help of spherical coordinates (we call them *parametric* variables) θ, γ, and r $(= 1$ here), the front can be described as follows:

$$\left.\begin{array}{l} f_1(\theta,\gamma) = \cos\theta\cos(\gamma+\pi/4), \\ f_2(\theta,\gamma) = \cos\theta\sin(\gamma+\pi/4), \\ f_3(\theta,\gamma) = \sin\theta, \\ \quad \text{where} \quad 0 \le \theta \le \pi/2, \\ \qquad\qquad -\pi/4 \le \gamma \le \pi/4. \end{array}\right\} \tag{6.2}$$

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.

Step 2: Build the objective space. Using the chosen Pareto-optimal surface, we construct the complete objective space. A simple way to construct the rest of the objective space is to construct an extreme boundary surface parallel to the Pareto-optimal surface, so that the hyper-volume bounded by these two surfaces constitute the entire objective space. This can be achieved by introducing an additional variable in the parametric

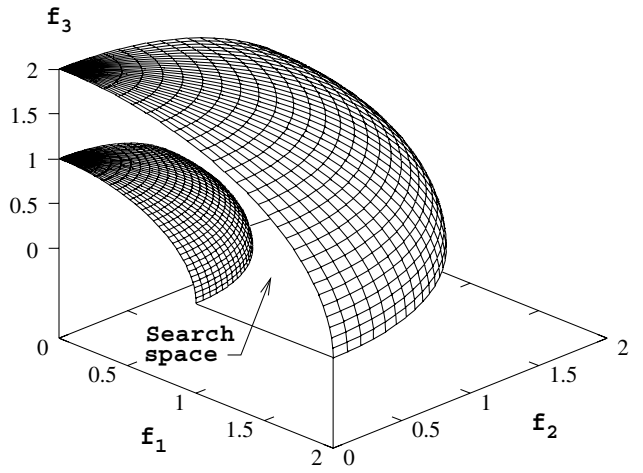

Figure 6.2. Overall search space is bounded by the two spheres.

equations. For the example problem, this can be achieved by multiplying the above three functions with a radius term, which takes a value greater than or equal to one. Different values of the 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:

$$\left.\begin{array}{l} \text{Minimize } f_1(\theta,\gamma,r) = (1+g(r))\cos\theta\cos(\gamma+\pi/4), \\ \text{Minimize } f_2(\theta,\gamma,r) = (1+g(r))\cos\theta\sin(\gamma+\pi/4), \\ \text{Minimize } f_3(\theta,\gamma,r) = (1+g(r))\sin\theta, \\ \qquad 0 \le \theta \le \pi/2, \\ \qquad -\pi/4 \le \gamma \le \pi/4, \\ \qquad g(r) \ge 0. \end{array}\right\} \tag{6.3}$$

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

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

Figure 6.2 shows the overall objective space with any function for $g(r)$ with $0 \le g(r) \le 1$. We shall discuss more about different $g(r)$ functions a little later. The above construction procedure illustrates how easily a multiobjective test problem can be constructed from an initial choice of a Pareto-optimal surface.

Step 3: Construct the decision space. The above construction of the objective space requires exactly M variables, describing an objective vector anywhere in the objective space. The final task is to map each decision variable vector to an objective vector. This mapping provides an additional flexibility in constructing a test problem. For this mapping,

any number of decision variables (n) can be chosen. If $n < M$, not all parametric variables are independent, thereby causing a reduction in dimensionality of the Pareto-optimal surface. For the three-objective example problem, the three parametric variables used earlier (θ, γ, and r) can each be considered as a function of n decision variables of the underlying problem:

$$\theta = \theta(x_1, x_2, \ldots, x_n), \tag{6.4}$$

$$\gamma = \gamma(x_1, x_2, \ldots, x_n), \tag{6.5}$$

$$r = r(x_1, x_2, \ldots, x_n). \tag{6.6}$$

The functions must be so chosen that they satisfy the lower and upper bounds of θ, γ and $g(r)$ mentioned in Equation 6.3.

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.

6.4.1 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 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 multiobjective optimizer can get stuck to.

Moreover, even using a unimodal g function, variable density of solutions can be introduced in the search space. For example, for the three-objective example problem, if $g(r) = r^{10}$ is used, denser solutions exist away from the Pareto-optimal front. Figure 6.3 shows 15,000 solutions, which are randomly created in the decision variable space of the above problem. On the objective space, they are shown to be biased away from the Pareto-optimal front. For such a biased search space away from the Pareto-optimal front, multiobjective optimizers may have difficulties in converging quickly to the desired front.

6.4.2 Difficulties Across the Pareto-optimal Front

By using a non-linear mapping between parametric and decision variables, some portion of the search space can be made to have more dense solutions than the rest of the search space. For example, in the three-objective problem a variable density of solutions can be created on the Pareto-optimal front by choosing non-linear expressions for the parametric variables θ and γ in Equations 6.4 to 6.5. For example, Figures 6.4 and 6.5 show the problem stated in Equation 6.3 with

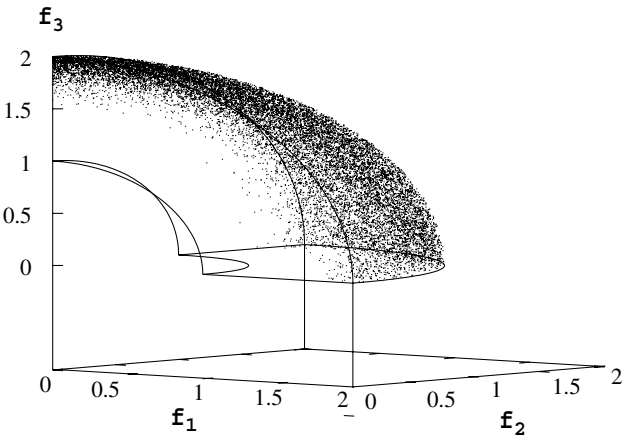

Figure 6.3. The effect of a non-linear g function.

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

respectively. In both cases, $g(r) = r = x_3$ is chosen. In order to satisfy the bounds in Equation 6.3, we have chosen $0 \leq x_1, x_2, x_3 \leq 1$ 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 parametric 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 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.

6.4.3 Generic Sphere Problem

The following is a generic problem to that described in Equation 6.3, having M objectives.

$$\left.\begin{array}{ll}
\text{Minimize } f_1(\boldsymbol{\theta}, \mathbf{r}) = (1 + g(\mathbf{r})) \cos\theta_1 \cos\theta_2 \cdots \cos\theta_{M-2} \cos\theta_{M-1}, \\
\text{Minimize } f_2(\boldsymbol{\theta}, \mathbf{r}) = (1 + g(\mathbf{r})) \cos\theta_1 \cos\theta_2 \cdots \cos\theta_{M-2} \sin\theta_{M-1}, \\
\text{Minimize } f_3(\boldsymbol{\theta}, \mathbf{r}) = (1 + g(\mathbf{r})) \cos\theta_1 \cos\theta_2 \cdots \sin\theta_{M-2}, \\
\vdots \qquad\qquad \vdots \\
\text{Minimize } f_{M-1}(\boldsymbol{\theta}, \mathbf{r}) = (1 + g(\mathbf{r})) \cos\theta_1 \sin\theta_2, \\
\text{Minimize } f_M(\boldsymbol{\theta}, \mathbf{r}) = (1 + g(\mathbf{r})) \sin\theta_1, \\
\qquad 0 \leq \theta_i \leq \pi/2, \quad \text{for } i = 1, 2, \ldots, (M-1), \\
\qquad g(\mathbf{r}) \geq 0.
\end{array}\right\} \quad (6.7)$$

[1] In three-objective knapsack problems, such a biased search space is observed elsewhere [15].

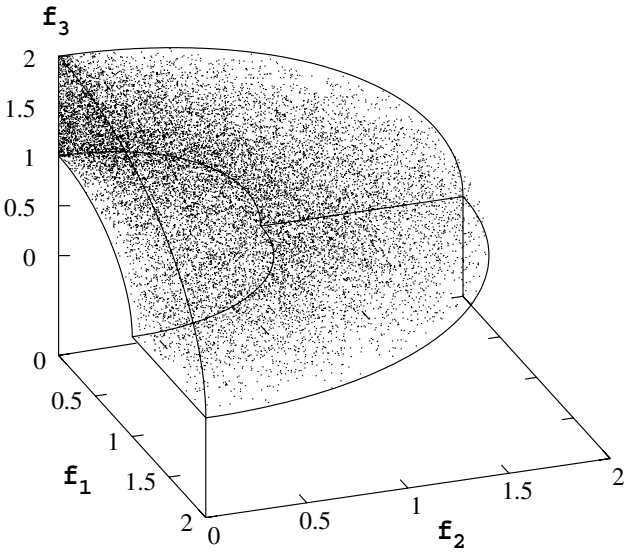

Figure 6.4. Linear mapping (15,000 points).

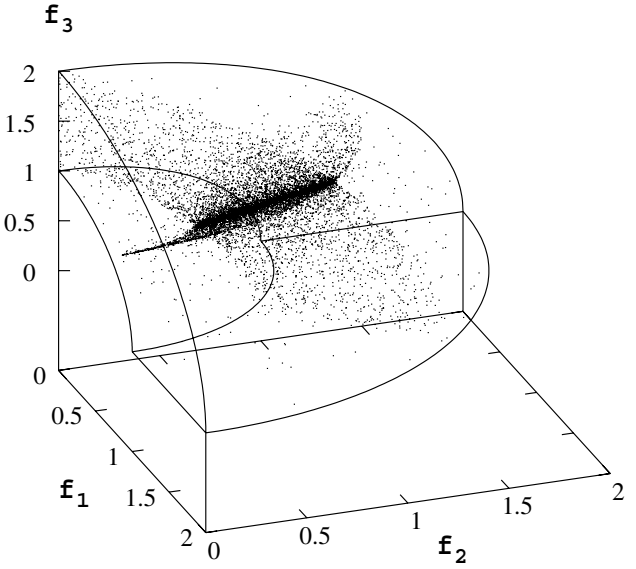

Figure 6.5. Non-linear mapping introduces bias in the search space (15,000 points).

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

$$\theta_i = \frac{\pi}{2} x_i, \quad \text{for } i = 1, 2, \ldots, (M-1). \tag{6.8}$$

The above mapping and the condition on θ_i in Equation 6.7 restrict each of the above x_i to lie within $[0, 1]$. The rest $(n - M + 1)$ of the decision variables are defined as the $\mathbf{r}$ vector (or $r_i = x_{M+i-1}$ for $i = 1, 2, \ldots, (n - M + 1)$) and a suitable $g(\mathbf{r})$ is chosen. It is clear that the above generic sphere function is defined for $n \geq M$.

The Pareto-optimal surface always occurs for the minimum of $g(\mathbf{r})$ function. For example, if the function $g(\mathbf{r}) = \|\mathbf{r}\|^2$ with $r_i \in [-1, 1]$ is chosen, the Pareto-optimal surface corresponds to $r_i = 0$ and the optimal function values must satisfy the following condition:

$$\sum_{i=1}^{M} (f_i^*)^2 = 1. \tag{6.9}$$

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

6.4.4 Curve Problem

Instead of having a complete M-dimensional surface as the Pareto-optimal surface, a lower-dimensional surface can also be chosen as the Pareto-optimal front to the above problem. We realize that in this case not all θ_i variables will be independent to each other. For example, in the case of an M-dimensional curve as the Pareto-optimal front, 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:

$$\theta_i = \frac{\pi}{4(1 + g(\mathbf{r}))} \left(1 + 2g(\mathbf{r})x_i\right), \quad \text{for } i = 2, 3, \ldots, (M - 1), \tag{6.10}$$

The above mapping ensures that the curve is the only non-dominated region in the entire search space. Since $g(\mathbf{r}) = 0$ 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 6.6 shows a sketch of the objective space and the resulting Pareto-optimal curve for a three-objective version of the above problem. One drawback with this formulation is that 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 $g(\mathbf{r})$ function with a higher density of solutions away from $g(\mathbf{r}) = 0$ (such as $g(\mathbf{r}) = 1/\|\mathbf{r}\|^{\alpha}$, where $\alpha \gg 1$) can be used. Using a multi-modal $g(\mathbf{r})$ 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 all the objectives are maximized in the above problem, the top surface becomes the desired Pareto-optimal front. Since there exist less dense

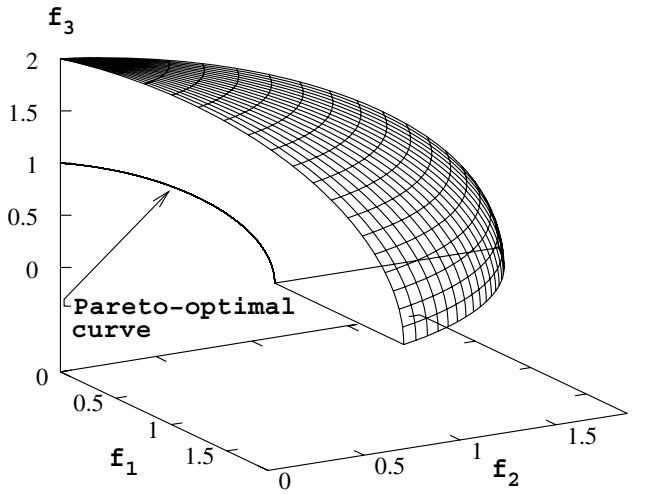

Figure 6.6. The search space and the Pareto-optimal curve.

solutions on this surface, this problem may be a difficult maximization test problem.

Degeneration of the Pareto-optimal front to a low-dimensional surface (such as a curve) is not a mere fantasy. This can happen in problems where some objectives are likely to be non-conflicting to each other. For example, in a recent gearbox design problem having three objectives of minimizing overall volume of the gearbox, maximizing power delivered, and minimizing distance between input-output shafts, a three-dimensional curve appeared as the obtained non-dominated front [16]. This is because of the fact that minimization of the first and the third objectives are not intuitively contradictory to each other.

6.4.5 Test Problem Generator

The earlier study [6] suggested a generic multiobjective 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:

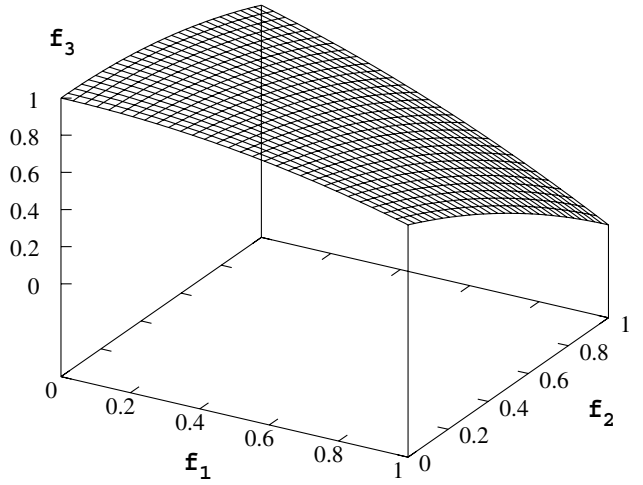

Figure 6.7. A non-convex Pareto-optimal front.

$$\left.\begin{array}{l}\text{Minimize } f_1(\mathbf{x}_1),\\ \text{Minimize } f_2(\mathbf{x}_2),\\ \quad\vdots\\ \text{Minimize } f_{M-1}(\mathbf{x}_{M-1}),\\ \text{Minimize } f_M(\mathbf{x}) = g(\mathbf{x}_M)h\left(f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \ldots, f_{M-1}(\mathbf{x}_{M-1}), g(\mathbf{x}_M)\right),\\ \text{subject to } \mathbf{x}_i \in \mathbb{R}^{|\mathbf{x}_i|}, \quad \text{for } i = 1, 2, \ldots, M.\end{array}\right\} \tag{6.11}$$

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

$$f_M = g^* h(f_1, f_2, \ldots, f_{M-1}). \tag{6.12}$$

Since g^* is a constant, 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:

$$h(f_1, f_2, \ldots, f_{M-1}) = 1 - \left(\frac{\sum_{i=1}^{M-1} f_i}{\beta}\right)^{\alpha}, \tag{6.13}$$

where $\alpha > 1$. Figure 6.7 shows a non-convex Pareto-optimal front with $\alpha = 2$ for $M = 3$ and $\beta = 0.5$.

A disjoint set of Pareto-optimal fronts can be constructed by simply choosing a multi-modal h function as done in the case of two-objective test problem design [8]. Figure 6.8 illustrates a disconnected set of Pareto-optimal

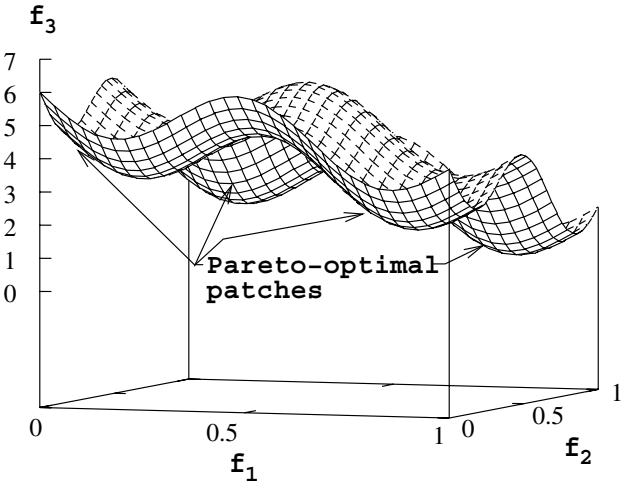

Figure 6.8. A disjoint set of Pareto-optimal regions.

surfaces (for three objectives), which can be generated from the following generic h function:

$$h(f_1, f_2, \ldots, f_{M-1}) = 2M - \sum_{i=1}^{M-1} \left(2f_i + \sin(3\pi f_i)\right). \tag{6.14}$$

Once the h function is chosen, a g function can be chosen to construct the entire objective space. It is important to note that the g function is defined over a set of decision variables. Recall also 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 of solutions in the objective 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 [8] can also be used here.

The recently suggested two-objective counting-ones-counting-zeros problem (COCZ problem) is a test problem developed using the bottom-up approach for running time complexity analysis [17].

6.4.6 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 appropriate functions.

Since the Pareto-optimal front must be expressed mathematically, some complicated Pareto-optimal fronts can be difficult to write mathematically. Like the two-objective problems suggested in Deb [8], these problems may also be criticized because of their variable-wise decomposable nature, but we would like to emphasize that this feature is primarily used to facilitate the introduction of different problem difficulties in a simple manner. As before, the problems can be made more complex by using the variable mapping ($\mathbf{x} = \mathcal{M}\mathbf{y}$) discussed in Section 6.1. Nevertheless, the ability to control different features of the problem is the main strength of this approach.

6.5 Constraint Surface Approach

Unlike starting from a pre-defined Pareto-optimal surface in the bottom-up approach, the constraint surface approach begins by predefining the overall search space. In the following, we describe the construction procedure.

Step 1: Choose a basic objective space. First, a simple M-dimensional bounded region, such as a rectangular hyper-box or an M-dimensional hyper-sphere, is assumed. We illustrate the construction principle here with a hyper-box. Each objective function value is restricted to lie within a predefined lower and a upper bound. The resulting problem is as follows:

$$\left.\begin{array}{ll} \text{Minimize} & f_1(\mathbf{x}), \\ \text{Minimize} & f_2(\mathbf{x}), \\ \vdots & \vdots \\ \text{Minimize} & f_M(\mathbf{x}), \\ \text{Subject to} & f_i^{(L)} \leq f_i(\mathbf{x}) \leq f_i^{(U)} \quad \text{for } i = 1, 2, \ldots, M. \end{array}\right\} \tag{6.15}$$

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 6.9 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.

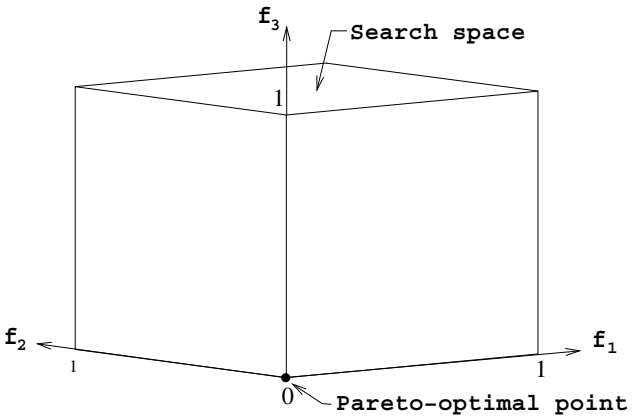

Figure 6.9. Entire cube is the search space. The origin is the sole Pareto-optimal solution.

Step 2: Eliminate part of the objective space. Next, a number of constraints (linear or non-linear) involving the objective function values are added:

$$g_j(f_1, f_2, \ldots, f_M) \geq 0, \quad \text{for } j = 1, 2, \ldots, J. \tag{6.16}$$

This is done to chop off portions of the original bounded region systematically. Figure 6.10 shows the resulting feasible region after adding the following two linear constraints on the originally chosen three-objective rectangular box:

$$g_1 \equiv f_1 + f_3 - 0.5 \geq 0,$$
$$g_2 \equiv f_1 + f_2 + f_3 - 0.8 \geq 0.$$

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 6.10 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.

Step 3: Map decision variable space to objective space. The final task is to map each decision variable vector to the objective space. This can be achieved by choosing each objective function f_i as a linear or non-linear function of n decision variables.

Interestingly, there exist two-variable and three-variable constrained test problems TNK [18] and problems [19] in the literature using the above concept. In these problems, 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 space. As an illustration of further difficulties through non-linear

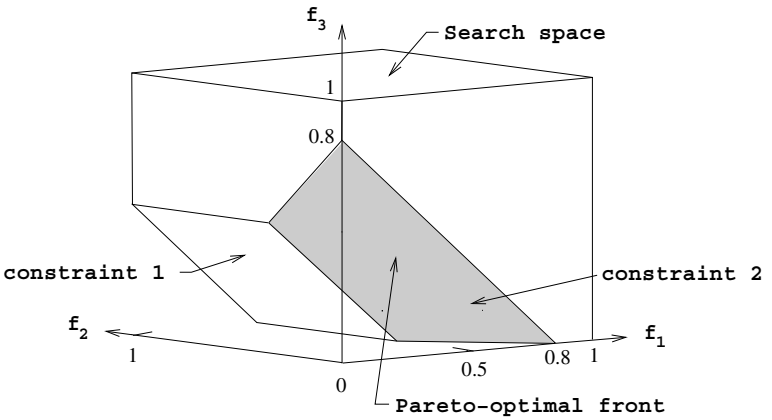

Figure 6.10. Two constraints are added to eliminate a portion of the cube, thereby resulting in a more interesting Pareto-optimal front.

f_i, we construct the following three-objective problem with a bias in the search space:

$$\left.\begin{array}{ll} \text{Minimize} & f_1(x_1) = 1 + (x_1 - 1)^5, \\ \text{Minimize} & f_2(x_2) = x_2, \\ \text{Minimize} & f_3(x_3) = x_3, \\ \text{Subject to} & g_1 \equiv f_3^2 + f_1^2 - 0.5 \geq 0, \\ & g_2 \equiv f_3^2 + f_2^2 - 0.5 \geq 0, \\ & 0 \leq x_1 \leq 2, \\ & 0 \leq x_2, x_3 \leq 1. \end{array}\right\} \tag{6.17}$$

Figure 6.11 shows 25,000 feasible solutions randomly generated in the decision variable space. The 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 the 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.

6.5.1 Advantages and Disadvantages

The construction process here is simpler compared to the bottom-up approach. Simple geometric constraints can be used to construct the feasible search

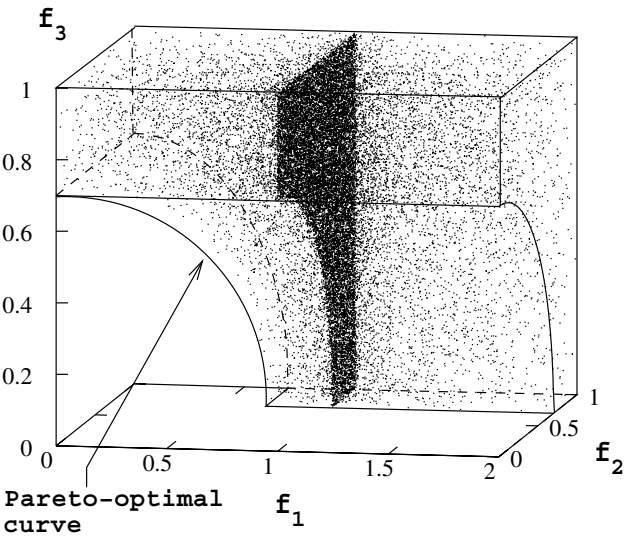

Figure 6.11. Non-linearity in functionals produces non-uniform density of solutions.

space. Using this procedure, different shapes (convex, non-convex, or discrete) of the Pareto-optimal region can be generated. Using a systematic choice of constraint surfaces, scalable test problems can be constructed. 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.

6.6 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 [20], SPEA [15], NSGA-II [21] 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 need 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, calculation of a metric to compute distances from all neighbors gets computationally expensive because of the added dimensionality. Although a widely distributed set of solutions can be found, as using NSGA-II or SPEA2, the obtained distribution can be far from being a uniformly distributed set of points on the Pareto-optimal surface. Niching methods 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 a large number of objectives. The test problems suggested in this study 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 a large number of objectives, it is also important to keep track of the convergence to the true Pareto-optimal front. 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 [6], thereby reducing the effect of the 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.

6.7 Test Problem Suite

Using the latter two approaches of test problem design discussed in this study, we suggest here a representative set of test problems – DTLZ problems named with the first letter of the last names of the authors. In all cases, the problem can be made more difficult by using a different set of variables $\mathbf{y}$ obtained from the decision variable vector $\mathbf{x}$ by a transformation: $\mathbf{y} = \mathcal{M}\mathbf{x}$ (where $\mathcal{M}$ is a $n \times n$ transformation matrix). However, other more interesting and useful test problems can also be designed using the techniques of this study.

6.7.1 Test Problem DTLZ1

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

$$\left.\begin{array}{l} \text{Minimize } f_1(\mathbf{x}) = \frac{1}{2}x_1 x_2 \cdots x_{M-1}(1+g(\mathbf{x}_M)), \\ \text{Minimize } f_2(\mathbf{x}) = \frac{1}{2}x_1 x_2 \cdots (1-x_{M-1})(1+g(\mathbf{x}_M)), \\ \quad \vdots \\ \text{Minimize } f_{M-1}(\mathbf{x}) = \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}_M)), \\ \text{Minimize } f_M(\mathbf{x}) = \frac{1}{2}(1-x_1)(1+g(\mathbf{x}_M)), \\ \text{subject to } 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n. \end{array}\right\} \tag{6.18}$$

The last $k = (n - M + 1)$ variables are represented as $\mathbf{x}_M$. 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:

$$g(\mathbf{x}_M) = 100\left[|\mathbf{x}_M| + \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]. \tag{6.19}$$

The Pareto-optimal solution corresponds to $x_i = 0.5$ (for all $x_i \in \mathbf{x}_M$) and the objective function values lie on the linear hyper-plane: $\sum_{m=1}^{M} f_m^* = 0.5$. The only difficulty provided by this problem is the convergence to the Pareto-optimal hyper-plane. The search space contains $(11^k - 1)$ local Pareto-optimal fronts, where an MOEA can get attracted before reaching the global Pareto-optimal front.

To demonstrate how a couple of state-of-the-art MOEAs perform on this and other test problems suggested in this study, we have applied NSGA-II and SPEA2 to the three-objective ($M = 3$) version of the problems. In all cases, a population of size 100 is used. Since all test problems involve real parameters, the SBX recombination operator (with $\eta_c = 15$) and a variable-wise polynomial mutation operator (with $\eta_m = 20$) [6] are used in all cases. The crossover probability of 1.0 and mutation probability of $1/n$ are used.

The performances of NSGA-II and SPEA2 on DTLZ1 after 300 generations are shown in Figures 6.12 and 6.13, 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 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 needed in the objective-wise crowding approach of NSGA-II. The problem can be made more difficult by using a more difficult g function or a variable transformation technique suggested earlier. By merely increasing the number of decision variables used in $\mathbf{x}$, the problem can be made harder.

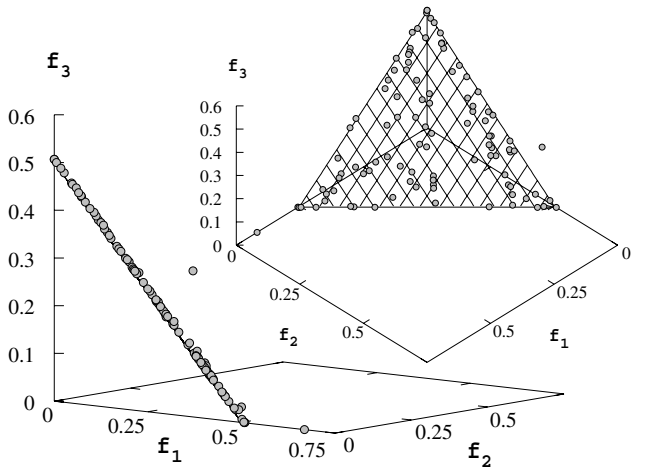

Figure 6.12. The NSGA-II population on test problem DTLZ1.

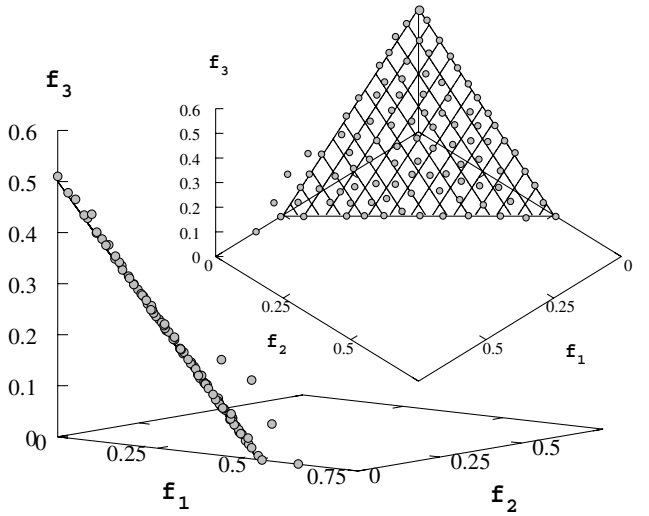

Figure 6.13. The SPEA2 population on test problem DTLZ1.

6.7.2 Test Problem DTLZ2

This test problem is identical to the problem described in Section 6.4.3:

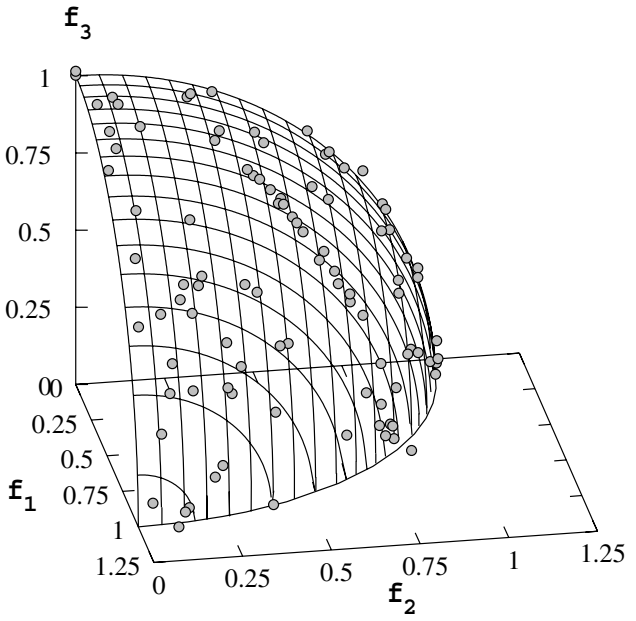

Figure 6.14. The NSGA-II population on test problem DTLZ2.

$$
\left.
\begin{array}{l}
\text{Min. } f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2), \\
\text{Min. } f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1\pi/2) \cdots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2), \\
\text{Min. } f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1\pi/2) \cdots \sin(x_{M-2}\pi/2), \\
\vdots \quad \vdots \\
\text{Min. } f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1\pi/2), \\
\text{with } g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \\
\qquad 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}
\right\} \tag{6.20}
$$

Once again, the $\mathbf{x}$ vector is constructed with $k = n - M + 1$ variables. The Pareto-optimal solutions corresponds to $x_i = 0.5$ for all $x_i \in \mathbf{x}_M$ and all objective function values must satisfy Equation 6.9. NSGA-II and SPEA2 with identical parameter setting as in DTLZ1 simulation runs and with $k = 10$ find Pareto-optimal solutions very close to the true Pareto-optimal front after 300 generations, as shown in Figures 6.14 and 6.15, respectively.

This function can also be used to investigate an MOEA's ability to scale up its performance in a large number of objectives. Like in DTLZ1, for $M > 3$, the Pareto-optimal solutions must lie inside the first octant of the unit sphere in a three-objective plot with f_M as one of the axes. Since all Pareto-optimal solutions need to satisfy $\sum_{m=1}^{M} f_m^2 = 1$, the difference between the left term with the obtained solutions and one can be used as a metric for convergence as well. Besides the suggestions given in DTLZ1, the problem can be made

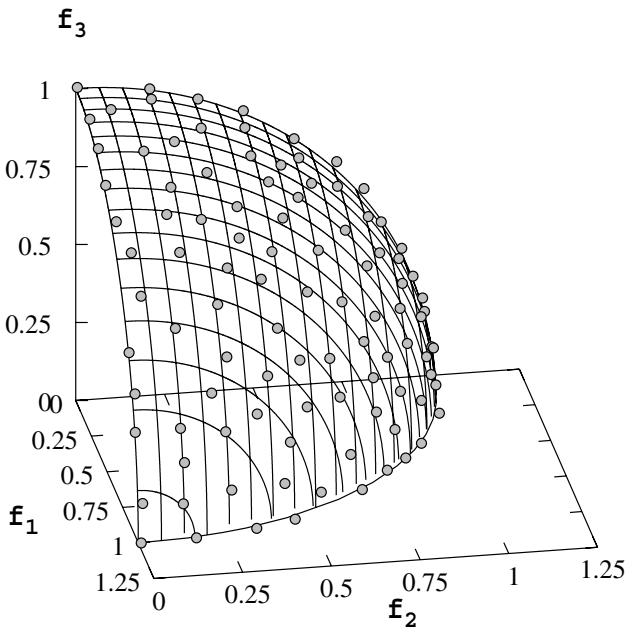

Figure 6.15. The SPEA2 population on test problem DTLZ2.

more difficult by replacing each variable x_i (for $i = 1$ to $(M-1)$) with the mean value of p variables: $x_i = \frac{1}{p}\sum_{k=(i-1)p+1}^{ip} x_k$.

6.7.3 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 6.19:

$$\left.\begin{array}{l}
\text{Min. } f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cdots\cos(x_{M-2}\pi/2)\cos(x_{M-1}\pi/2),\\
\text{Min. } f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cdots\cos(x_{M-2}\pi/2)\sin(x_{M-1}\pi/2),\\
\text{Min. } f_3(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cdots\sin(x_{M-2}\pi/2),\\
\vdots \quad \vdots\\
\text{Min. } f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1\pi/2),\\
\text{with } g(\mathbf{x}_M) = 100\left[|\mathbf{x}_M| + \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right],\\
\qquad 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n.
\end{array}\right\} \tag{6.21}$$

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 $x_i = 0.5$

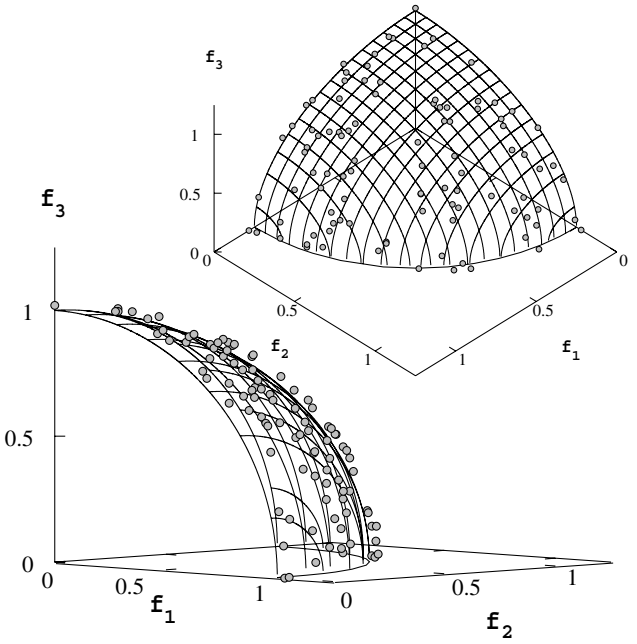

Figure 6.16. The NSGA-II population on test problem DTLZ3.

for $x_i \in \mathbf{x}_M$. The next local Pareto-optimal front is at $g^* = 1$. NSGA-II and SPEA2 populations (using $k = 10$) after 500 generations are shown in the true Pareto-optimal fronts in Figures 6.16 and 6.17. 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 made more difficult by using a larger k or a higher-frequency cosine function.

6.7.4 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 parametric variable mapping:

$$\left.\begin{array}{l}
\text{Min. } f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^\alpha \pi/2) \cdots \cos(x_{M-2}^\alpha \pi/2) \cos(x_{M-1}^\alpha \pi/2), \\
\text{Min. } f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^\alpha \pi/2) \cdots \cos(x_{M-2}^\alpha \pi/2) \sin(x_{M-1}^\alpha \pi/2), \\
\text{Min. } f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(x_1^\alpha \pi/2) \cdots \sin(x_{M-2}^\alpha \pi/2), \\
\vdots \quad \vdots \\
\text{Min. } f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(x_1^\alpha \pi/2), \\
\text{with } g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \\
\qquad 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}\right\} \tag{6.22}$$

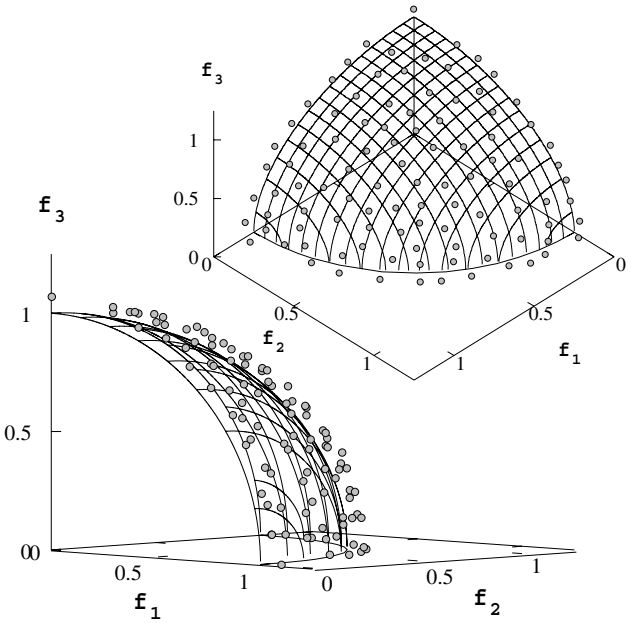

Figure 6.17. The SPEA2 population on test problem DTLZ3.

The parameter $\alpha = 100$ is suggested here. This modification allows a dense set of solutions to exist near the f_M-f_1 plane (as in Figure 6.5). NSGA-II and SPEA2 populations (using $k = 10$) at the end of 200 generations are shown in Figures 6.18 and 6.19, respectively. For this problem, the final population is found to be dependent on the initial population. But in both methods, we have obtained either of the three different outcomes: (1) all solutions are in the f_3-f_1 plane, (2) all solutions are in the f_1-f_2 plane, or (3) 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 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 Section 6.4.2. 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 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.

6.7.5 Test Problem DTLZ5

The mapping of θ_i in the test problem DTLZ2 can be replaced with that given in Equation 6.10:

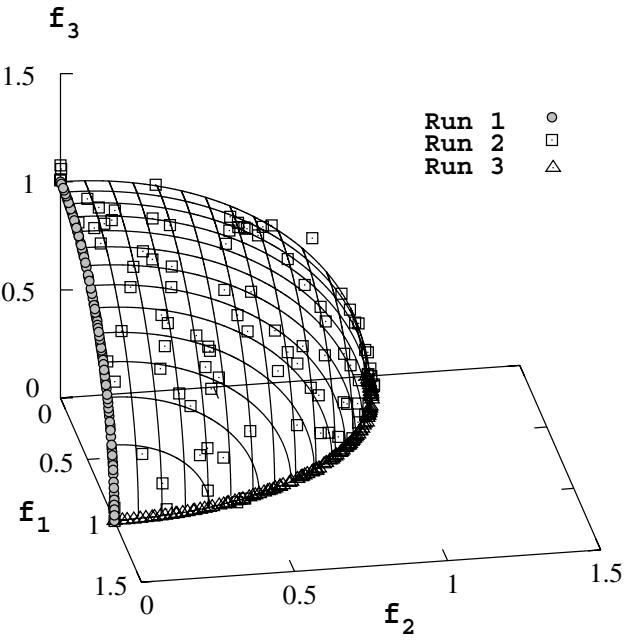

Figure 6.18. The NSGA-II population on test problem DTLZ4. Three different simulation runs are shown.

$$
\left.
\begin{array}{l}
\text{Min. } f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1 \pi/2) \cdots \cos(\theta_{M-2}\pi/2) \cos(\theta_{M-1}\pi/2), \\
\text{Min. } f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1 \pi/2) \cdots \cos(\theta_{M-2}\pi/2) \sin(\theta_{M-1}\pi/2), \\
\text{Min. } f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1 \pi/2) \cdots \sin(\theta_{M-2}\pi/2), \\
\vdots \quad \vdots \\
\text{Min. } f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(\theta_1 \pi/2), \\
\text{with } \theta_i = \frac{\pi}{4(1+g(\mathbf{x}_M))} (1 + 2g(\mathbf{x}_M) x_i), \quad \text{for } i = 2, 3, \ldots, (M-1), \\
\qquad g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \\
\qquad 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}
\right\} \tag{6.23}
$$

The Pareto-optimal front corresponds to $x_i = 0.5$ for all $x_i \in \mathbf{x}_M$ and function values satisfy $\sum_{m=1}^{M} f_m^2 = 1$. 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 6.20 and 6.21 obtained using NSGA-II and SPEA2 after 200 generations and with other parameter settings as before and with $k = 10$.

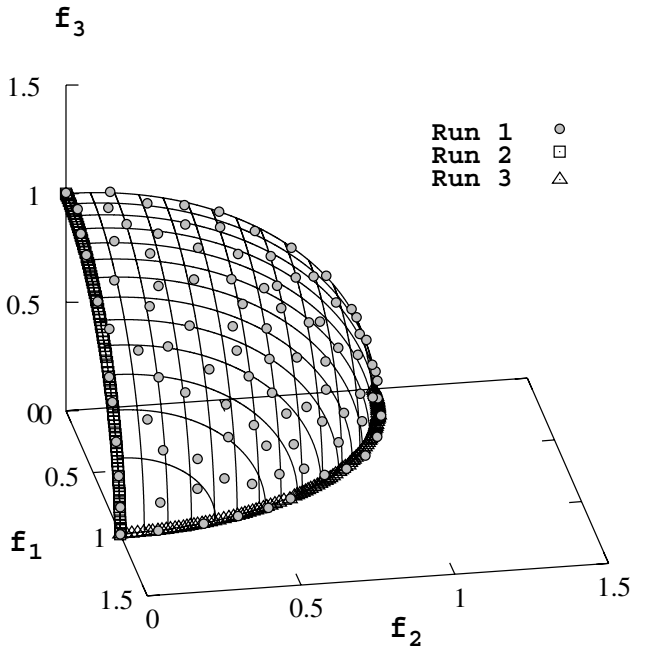

Figure 6.19. The SPEA2 population on test problem DTLZ4. Three different simulation runs are shown.

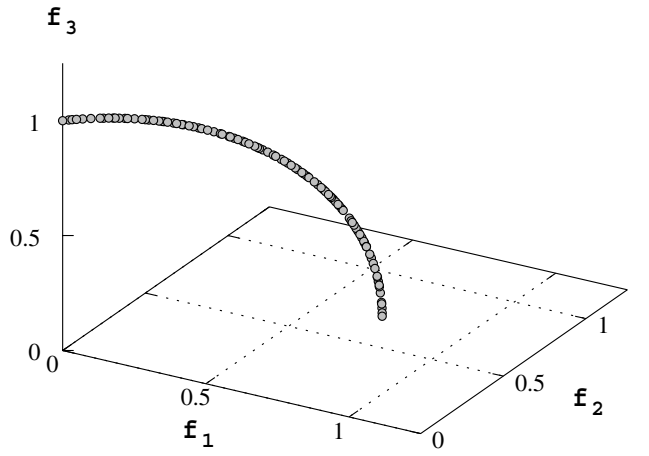

Figure 6.20. The NSGA-II population on test problem DTLZ5.

6.7.6 Test Problem DTLZ6

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

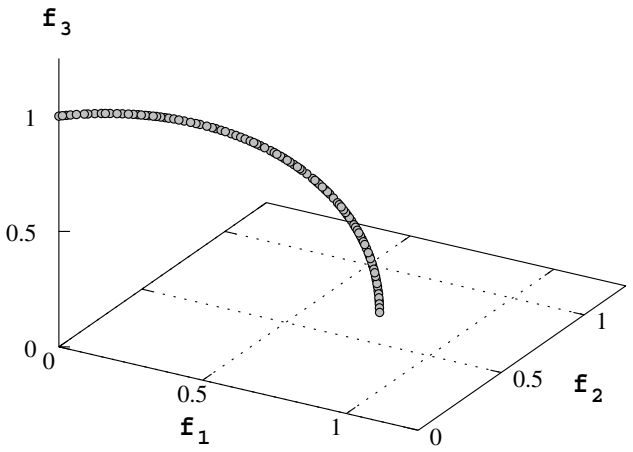

Figure 6.21. The SPEA2 population on test problem DTLZ5.

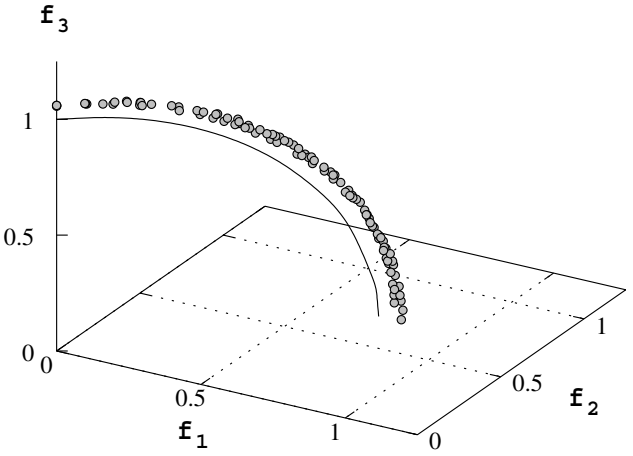

Figure 6.22. The NSGA-II population on test problem DTLZ6.

a different g function:

$$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} x_i^{0.1}. \tag{6.24}$$

Here, the Pareto-optimal front corresponds to $x_i = 0$ for all $x_i \in \mathbf{x}_M$. 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 6.22 and 6.23, 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 Pareto-optimal curve. The lack of convergence to the

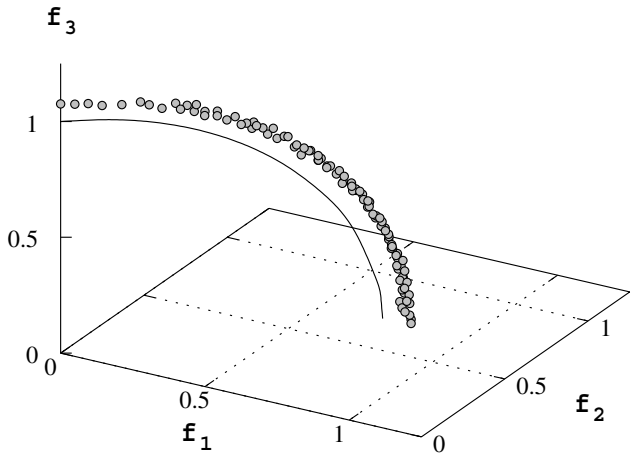

Figure 6.23. The SPEA2 population on test problem DTLZ6.

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 Section 6.8.

6.7.7 Test Problem DTLZ7

This test problem is constructed using the problem stated in Equation 6.11. This problem has a disconnected set of Pareto-optimal regions:

$$\left.\begin{array}{l}
\text{Minimize } f_1(\mathbf{x}_1) = x_1, \\
\text{Minimize } f_2(\mathbf{x}_2) = x_2, \\
\qquad \vdots \\
\text{Minimize } f_{M-1}(\mathbf{x}_{M-1}) = x_{M-1}, \\
\text{Minimize } f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M))h(f_1, f_2, \ldots, f_{M-1}, g), \\
\quad \text{where } g(\mathbf{x}_M) = 1 + \frac{9}{|\mathbf{x}_M|} \sum_{x_i \in \mathbf{x}_M} x_i, \\
\qquad h(f_1, f_2, \ldots, f_{M-1}, g) = M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} \left(1 + \sin(3\pi f_i)\right) \right], \\
\text{subject to } 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}\right\} \tag{6.25}$$

This test problem has 2^{M-1} disconnected Pareto-optimal regions in the search space. The functional g requires $k = |\mathbf{x}_M| = n - M + 1$ decision variables. 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. For a problem with $k = 20$ and $M = 3$, Figures 6.24 and 6.25 show the NSGA-II and SPEA2 populations after 200 generations. It is clear that both algorithms are able to find and maintain stable and distributed subpopulations

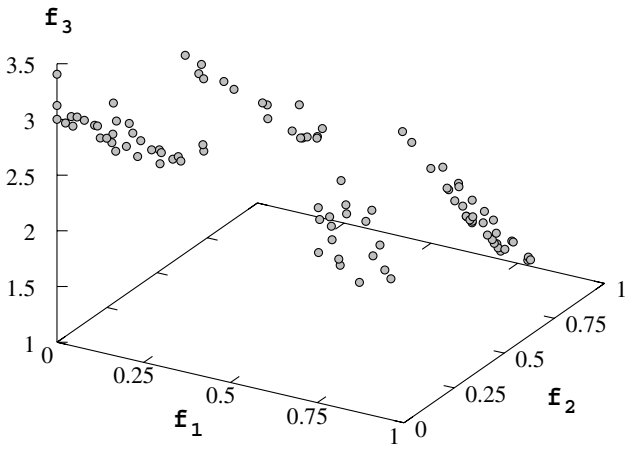

Figure 6.24. The NSGA-II population on test problem DTLZ7.

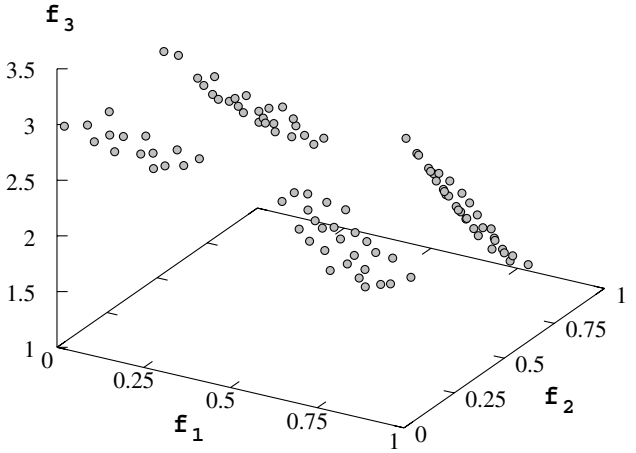

Figure 6.25. The SPEA2 population on test problem DTLZ7.

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 described in Equation 6.19.

6.7.8 Test Problem DTLZ8

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

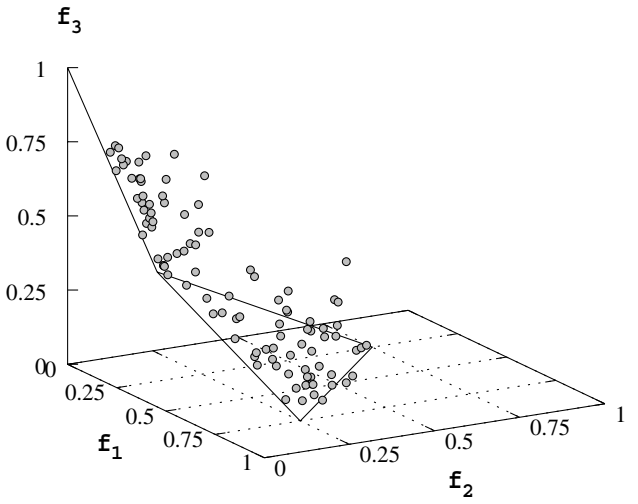

Figure 6.26. The NSGA-II population of non-dominated solutions on test problem DTLZ8.

$$\left.\begin{array}{l} \text{Minimize} \quad f_j(\mathbf{x}) = \frac{1}{\lfloor \frac{n}{M} \rfloor} \sum_{i=\lfloor (j-1)\frac{n}{M} \rfloor}^{\lfloor j \frac{n}{M} \rfloor} x_i, \quad j = 1, 2, \ldots, M, \\ \text{Subject to } g_j(\mathbf{x}) = f_M(\mathbf{x}) + 4 f_j(\mathbf{x}) - 1 \geq 0, \quad \text{for } j = 1, 2, \ldots, (M-1), \\ \qquad\qquad g_M(\mathbf{x}) = 2 f_M(\mathbf{x}) + \min_{\substack{i,j=1 \\ i \neq j}}^{M-1} [f_i(\mathbf{x}) + f_j(\mathbf{x})] - 1 \geq 0, \\ \qquad\qquad 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, n. \end{array}\right\} \tag{6.26}$$

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 6.26 and 6.27 show NSGA-II and SPEA2 populations after 500 generations. The Pareto-optimal region (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 Section 6.8.

6.7.9 Test Problem DTLZ9

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

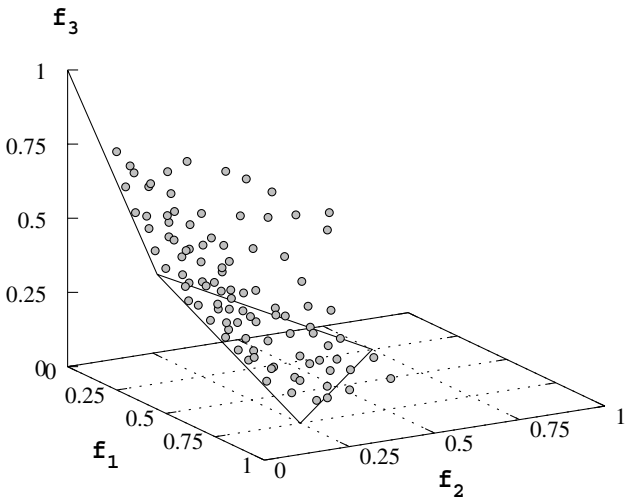

Figure 6.27. The SPEA2 population of non-dominated solutions on test problem DTLZ8.

$$\left.\begin{array}{l} \text{Minimize} \quad f_j(\mathbf{x}) = \sum_{i=\lfloor (j-1)\frac{n}{M} \rfloor}^{\lfloor j \frac{n}{M} \rfloor} x_i^{0.1}, \quad j = 1, 2, \ldots, M, \\ \text{Subject to } g_j(\mathbf{x}) = f_M^2(\mathbf{x}) + f_j^2(\mathbf{x}) - 1 \geq 0, \quad \text{for } j = 1, 2, \ldots, (M-1), \\ \qquad\qquad 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, n. \end{array}\right\} \tag{6.27}$$

Here too, the number of variables is considered to be larger than the number of objectives. For this problem, we also 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 6.28 and 6.29 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 algorithms could not cover the entire range of the circle and there exist many non-dominated solutions away from the Pareto-optimal front.

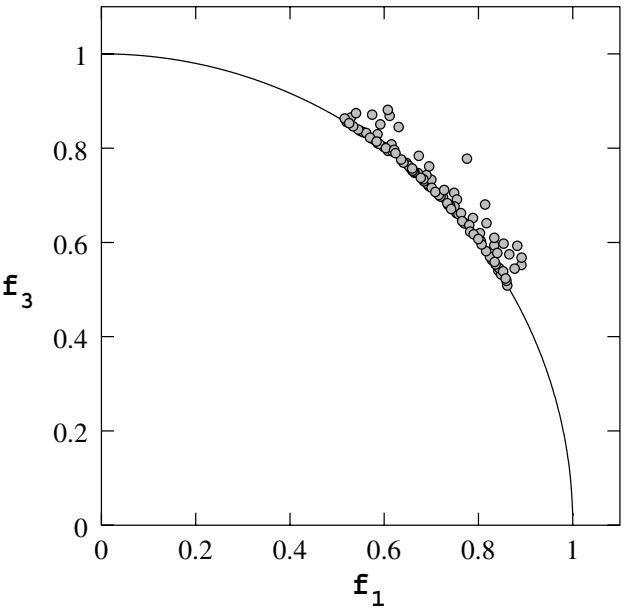

Figure 6.28. The NSGA-II population on test problem DTLZ9.

6.8 Redundant Solutions

Many of the above test problems (such as DTLZ6, DTLZ8, and DTLZ9) introduce a different kind of difficulty to multiobjective real-parameter optimization techniques. In these problems, the Pareto-optimal front 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 front, an MOEA which works with the domination concept can find a set of non-dominated solutions all of which may not be on the true Pareto-optimal front. Figures 6.30 and 6.31 demonstrate this matter, for two and three-objective minimization problems, respectively.

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. That is, if no other Pareto-optimal solutions within the line joining A and B are found, any solution from the shaded region would be non-dominated with both A and B and may exist in an MOEA population. In such cases, the obtained set of solutions may wrongly depict a higher-dimensional surface or a redundant surface as the obtained Pareto-optimal front. Another study [22] has also recognized that this feature of problems can cause domination-based MOEAs difficulty in finding the true Pareto-optimal solutions. It is worth highlighting here that with the increase in the dimensionality of the objective space, the probability of occurrence of such redundant solutions is more. Figures 6.30 and 6.31 illustrate that the region containing such redundant

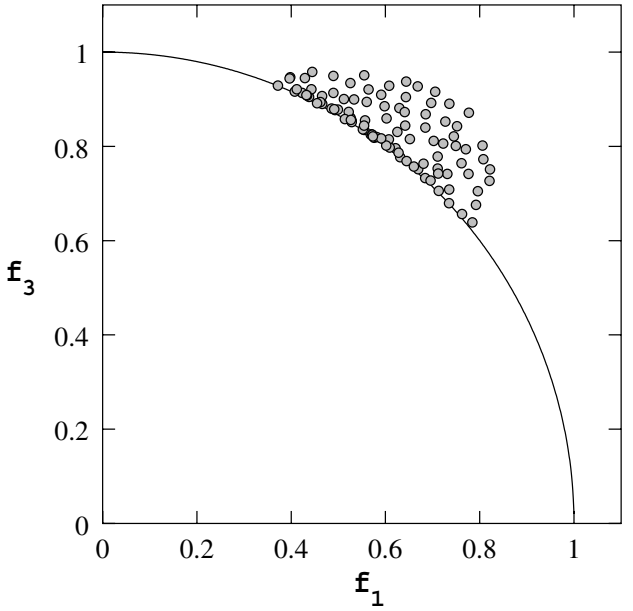

Figure 6.29. The SPEA2 population on test problem DTLZ9.

solutions would in general be more for a problem having more objectives. We term this difficulty associated with the dimension of the objective space as the problem of 'redundancy' in the context of multiobjective optimization. In handling such problems, MOEAs with the newly suggested ϵ-dominance concept [23] introduced by the authors may be found useful. A recent study [24] has demonstrated that the use of ϵ-dominance concept is one way to reduce the redundancy problem. However, this is a serious difficulty in the context to multiobjective optimization and must be addressed further.

6.9 Conclusions

In this study, we have suggested three approaches for systematically designing scalable test problems for multiobjective optimization. The first approach simply uses a translated set of single-objective functions. 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

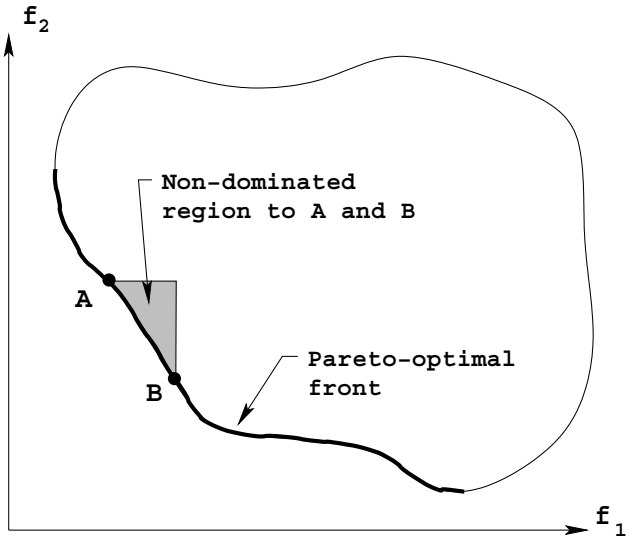

Figure 6.30. The shaded region is non-dominated with Pareto-optimal solutions A and B (for two objectives).

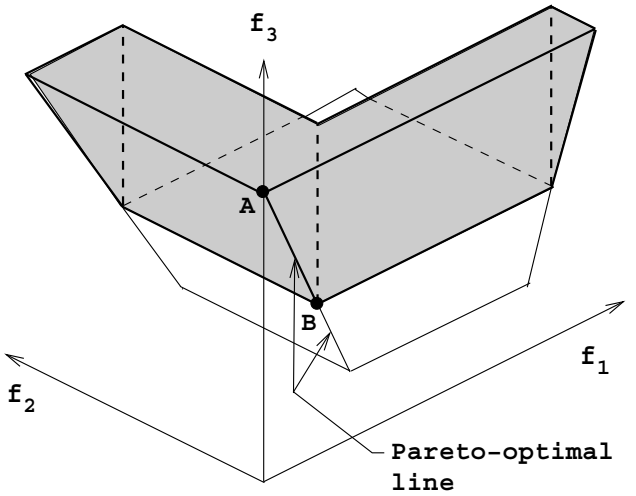

Figure 6.31. The shaded region is non-dominated with Pareto-optimal solutions A and B (for three objectives).

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 study, 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 a two-objective constrained test problem design elsewhere [25] can also be used with the proposed procedures for constrained test problem design.

Acknowledgments

This study originated during the first author's visit to ETH Zurich. The authors acknowledge the support from the Swiss National Science Foundation under the ArOMA project 2100-057156.99/1.

A Appendix: Another Test Problem Using the Bottom-up Approach: The Comet Problem

To demonstrate the ease of using the bottom-up approach to design test problems further, we create one more problem which has a comet-like Pareto-optimal front. Starting from a widely spread region, the Pareto-optimal front continuously reduces to a thinner region. Finding a wide variety of solutions in both broad and thin portions of the Pareto-optimal region simultaneously becomes a challenging task for any multiobjective optimizer, including classical methods:

$$\left.\begin{array}{l} \text{Minimize } f_1(\mathbf{x}) = (1+g(x_3))(x_1^3x_2^2 - 10x_1 - 4x_2), \\ \text{Minimize } f_2(\mathbf{x}) = (1+g(x_3))(x_1^3x_2^2 - 10x_1 + 4x_2), \\ \text{Minimize } f_3(\mathbf{x}) = 3(1+g(x_3))x_1^2, \\ \qquad 1 \le x_1 \le 3.5, \\ \qquad -2 \le x_2 \le 2, \\ \qquad g(x_3) \ge 0. \end{array}\right\} \tag{6.28}$$

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 6.32 shows the Pareto-optimal front on the $x_3 = 0$ surface. For better illustration purposes, the figure is plotted with negative 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

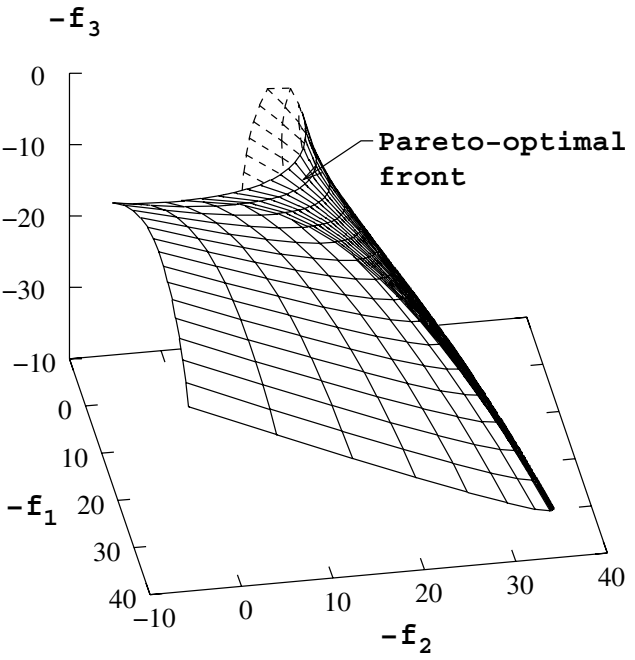

Figure 6.32. The comet problem.

argue that the classical generating methods will require a large computational overhead in solving the above problems. Figure 6.33 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 [6, 26] 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 combinations of ϵ_1 and ϵ_2, is as follows:

$$\left.\begin{array}{rl} \text{Minimize} & f_3(\mathbf{x}), \\ \text{subject to} & f_2(\mathbf{x}) \leq \epsilon_2, \\ & f_1(\mathbf{x}) \leq \epsilon_1, \\ & \mathbf{x} \in \mathcal{D}, \end{array}\right\} \tag{6.29}$$

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 [26]. The figure illustrates a scenario with $\epsilon_2 = -30$. It can be seen from the figure that the 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-

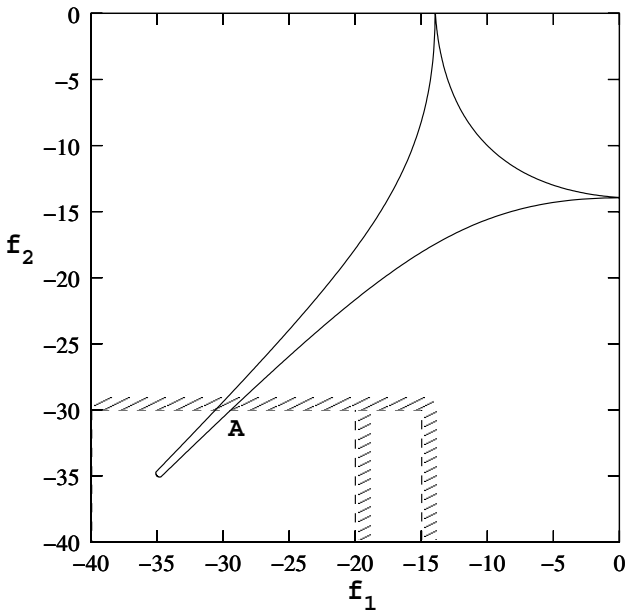

Figure 6.33. Classical generating method with the ϵ-constraint method will produce redundant single-objective optimization problems.

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 is 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 allows them to find a wide variety of solutions simultaneously with the action of a niche-preserving operator.

References

1. Schaffer, JD *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*. PhD thesis, Nashville, TN: Vanderbilt University, 1984.
2. Kursawe, F A Variant of Evolution Strategies for Vector Optimization. In *Parellel Problem Solving from Nature I (PPSN-I)*, pp. 193-197, 1990.
3. Fonseca, CM and Fleming, PJ An Overview of Evolutionary Algorithms in Multi-objective Optimization. *Evolutionary Computation Journal*, 1995; 3(1): 1-16.
4. Poloni, C, Giurgevich, A, Onesti, L and Pediroda, V 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*, 2000; 186(2–4): 403-420.
5. Viennet, R Multicriteria Optimization Using a Genetic Algorithm for Determining the Pareto Set. *International Journal of Systems Science*, 1996; 27(2): 255-260.
6. Deb, K *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.
7. Van Veldhuizen, D *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD Thesis, Dayton, OH: Air Force Institute of Technology, 1999. Technical Report No. AFIT/DS/ENG/99-01.
8. Deb, K Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation Journal*, 1999; 7(3): 205-230.
9. Zitzler, E, Deb, K and Thiele, L Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation Journal*, 2000; 8(2): 125-148.
10. Deb, K, Thiele, L, Laumanns, M and Zitzler, E Scalable Multi-objective Optimization Test Problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pp. 825-830, 2002.
11. Coello, CAC, VanVeldhuizen, DA, and Lamont G *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer Academic Publishers, 2002.
12. Bleuler, S, Laumanns, M, Thiele, L and Zitzler, E PISA - A Platform and Programming Language Independent Interface for Search Algorithms. In *Evolutionary Multi-Criterion Optimization (EMO 2003)*, Lecture Notes in Computer Science, Berlin, 2003. Springer.
13. Laumanns, M, Rudolph, G and Schwefel, HP A Spatial Predator-prey Approach to Multi-objective Optimization: A Preliminary Study. In *Proceedings of the Parallel Problem Solving from Nature, V*, pp. 241-249, 1998.
14. Laumanns, M, Thiele, L, Ziztler, E, Welzl, E and Deb, K Running Time Analysis of Multi-objective Evolutionary Algorithms on a Simple Discrete Optimization Problem. In *Proceedings of the Seventh Conference on Parallel Problem Solving from Nature (PPSN-VII)*, pp. 44-53, 2002.
15. Zitzler, E and Thiele, L Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 1999; 3(4): 257-271.
16. Deb, K and Jain, S Multi-speed Gearbox Design Using Multi-objective Evolutionary Algorithms. *ASME Transactions on Mechanical Design*, 2003; 125(3): 609-619.
17. Laumanns, M, Thiele, L and Zitzler, E Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-boolean Functions. *IEEE Transactions on Evolutionary Computation*, 2004. Accepted for publication.
18. Tanaka, M 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, 1995.
19. Tamaki, H Multi-objective Optimization by Genetic Algorithms: A Review. In *Proceedings of the Third IEEE Conference on Evolutionary Computation*, pp. 517-522, 1996.

20. Knowles, JD and Corne, DW Approximating the Non-dominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation Journal*, 2000; 8(2): 149-172.
21. Deb, K, Agrawal, S, Pratap, A and Meyarivan, T A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002; 6(2):182-197.
22. Kokolo, I, Kita, H, and Kobayashi, S Failure of Pareto-based Moeas: Does Non-dominated Really Mean Near to Optimal? In *Proceedings of the Congress on Evolutionary Computation 2001*, pp. 957-962, 2001.
23. Laumanns, M, Thiele, L, Deb, K and Zitzler, E Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 2002; 10(3): 263-282.
24. Deb, K, Mohan, M, and Mishra, S Towards a Quick Computation of Well-spread Pareto-optimal Solutions. In *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) Conference (LNCS 2632)*, pp. 222-236, 2003.
25. Deb, K, Pratap, A and Meyarivan, T Constrained Test Problems for Multi-objective Evolutionary Optimization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, pp. 284-298, 2001.
26. Miettinen, K, *Nonlinear Multiobjective Optimization*, Boston, Kluwer, 1999.