Scalable Multi-Objective Optimization Test Problems

Kalyanmoy Deb
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur
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

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 suggest 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 ability to control 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 these 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 having a better understanding of the working principles of MOEAs.

I. 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. Viennet's (1996) test problem VNT has a discrete set of Pareto-optimal fronts, but was designed for three objectives only. Similar shortcomings prevail 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. 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 with arbitrary number of objectives. Besides testing an MOEA's ability to solve problems with many objectives, the proposed test problems can also be used for systematically comparing two or more 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.

II. Desired Features of Test Problems

Here, we suggest that the following features must be present in a test problem suite for adequately testing an MOEA: (i) test problems should be easy to construct, (ii) test problems should be scalable to have any number of decision variables, (iii) test problems should be scalable to have any number of objectives, (iv) the resulting Pareto-optimal front (continuous or discrete) must be easy to comprehend, and its shape and location should be exactly known, and (v) 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.

III. Different Methods of Test Problem Design

We discuss a number of different ways to systematically design test problems for multi-objective optimization: (i) multiple single-objective functions approach, (ii) bottom-up approach, and (iii) 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): (i) $f_1(x) = x^2$ (ii) $f_2(x) = (x-2)^2$.

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. 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). Although it may look simple, the resulting Pareto-optimal set may be difficult to comprehend in such problems.

IV. 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. 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:

$$\left.\begin{array}{rcl} f_1(\theta,\gamma) & = & \cos\theta\cos(\gamma+\pi/4), \\ f_2(\theta,\gamma) & = & \cos\theta\sin(\gamma+\pi/4), \\ f_3(\theta,\gamma) & = & \sin\theta, \\ \text{where} & & 0 \le \theta \le \pi/2, \quad -\pi/4 \le \gamma \le \pi/4. \end{array}\right\} \quad (1)$$

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 can be achieved by multiplying the above three functions with a term, which takes a value greater than or equal to one:

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

As described earlier, the Pareto-optimal solutions for the above problem are as follows: $0 \le \theta^* \le \pi/2$, $-\pi/4 \le \gamma^* \le \pi/4$, $g(r^*) = 0$. Figure 2 shows the overall objective search space with any function for $g(r)$ with $0 \le g(r) \le 1$.

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:

$$\theta = \theta(x_1,\ldots,x_n), \quad \gamma = \gamma(x_1,\ldots,x_n), \quad r = r(x_1,\ldots,x_n).$$

The functions must be so chosen that they satisfy the lower and upper bounds of θ, γ and $g(r)$ mentioned in equation 2. The above construction procedure can be used to introduce different modes of difficulty described earlier.

A. 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 $g(r) = r^{0.1}$ is used, denser solutions exist away from the Pareto-optimal front.

B. Difficulties across the Pareto-optimal front

By using a non-linear meta-variable mapping, 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 must be manipulated. 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.

C. 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:

$$\left.\begin{array}{rl} \text{Minimize} & f_1(\mathbf{x}_1), \\ \vdots & \vdots \\ \text{Minimize} & f_{M-1}(\mathbf{x}_{M-1}), \\ \text{Minimize} & f_M(\mathbf{x}) = g(\mathbf{x}_M)h(f_1,\ldots,f_{M-1},g), \\ \text{subject to} & \mathbf{x}_i \in \mathbb{R}^{|\mathbf{x}_i|}, \quad \text{for } i = 1,2,\ldots,M. \end{array}\right\} \quad (3)$$

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 $f_M = g^* h(f_1, f_2, \ldots, f_{M-1})$. In the bottom-up approach of test problem design, the user can first choose an h function in terms of the objective function values. For example, 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 3 illustrates a disconnected set of

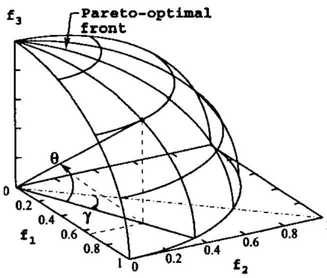

Fig. 1. First quadrant of a unit sphere as a Pareto-optimal front.

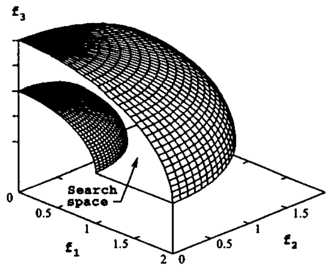

Fig. 2. Overall search space is bounded by the two spheres.

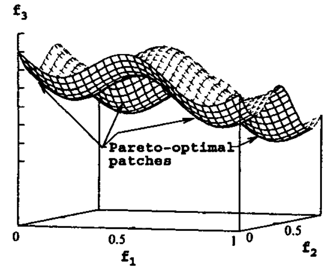

Fig. 3. A disjoint set of Pareto-optimal regions.

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

$$h(f_1, f_2, \ldots, f_{M-1}) = 2M - \sum_{i=1}^{M-1} (2f_i + \sin(3\pi f_i)). \quad (4)$$

Once the h function is chosen, a g function can be chosen to construct the entire objective search space. 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.

D. 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 search space is constructed from an identical functional, the search space is structured.

V. Constraint Surface Approach

The constraint surface approach begins by predefining a simple overall search space:

$$\left.\begin{array}{lll} \text{Minimize} & f_1(\mathbf{x}), & \\ \vdots & \vdots & \\ \text{Minimize} & f_M(\mathbf{x}), & \\ \text{Subject to} & f_i^{(L)} \le f_i(\mathbf{x}) \le f_i^{(U)} & \text{for } i = 1, 2, \ldots, M. \end{array}\right\} \quad (5)$$

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 $(f_1^{(L)}, f_2^{(L)}, \ldots, f_1^{(L)})^T$. The problem is now made more interesting by adding a series of constraints (linear or non-linear):

$$g_j(f_1, f_2, \ldots, f_M) \ge 0, \quad \text{for } j = 1, 2, \ldots, J. \quad (6)$$

Each constraint practically eliminates some portion of the original rectangular region systematically. Figure 4 shows the resulting feasible region after adding the following two linear constraints:

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

The objective of the above problem is to find the non-dominated portion of the boundary of the feasible search space. Difficulties can be introduced by using varying density

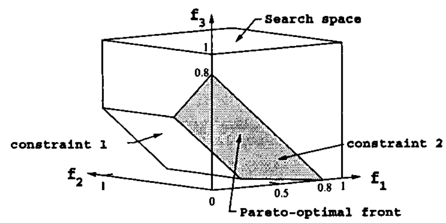

Fig. 4. Two constraints eliminate a portion of the cube.

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 in Tanaka (1995) and in Tamaki (1996) using the above concept.

A. Advantages and Disadvantages

The construction process here is much simpler compared to the bottom-up approach. Using this procedure, different shapes (convex, non-convex, or discrete) of the Pareto-optimal region can be generated.

However, the resulting Pareto-optimal front will, in general, be hard to express mathematically and to comprehend visually. 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.

VI. 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 measured by them and suggested three different categories: (i) metrics that evaluate closeness to the Pareto-optimal front, (ii) metrics that evaluate diversity in obtained solutions, and (iii) 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^*. Although almost all metrics for convergence suggested in the context of two-objective problems can be applied to problems having more than two objectives, most existing performance metrics for evaluating the distribution of solutions can not be used in higher-objective optimization problems. This is because the calculation of diversity measure in higher dimensions is not straightforward and often computationally expensive. Further research in this direction is necessary to develop better performance metrics.

VII. 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.

A. Test Problem DTLZ1

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

$$\left.\begin{array}{rl}\text{Minimize} & f_1(\mathbf{x}) = \frac{1}{2}x_1x_2\cdots x_{M-1}(1+g(\mathbf{x}_M)),\\ \text{Minimize} & f_2(\mathbf{x}) = \frac{1}{2}x_1x_2\cdots(1-x_{M-1})(1+g(\mathbf{x}_M)),\\ \vdots & \vdots\\ \text{Minimize} & f_{M-1}(\mathbf{x}) = \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}_M)),\\ \text{Minimize} & f_M(\mathbf{x}) = \frac{1}{2}(1-x_1)(1+g(\mathbf{x}_M)),\\ \text{subject to} & 0 \le x_i \le 1, \quad \text{for } i=1,2,\ldots,n.\end{array}\right\} \tag{7}$$

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:

$$g(\mathbf{x}_M) = 100\left(|\mathbf{x}_M| + \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right). \tag{8}$$

The Pareto-optimal solution corresponds to $x_i^* = 0.5$ ($x_i^* \in \mathbf{x}_M$) and the objective function values lie on the linear hyper-plane: $\sum_{m=1}^{M} f_m^* = 0.5$. 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 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. The performance of NSGA-II is shown in Figure 5. The figure shows that NSGA-II is able to converge close to the global Pareto-optimal front with a wide spread of solutions, but the distribution of solutions across the Pareto-optimal front is not uniform. A better distribution of solutions (at the cost of increased computational time) was obtained using SPEA2 (Zitzler, Laumanns, and Thiele, 2001) and is not reported here.

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.

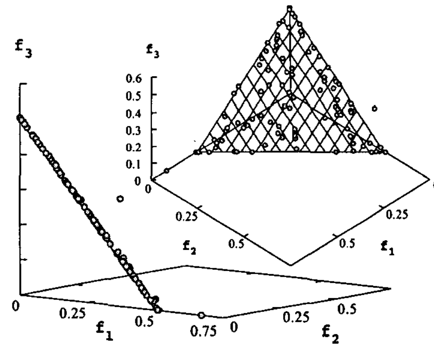

Fig. 5. The NSGA-II population on test problem DTLZ1.

B. Test Problem DTLZ2

This test problem has a spherical Pareto-optimal front as in Figure 1:

$$\begin{array}{rl}\text{Minimize} & f_1(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2),\\ \text{Minimize} & f_2(\mathbf{x}) = (1+g(\mathbf{x}_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2),\\ \vdots & \vdots\\ \text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))\sin(x_1\pi/2),\\ & 0\le x_i\le 1, \quad \text{for } i=1,2,\ldots,n,\\ \text{where} & g(\mathbf{x}_M) = \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2.\end{array} \tag{9}$$

The Pareto-optimal solutions corresponds to $x_i^* = 0.5$ ($x_i^* \in \mathbf{x}_M$) and all objective function values must satisfy the $\sum_{m=1}^{M}(f_m^*)^2 = 1$. As in the previous problem, the distribution of solutions ($k = |\mathbf{x}_M| = 10$) obtained with NSGA-II is poor. This function can also be used to investigate an MOEA's ability to scale up its performance in large number of objectives.

C. 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 8. This introduces many local Pareto-optimal fronts, on which an MOEA can get attracted. NSGA-II population after 500 generations could not converge to the global Pareto-optimal front (results are not shown here for brevity).

D. 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: $x_i \to x_i^\alpha$. The parameter $\alpha = 100$ is suggested here. This modification allows a dense set of solutions to exist near the f_M-f_1 plane. NSGA-II population at the end of 200 generations are shown in Figure 6. For this problem, the final population is dependent on the initial population, as shown in the figure for three independent NSGA-II runs.

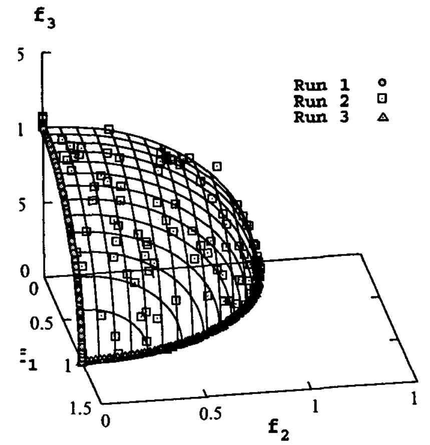

Fig. 6. The NSGA-II population on test problem DTLZ4. Three different simulation runs are shown.

E. Test Problem DTLZ5

The mapping of θ_i in the test problem DTLZ2 can be replaced with $\theta_i = \frac{\pi}{4(1+g(r))}(1+2g(r)x_i)$, for $i = 2, 3, \ldots, (M-1)$, and $g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} x_i^{0.1}$ is used. This problem will test an MOEA's ability to converge to a degenerated 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. The size of $\mathbf{x}_M$ vector is chosen as 10. The population after 500 generations of NSGA-II is shown in Figure 7. The lack of convergence to the true front in this problem causes NSGA-II to find a dominated surface as the obtained front, whereas the true Pareto-optimal front is a curve. Similar performance is also observed with SPEA2.

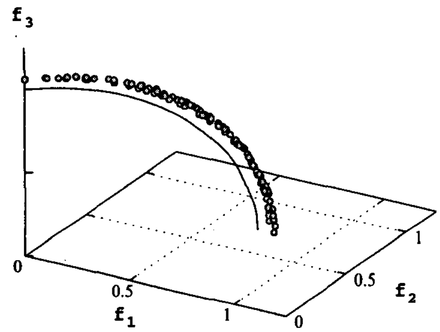

Fig. 7. The NSGA-II population on test problem DTLZ5.

F. Test Problem DTLZ6

This problem is constructed based on equation 3:

$$\begin{array}{rl} \text{Minimize} & f_1(\mathbf{x}_1) = x_1, \\ \vdots & \vdots \\ \text{Minimize} & f_{M-1}(\mathbf{x}_{M-1}) = x_{M-1}, \\ \text{Minimize} & f_M(\mathbf{x}) = (1+g(\mathbf{x}_M))h(f_1, f_2, \ldots, f_{M-1}, g), \\ \text{where} & g(\mathbf{x}_M) = 1 + \frac{9}{|\mathbf{x}_M|}\sum_{x_i \in \mathbf{x}_M} x_i, \\ & h = M - \sum_{i=1}^{M-1}\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right], \\ \text{subject to} & 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n. \end{array} \tag{10}$$

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 $k = 20$. This problem will test an algorithm's ability to maintain subpopulation in different Pareto-optimal regions. For brevity, we do not show any simulation results here.

G. Test Problem DTLZ7

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

$$\begin{array}{rl} \text{Min.} & f_j(\mathbf{x}) = \frac{1}{\lfloor \frac{n}{M} \rfloor}\sum_{i=\lfloor (j-1)\frac{n}{M}\rfloor}^{\lfloor j\frac{n}{M}\rfloor} x_i, \quad j = 1, \ldots, M, \\ \text{s.t.} & g_j(\mathbf{x}) = f_M(\mathbf{x}) + 4f_j(\mathbf{x}) - 1 \ge 0, \quad \text{for } j = 1, \ldots, (M-1), \\ & g_M(\mathbf{x}) = 2f_M(\mathbf{x}) + \min_{\substack{i,j=1 \\ i \ne j}}^{M-1}\left[f_i(\mathbf{x}) + f_j(\mathbf{x})\right] - 1 \ge 0, \\ & 0 \le x_i \le 1, \quad \text{for } i = 1, \ldots, n. \end{array} \tag{11}$$

Here, 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. Figure 8 shows NSGA-II population after 500 generations.

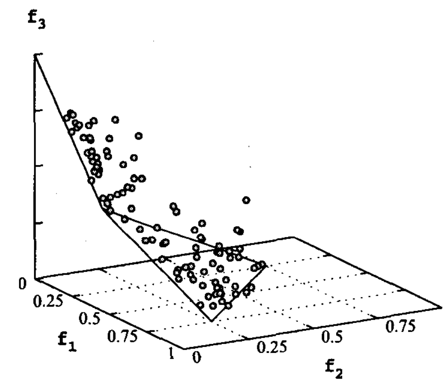

Fig. 8. The NSGA-II population of non-dominated solutions on test problem DTLZ7.

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. Figure 9 demonstrates this matter. With

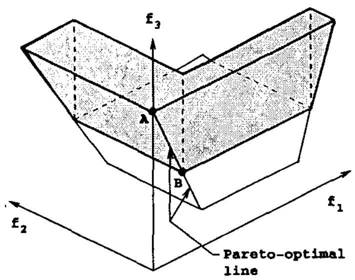

Fig. 9. The shaded region is non-dominated with Pareto-optimal solutions A and B.

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 (as in DTLZ5 or DTLZ7), 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. 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, it is worth highlighting here that with the increase in the dimensionality of the objective space, the probability of occurrence of such difficulties is higher.

VIII. 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. 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 portions of the original hyper-box. A number of test problems have been suggested and attempted to solve using two popular state-of-the-art MOEAs for their systematic use in practice.

Acknowledgment

The support from Swiss National Science Foundation under the ArOMA project 2100-057156.99/1 is acknowledged.

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., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Proc. of the Parallel Problem Solving from Nature VI,* (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.

Kokolo, I., Kita, H., and Kobayashi, S. (2001). Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? *Proc. 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. *Proc. of the 7th International Mendel Conference on Soft Computing (MENDEL 2001).*

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.

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

Tanaka, M. (1995). GA-based decision support system for multi-criteria optimization. In *Proc. 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.* PhD Thesis, Dayton: 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., 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.