Constrained Test Problems for Multi-objective Evolutionary Optimization

Kalyanmoy Deb, Amrit Pratap, and T. Meyarivan

Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
{deb,apratap,mary}@iitk.ac.in
http://www.iitk.ac.in/kangal

Abstract. Over the past few years, researchers have developed a number of multi-objective evolutionary algorithms (MOEAs). Although most studies concentrated on solving unconstrained optimization problems, there exists a few studies where MOEAs have been extended to solve constrained optimization problems. As the constraint handling MOEAs gets popular, there is a need for developing test problems which can evaluate the algorithms well. In this paper, we review a number of test problems used in the literature and then suggest a set of tunable test problems for constraint handling. Finally, NSGA-II with an innovative constraint handling strategy is compared with a couple of existing algorithms in solving some of the test problems.

1 Introduction

Multi-objective evolutionary algorithms (MOEAs) have amply demonstrated the advantage of using population-based search algorithms for solving multi-objective optimization problems [1,14,8,5,6,11]. In multi-objective optimization problems of varying degrees of complexities, elitist MOEAs have demonstrated their abilities in converging close to the true Pareto-optimal front and in maintaining a diverse set of solutions. Despite all these developments, there seem to be not enough studies concentrating procedures for handling constraints. Constraint handling is a crucial part of real-world problem solving and it is time that MOEA researchers focus on solving constrained multi-objective optimization problems.

To evaluate any algorithm, there is a need for well understood and tunable test problems. Although there exists a few constraint handling procedures in the literature, they all have been tried on simple problems, particularly having only a few decision variables and having inadequately nonlinear constraints. When an algorithm performs well on such problems, it becomes difficult to evaluate the true merit of it, particularly in the context of its general problem solving.

In this paper, we briefly outline a few constraint handling methods suggested in the literature. Thereafter, we review a few popular test problems for their degrees of difficulty. Next, we propose a *tunable* test problem generator

E. Zitzler et al. (Eds.): EMO 2001, LNCS 1993, pp. 284–298, 2001.
© Springer-Verlag Berlin Heidelberg 2001

having six parameters, which can be set to obtain constrained test problems of desired degree of difficulty. Simulation results with NSGA-II [1] along with the constrained-domination principle and with two other constrained handling procedures bring out the superiority of NSGA-II in solving difficult constrained optimization problems. The difficulty demonstrated by the test problems will qualify them to be used as standard constrained multi-objective test problems in the years to come.

2 Constrained Multi-objective Evolutionary Algorithms

There exists only a few studies where an MOEA is specifically designed for handling constraints. Among all methods, the usual penalty function approach [11,2] where a penalty proportional to the total constraint violation is added to all objective functions. When applying this procedure, all constraints and objective functions must be normalized.

Jimenez et al. [7] suggested a procedure which carefully compares two solutions in a binary tournament selection. If one solution is feasible and other is not, the feasible solution is chosen. If both solutions are feasible, Horn et al.'s [6] niched-Pareto GA is used. On the other hand, if both solutions are infeasible, the solution closer to the constraint boundary is more likely to be chosen.

Ray et al. [10] suggested a more elaborate constraint handling technique, where constraint violations of all constraints are not simply summed together, instead a non-domination check of constraint violations is made. Based on separate non-domination sorting of a population using objective functions alone, constraint violations alone, and objective function and constraint violation together, the algorithm demands a large computational complexity.

Recently, Deb et al. [1] defined a *constraint-domination* principle, which differentiates infeasible from feasible solutions during the non-dominated sorting procedure:

Definition 1 *A solution i is said to constrained-dominate a solution j, if any of the following conditions is true:*

1. *Solution i is feasible and solution j is not.*
2. *Solutions i and j are both infeasible, but solution i has a smaller overall constraint violation.*
3. *Solutions i and j are feasible and solution i dominates solution j.*

This requires no additional constraint-handling procedure to be used in an MOEA. Since the procedure is generic, the above constraint-domination principle can be used with any unconstrained MOEAs.

3 Past Test Problems

In the context of multi-objective optimization, constraints may cause hindrance to an multi-objective EA (MOEA) to converge to the true Pareto-optimal region and also may cause difficulty in maintaining a diverse set of Pareto-optimal solutions. It is intuitive that the success of an MOEA in tackling both these problems will largely depend on the constraint-handling technique used. However, first we present some test problems commonly used in the literature and then discuss a systematic procedure of developing a test problem generator. Veldhuizen [13] cites a number of constrained test problems used by several researchers. We present here three problems which are often used by researchers.

Srinivas and Deb [11] used the following function:

$$\mathsf{SRN}: \begin{cases} \text{Minimize} & f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \\ \text{Minimize} & f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2, \\ \text{Subject to} & c1(\mathbf{x}) \equiv x_1^2 + x_2^2 \leq 225, \\ & c2(\mathbf{x}) \equiv x_1 - 3x_2 + 10 \leq 0, \\ & -20 \leq x_1 \leq 20, \\ & -20 \leq x_2 \leq 20. \end{cases} \tag{1}$$

Figure 1 shows the feasible decision variable space and the Pareto-optimal set. The Pareto-optimal solutions [11] correspond to $x_1^* = -2.5$, $x_2^* \in [-14.79, 2.50]$. The feasible objective space along with the Pareto-optimal solutions are shown in Figure 2. The only difficulty the constraints introduce in this problem is that

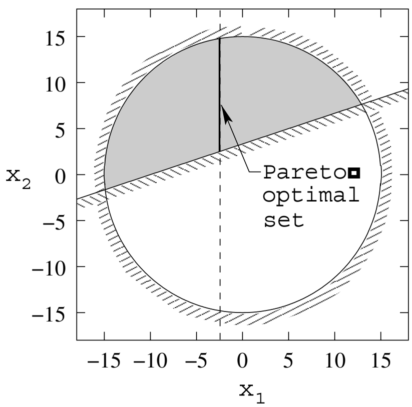

Fig. 1. Decision variable space for the test problem SRN.

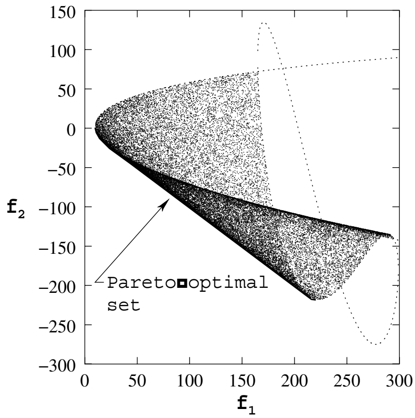

Fig. 2. Objective space for the test problem SRN.

they eliminate some part of the unconstrained Pareto-optimal set (shown with a dashed line in Figure 1).

Tanaka [12] suggested the following two-variable problem:

$$\text{TNK}: \begin{cases} \text{Minimize} \quad f_1(\mathbf{x}) = x_1, \\ \text{Minimize} \quad f_2(\mathbf{x}) = x_2, \\ \text{Subject to } c1(\mathbf{x}) \equiv x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\frac{x}{y}\right) \geq 0, \\ \qquad\qquad c2(\mathbf{x}) \equiv (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \\ \qquad\qquad 0 \leq x_1 \leq \pi, \\ \qquad\qquad 0 \leq x_2 \leq \pi. \end{cases} \tag{2}$$

The feasible decision variable space is shown in Figure 3. Since $f_1 = x_1$ and

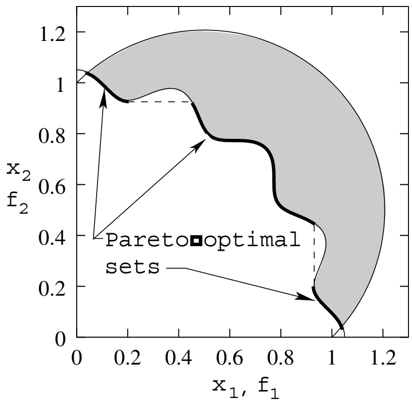

Fig. 3. The feasible decision variable and objective search spaces is shown for TNK.

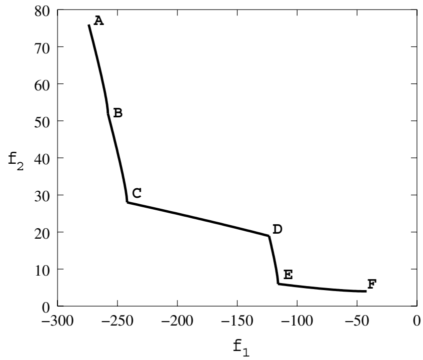

Fig. 4. Pareto-optimal region for the test problem OSY.

$f_2 = x_2$, the feasible objective space is also the same as the feasible decision variable space. The unconstrained decision variable space consists of all solutions in the square $0 \leq (x_1, x_2) \leq \pi$. Thus, the only unconstrained Pareto-optimal solution is $x_1^* = x_2^* = 0$. However, the inclusion of the first constraint makes this solution infeasible. The constrained Pareto-optimal solutions lie on the boundary of the first constraint. Since the constraint function is periodic and the second constraint function must also be satisfied, not all solutions on the boundary is Pareto-optimal. The disconnected Pareto-optimal sets are shown in the figure with thick solid lines. Since the Pareto-optimal solutions lie on a non-linear constraint surface, an optimization algorithm may find difficulty in finding a spread of solutions across the entire Pareto-optimal front.

Osyczka and Kundu [9] used the following six-variable test problem:

$$
\text{OSY}: \begin{cases}
\text{Minimize} & f_1(\mathbf{x}) = -\big(25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 \\
 & \qquad\qquad +(x_5-1)^2\big), \\
\text{Minimize} & f_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2, \\
\text{Subject to} & c_1(\mathbf{x}) \equiv x_1 + x_2 - 2 \geq 0, \\
 & c_2(\mathbf{x}) \equiv 6 - x_1 - x_2 \geq 0, \\
 & c_3(\mathbf{x}) \equiv 2 - x_2 + x_1 \geq 0, \\
 & c_4(\mathbf{x}) \equiv 2 - x_1 + 3x_2 \geq 0, \\
 & c_5(\mathbf{x}) \equiv 4 - (x_3-3)^2 - x_4 \geq 0, \\
 & c_6(\mathbf{x}) \equiv (x_5-3)^2 + x_6 - 4 \geq 0, \\
 & 0 \leq x_1, x_2, x_6 \leq 10, 1 \leq x_3, x_5 \leq 5, 0 \leq x_4 \leq 6.
\end{cases}
\tag{3}
$$

There are six constraints, of which four are linear. Since, it is a six variable problem, it is difficult to show the feasible decision variable space.However, a careful analysis of the constraints and the objective function reveals the constrained Pareto-optimal front, as shown in Figure 4. The Pareto-optimal region is a concatenation of five regions. Every region lies on the intersection of certain constraints. But for the entire Pareto-optimal region, $x_4^* = x_6^* = 0$. Table 1 shows the other variable values for each of the five regions. Since the entire

Table 1. Pareto-optimal regions for the problem OSY.

Region	Optimal values				Active constraints
	x_1^*	x_2^*	x_3^*	x_5^*	
AB	5	1	$(1,\ldots,5)$	5	2,4,6
BC	5	1	$(1,\ldots,5)$	1	2,4,6
CD	$(4.056,\ldots,5)$	$(x_1^*-2)/3$	1	1	4,5,6
DE	0	2	$(1,\ldots,3.732)$	1	1,3,6
EF	$(0,\ldots,1)$	$2-x_1^*$	1	1	1,5,6

Pareto-optimal region demands an MOEA population to maintain subpopulations at different intersections of constraint boundaries, it is a difficult problem to solve.

4 Proposed Test Problems

Although the above test problems (and many others that are discussed elsewhere [13]) somewhat test an MOEA's ability to handle constrained multi-objective optimization problems, they have a number of difficulties: (i) they have a few decision variables, (ii) most objective functions and constraints are not adequately nonlinear, and (iii) they are not tunable for introducing varying degrees of complexity in constrained optimization. Like in the case of tunable unconstrained test problems [3], we also suggest here a number of test problems where the complexity of constrained search space can be controlled. The proposed test problems are designed to cause two different kinds of tunable difficulty to a multi-objective optimization algorithm:

1. Difficulty near the Pareto-optimal front, and
2. Difficulty in the entire search space.

We discuss them in the following subsections.

4.1 Difficulty in the Vicinity of the Pareto-Optimal Front

In this test problem, constraints do not make any major portion of the search region infeasible, except very near to the Pareto-optimal front. Constraints are designed in a way so that some portion of the the unconstrained Pareto-optimal region becomes infeasible. This way, the overall Pareto-optimal front will constitute some part of the unconstrained Pareto-optimal region and some part of the constraint boundaries. In the following, we present a generic test problem:

$$\mathsf{CTP1}: \begin{cases} \text{Minimize } f_1(\mathbf{x}) = x_1 \\ \text{Minimize } f_2(\mathbf{x}) = g(\mathbf{x}) \exp(-f_1(\mathbf{x})/g(\mathbf{x})) \\ \text{Subject to } c_j(\mathbf{x}) = f_2(\mathbf{x}) - a_j \exp(-b_j f_1(\mathbf{x})) \geq 0, \quad j = 1, 2, \ldots, J \end{cases} \tag{4}$$

Here, the function $g(\mathbf{x})$ can be a complex multi-variable function. There are J inequality constraints and the parameters (a_j, b_j) must be chosen in a way so that a portion of the unconstrained Pareto-optimal region is infeasible. We describe a procedure to calculate (a_j, b_j) parameters for J constraints:

Step 1 Set $j = 0$, $a_j = b_j = 1$. Also set $\Delta = 1/(J+1)$ and $x = \Delta$.
Step 2 Calculate $y = a_j \exp(-b_j x)$ and $a_{j+1} = (a_j + y)/2$, $b_{j+1} = -\frac{1}{x} \ln(y/a_{j+1})$. Increment $x = x + \Delta$ and $j = j + 1$.
Step 3 If $j < J$, Go to Step 2, else Stop.

For $J = 2$ constraints, the following parameter values are found:

j	a_j	b_j
1	0.858	0.541
2	0.728	0.295

Figure 5 shows the unconstrained Pareto-optimal region (in a dashed line), and two constraints. With the presence of both constraints, the figure demonstrates that about one-third portion of the original unconstrained Pareto-optimal region is now not feasible. The other two-third region of the constrained Pareto-optimal region comes from the two constraints.

Since a part of the constraint boundary of each constraint now constitutes the Pareto-optimal region, a spread in Pareto-optimal solutions requires decision

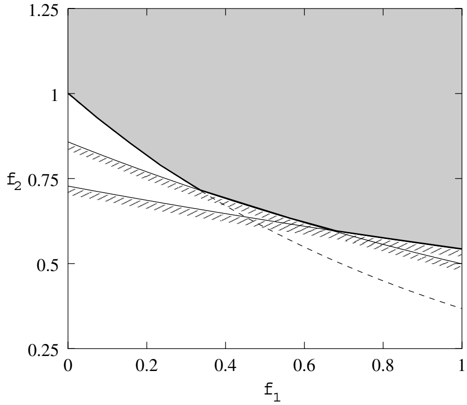

Fig. 5. Constrained test problem CTP1 with two $J = 2$ constraints.

variables ($\mathbf{x}$) to satisfy the inequality constraints with the equality sign. Each constraint is an implicit non-linear function of decision variables. Thus, it may be difficult to find a number of solutions on a non-linear constraint boundary. The complexity of the test problem can be further increased by using more constraints and by using a multi-modal or deceptive g function.

Besides finding and maintaining correlated decision variables to fall on the several constraint boundaries, there could be other difficulties near the Pareto-optimal front. The constraint functions can be such that the unconstrained Pareto-optimal region is now infeasible and the resulting Pareto-optimal set is a collection of a number of discrete regions. At the extreme, the Pareto-optimal region can become a collection of a set of discrete solutions. Let us first present such a function mathematically and then describe the difficulties in each case.

$$\text{CTP2–CTP7}: \left\{ \begin{array}{l} \text{Minimize } f_1(\mathbf{x}) = x_1 \\ \text{Minimize } f_2(\mathbf{x}) = g(\mathbf{x}) \left(1 - \frac{f_1(\mathbf{x})}{g(\mathbf{x})}\right) \\ \text{Subject to } c(\mathbf{x}) \equiv \cos(\theta)(f_2(\mathbf{x}) - e) - \sin(\theta) f_1(\mathbf{x}) \geq \\ \qquad\qquad a \left|\sin\left(b\pi \left(\sin(\theta)(f_2(\mathbf{x}) - e) + \cos(\theta) f_1(\mathbf{x})\right)^c\right)\right|^d . \end{array} \right. \quad (5)$$

The decision variable x_1 is restricted in $[0, 1]$ and the bounds of other variables depend on the chosen $g(\mathbf{x})$ function. The constraint has six parameters (θ, a, b, c, d, and e). In fact, the above problem can be used as a constrained test problem generator where constrained test problems with different complexities will evolve by simply tuning these six parameters. We demonstrate this by constructing different test problems, where the effect of each parameter is also discussed.

First, we use the following parameter values:

$$\theta = -0.2\pi, \quad a = 0.2, \quad b = 10, \quad c = 1, \quad d = 6, \quad e = 1.$$

The resulting feasible objective space is shown in Figure 6. It is clear from the figure that the unconstrained Pareto-optimal region becomes infeasible in the presence of the constraint. The periodic nature of the constraint boundary makes the Pareto-optimal region discontinuous, having a number of disconnected continuous regions. The task of an optimization algorithm would be to find as many such disconnected regions as possible. The number of disconnected regions can be controlled by increasing the value of the parameter b. It is also clear that with the increase in number of disconnected regions, an algorithm will have difficulty in finding representative solutions in all disconnected regions.

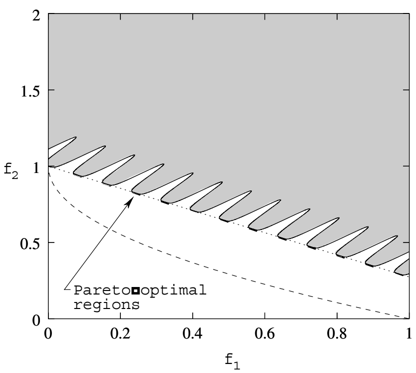

Fig. 6. Constrained test problem CTP2.

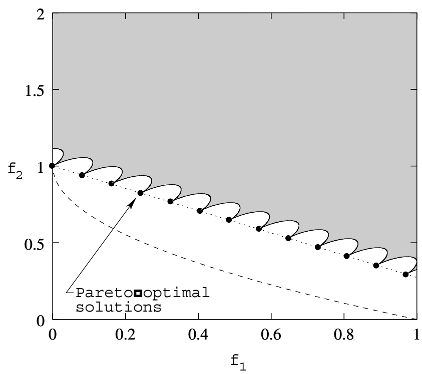

Fig. 7. Constrained test problem CTP3.

The above problem can be made more difficult by using a small value of d, so that in each disconnected region there exists only one Pareto-optimal solution. Figure 7 shows the feasible objective space for $d = 0.5$ and $a = 0.1$ (while other parameters are the same as that in the previous test problem). Although most of the feasible search space is continuous, near the Pareto-optimal region, the feasible search regions are disconnected, finally each subregion leading to a singular feasible Pareto-optimal solution. An algorithm will face difficulty in finding all discrete Pareto-optimal solutions because of the changing nature from continuous to discontinuous feasible search space near the Pareto-optimal region.

The problem can have a different form of complexity by increasing the value of parameter a, which has an effect of making the transition from continuous to discontinuous feasible region far away from the Pareto-optimal region. Since an algorithm now has to travel through a long narrow feasible *tunnel* in search of the lone Pareto-optimal solution at the end of tunnel, this problem will be more difficult to solve compared to the previous problem. Figure 8 shows one such problem with $a = 0.75$ and rest of the parameters same as that in the previous test problem.

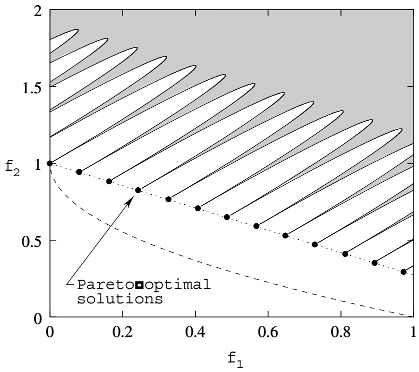

Fig. 8. Constrained test problem CTP4.

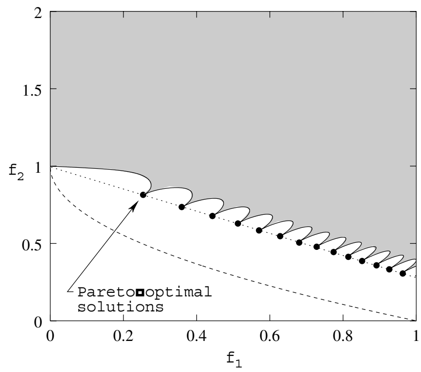

Fig. 9. Constrained test problem CTP5.

In all the three above problems, the disconnected regions are equally distributed in the objective space. The discrete Pareto-optimal solutions can be scattered non-uniformly by using $c \neq 1$. Figure 9 shows the feasible objective space for a problem with $c = 2$. Since $c > 1$, more Pareto-optimal solutions lie towards right (higher values of f_1). If, however, $c < 1$ is used, more Pareto-optimal solutions will lie towards left. For even higher number of Pareto-optimal solutions towards right, a larger value of c can be chosen. The difficulty will arise in finding all closely packed discrete Pareto-optimal solutions.

The parameters θ and e do not have a major role to play in terms of producing a significant complexity. The parameter θ controls the slope of the Pareto-optimal region, whereas the parameter e shifts the constraints up or down in the objective space. For the above problems, the Pareto-optimal solutions lie on the following straight line:

$$(f_2(\mathbf{x}) - e)\cos\theta = f_1(\mathbf{x})\sin\theta. \qquad (6)$$

It is interesting to note that the location of this line is independent of other major parameters (a, b, c, and d). The above equation reveals that the parameter e denotes the intercept of this line on the f_2 axis. The corresponding intercept on the f_1 axis is $-e/\tan\theta$.

It is important here to mention that although the above test problems will cause difficulty in the vicinity of the Pareto-optimal region, an algorithm has to maintain an adequate diversity much before it comes closer to the Pareto-optimal region. If an algorithm approaches the Pareto-optimal region without much diversity, it may be too late to create diversity among population members, as the feasible search region in the vicinity of the Pareto-optimal region is disconnected.

4.2 Difficulty in the Entire Search Space

Above test problems provide difficulty to an algorithm in the vicinity of the Pareto-optimal region. Difficulties may also come from the infeasible search regions present in the entire search space. Fortunately, the same constrained test problem generator can also be used here.

Figure 10 shows the feasible objective search space for the following parameter values:

$$\theta = 0.1\pi, \quad a = 40, \quad b = 0.5, \quad c = 1, \quad d = 2, \quad e = -2.$$

Notice that the above parameter values are very different from that used in the previous section. Particularly, the parameter value of a is two-order magnitude larger than before.

The objective space of this function has infeasible holes of differing widths towards the Pareto-optimal region. Since an algorithm has to overcome a number of such infeasible holes before coming to the island containing the Pareto-optimal front, an algorithm may face difficulty in solving this problem. Moreover, the unconstrained Pareto-optimal region is now not feasible. The entire constrained Pareto-optimal front lies on a part of the constraint boundary. (In this particular test problem, the Pareto-optimal region corresponds to all solutions satisfying $1 \leq ((f_2 - e)\sin\theta + f_1 \cos\theta) \leq 2$.) The difficulty can be increased further by widening the infeasible regions (or by using a small value of d).

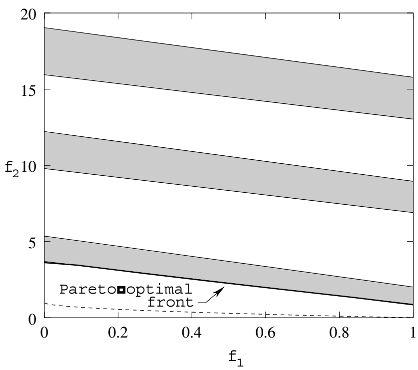

Fig. 10. Constrained test problem CTP6.

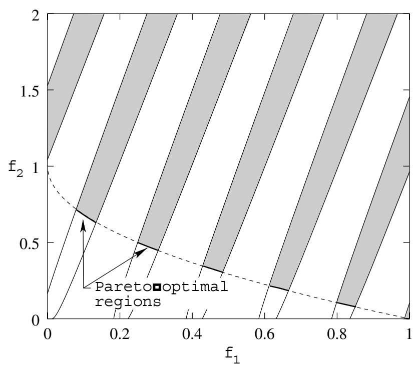

Fig. 11. Constrained test problem CTP7.

The infeasibility in the objective search space can also come along the Pareto-optimal region. Using the following parameter values

$$\theta = -0.05\pi, \quad a = 40, \quad b = 5, \quad c = 1, \quad d = 6, \quad e = 0,$$

we obtain the feasible objective space, as shown in Figure 11. This problem makes some portions of the unconstrained Pareto-optimal region feasible, thereby making a disconnected set of continuous regions. In order to find all such disconnected regions, an algorithm has to maintain an adequate diversity right from the beginning of a simulation run. Moreover, the algorithm also has to maintain its solutions feasible as it proceeds towards the Pareto-optimal region.

Once again, the difficulty can be increased by reducing the width of the feasible regions by simply using a smaller value of d. A careful thought will reveal that this test problem is similar to CTP4, except that a large value of the parameter a is now used. This initiates the disconnectedness in the feasible region far away from the Pareto-optimal region. If it initiates away from the search region dictated by the variable boundaries, the entire search region becomes a patch of feasible and infeasible regions.

With the above constraints, a combination of two or more effects can be achieved together in a problem by considering more than one such constraints.

4.3 More Than Two Objectives

Using the above concept, test problems having more than two objectives can also be developed. We modify equation 5 as follows. Using an M-dimensional transformation (rotational R and translational $\mathbf{e}$), we compute

$$\mathbf{f}' = R^{-1}(\mathbf{f} - \mathbf{e}). \tag{7}$$

The matrix R will involve $(M-1)$ rotation angles. Thereafter, the following constraint can be used:

$$f'_M \geq \sum_{j=1}^{M-1} a_j \left| \sin\left(b_j \pi {f'}_j^{c_j}\right)\right|^{d_j}. \tag{8}$$

Here, all a_j, b_j, c_j, d_j, and θ_j are parameters that must be set to get a desired effect. Like before, a combination of more than one such constraints can also be considered.

5 Simulation Results

We compare NSGA-II with the constrained-domination principle, Jimenez's algorithm and Ray et al.'s algorithms on test problems CTP1 till CTP7 with Rastrigin's function as the g functional.

In all algorithms, we have used the simulated binary crossover [4] with $\eta_c = 20$ and polynomial mutation operator [4] with $\eta_m = 20$, exactly the same as that used with NSGA-II. Identical recombination and mutation operators are used to investigate the effect of constraint handling ability of each algorithm. A crossover probability of 0.9 and mutation probability of $1/n$ (where n is the number of variables) are chosen. In all problems, five decision variables[1] are used. A population of size 100 and a maximum generation of 500 are used. For the Ray et al.'s algorithm, we have used a sharing parameter $\sigma_{share} = 0.158$. For Jimenez's algorithm, we has used $t_{dom} = 10$ and $\sigma_{share} = 0.158$. It is important to highlight that NSGA-II does not require any extra parameter setting.

In all problems, the Jimenez's algorithm performed poorly by not converging any where near the true Pareto-optimal region. Thus, we do not present the solutions using Jimenez's algorithm.

Figure 12 shows that both NSGA-II and Ray et al.'s algorithm are able to converge close to the true Pareto-optimal front on problem OSY, but NSGA-II's convergence ability as well as ability to find diverse solutions are better.

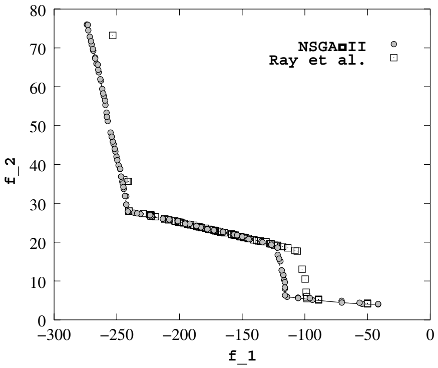

Fig. 12. Simulation results on OSY.

Figure 13 shows the population after 500 generations with NSGA-II and Ray et al.'s algorithm on CTP2. NSGA-II is able to find all disconnected Pareto-optimal solutions, whereas Ray et al.'s algorithm could not converge well on all disconnected regions.

Problem CTP3 degenerates to only one solution in each disconnected Pareto-optimal region. NSGA-II is able to find a solution very close to the true Pareto-optimal solution in each region (Figure 14). Ray et al.'s algorithm cannot quite converge near the Pareto-optimal solutions.

As predicted, problem CTP4 caused difficulty to both algorithms. Figure 15 shows that both algorithm could not get near the true Pareto-optimal solutions.

The non-uniformity in spacing of the Pareto-optimal solutions seems to be not a great difficulty to NSGA-II (Figure 16), but Ray et al's algorithm has some difficulty in converging to all Pareto-optimal solutions.

[1] As the simulation results show, only five variables caused enough difficulty to the three chosen MOEAs.

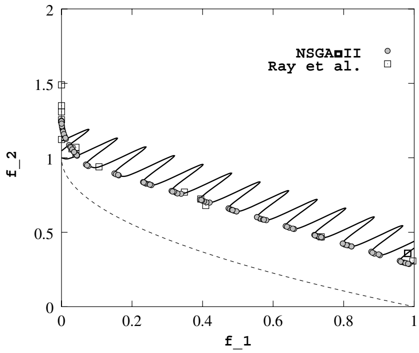

Fig. 13. Simulation results on CTP2.

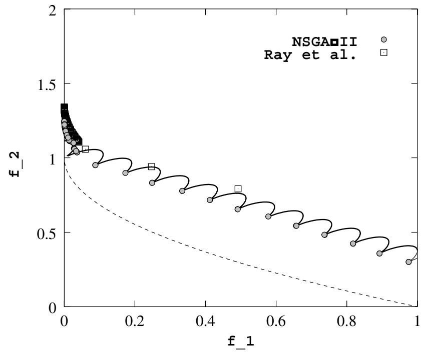

Fig. 14. Simulation results on CTP3.

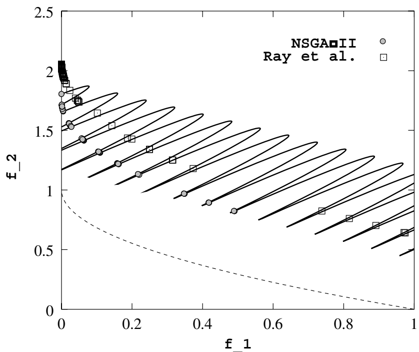

Fig. 15. Simulation results on CTP4.

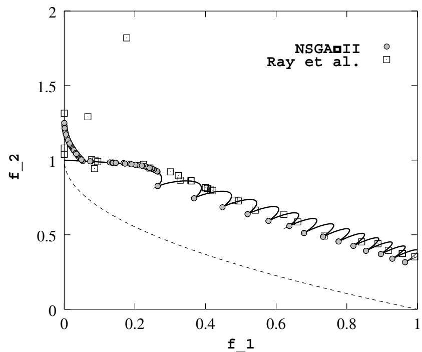

Fig. 16. Simulation results on CTP5.

When the entire search space consists of infeasible patches *parallel* to the Pareto-optimal front (CTP6 in Figure 17), both MOEAs are able to converge to the right feasible patch and finally very near to the true Pareto-optimal front. All feasible patches are marked with a 'F'. Although both algorithms performed well in this problem, NSGA-II is able to find a better convergence as well as a better spread of solutions.

However, when infeasible patches exist *perpendicular* to the Pareto-optimal front, both MOEAs had the most difficulty (Figure 18). None of them can find solutions closer to the true Pareto-optimal front. Although Ray et al's algorithm maintained a better spread of solutions, NSGA-II is able to come closer to the true front. We believe that this is one of the test problems where an algorithm must maintain a good spread of solutions from the beginning of the run and must also have a good converging ability. Algorithms which tend to converge anywhere in the Pareto-optimal front first and then work on finding a spread of

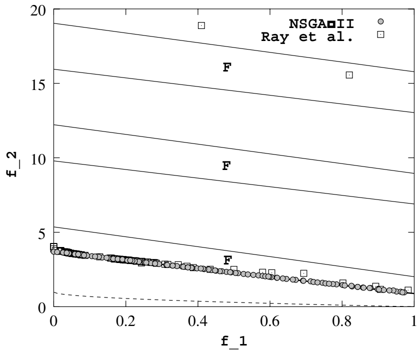

Fig. 17. Simulation results on CTP6.

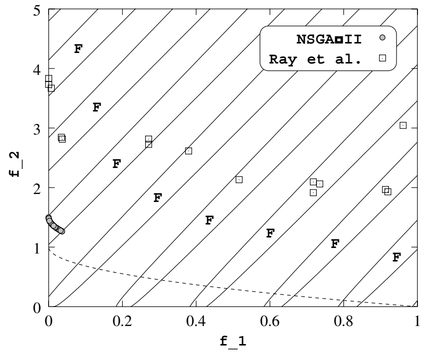

Fig. 18. Simulation results on CTP7.

solutions will end up finding solutions in a few of the feasible patches. NSGA-II shows a similar behavior in this problem.

6 Conclusion

In this paper, we have discussed the need of having tunable test problems for constrained multi-objective optimization. Test problems commonly used in most MOEA studies are simple and the level of complexity offered by the problems cannot be changed. We have presented a test problem generator which has six controllable parameters. By setting different parameter values, we have created a number of difficult constrained test problems. The difficulty in solving these test problems has been demonstrated by applying three constraint handling MOEAs on a number of these test problems. Although all algorithms faced difficulty in solving the test problems, NSGA-II with a constrained-domination principle has outperformed the other two MOEAs. The study has also shown that certain problem features can cause more difficulty than others.

Because of the tunable feature of the proposed test problems and demonstrated difficulties on a number of constrained MOEAs, we believe that these problems would be used as standard constrained test problems in the coming years of research in evolutionary multi-objective optimization.

Acknowledgements. Authors acknowledge the support provided by Department of Science and technology (DST) for this study.

References

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

2. Deb, K., Pratap, A., Moitra, S. (2000). Mechanical component design for multiple objectives using elitist non-dominated sorting GA. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 859–868.
3. Deb, K. (1999) Multi-objective genetic algorithms: Problem difficulties and construction of test Functions. *Evolutionary Computation, 7*(3), 205–230.
4. Deb, K. and Agrawal, R. B. (1995) Simulated binary crossover for continuous search space. *Complex Systems, 9* 115–148.
5. Fonseca, C. M. and Fleming, P. J. (1993) Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, Morgan Kauffman, San Mateo, California.
6. Horn, J. and Nafploitis, N., and Goldberg, D. E. (1994) A niched Pareto genetic algorithm for multi-objective optimization. In Michalewicz, Z., editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, IEEE Service Center, Piscataway, New Jersey.
7. Jiménez, F. and Verdegay, J. L. (1998). Constrained multiobejctive optimization by evolutionary algorithms. *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98)*, pp. 266–271.
8. Knowles, J. and Corne, D. (1999) The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway: New Jersey: IEEE Service Center, 98–105.
9. Osyczka, A. and Kundu, S. (1995). A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural Optimization*(10). 94–99.
10. Ray, T., Kang, T., and Chye, S. (in press). Multiobjective design optimization by an evolutionary algorithm, *Engineering Optimization*.
11. Srinivas, N. and Deb, K. (1995). Multi-Objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*(2), 221–248.
12. Tanaka, M. (1995). GA-based decision support system for multi-criteria optimization. *Proceedings of the International Conference on Systems, Man and Cybernetics-2*, pp. 1556–1561.
13. Van Veldhuizen, D. (1999). Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. *PhD Dissertation and Technical Report No. AFIT/DS/ENG/99-01*, Dayton, Ohio: Air Force Institute of Technology.
14. Zitzler, E. (1999). Evolutionary algorithms for multiobjective optimization: Methods and applications. Doctoral thesis ETH NO. 13398, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Germany: Shaker Verlag.