A Constraint Handling Strategy for Robust Multi-Criteria Optimization

Kalyanmoy Deb and Himanshu Gupta

Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208016, India
{deb,himg}@iitk.ac.in

KanGAL Report Number 2005001

Abstract. Robust multi-criteria optimization has emerged as an active research area in the past few years. A recent study proposed two different definitions of robust solutions in the context of multi-objective optimization. In this paper, we extend the concepts for finding robust solutions in the presence of active constraints. The meaning of robust solutions for constrained problems is demonstrated by suggesting three test problems and simulating an evolutionary multi-objective optimization method using the two definitions of robustness. The inclusion of constraints makes the multi-objective robust optimization procedure more pragmatic and the procedure is now ready to be applied to real-world problems.

1 Introduction

For the past decade or more, the primary focus of the research and application in the area of evolutionary multi-criterion optimization (EMO) has been placed in finding the globally best Pareto-optimal solutions. Such solutions are non-dominated to each other and there exists no other solution in the entire search space which dominates these solutions. Recently a lot of interest has developed for searching robust solutions. These solutions are relatively insensitive to the perturbations in variable space. Finding robust solutions is of immense importance because in a real world scenario it may not be possible to implement the obtained Pareto-optimal solution precisely. If the objectives are highly sensitive to perturbation in variable space the performance obtained may be quite degraded in comparison to the performance of obtained Pareto-optimal solutions.

Already there have been a lot of studies towards developing the robust optimization strategies both in single objective as well as multi-objective case. Branke [1–3] describes the issues in single objective robust optimization. Jin and Sendhoff [4] considers the issue of finding robust solutions in a single-objective optimization problem as a multi-objective optimization problem with the objectives being maximizing robustness and performance. Tsutsi and Ghosh [5] presented a mathematical model for obtaining robust solutions using the schema theorem for single-objective genetic algorithms. Parmee [6] suggest a hierarchical strategy of searching several high performance regions in a fitness landscape simultaneously.

Teich [7] discusses Pareto-front exploration with uncertain objectives. Hughes [8] discusses Evolutionary Multi-objective Ranking with Uncertainty and Noise. Recently Deb and Gupta [9] discussed various issues in robust multi-criteria optimization. They also described two strategies for finding robust solutions. They discuss four cases that can result from the manner robust frontier moves in the search space.

All these schemes suggested above have not taken into account the effect of presence of constraints in the optimization process. Real-world problems generally include a number of constraints. So unless the strategies for handling constraints are developed the overall robust multi-criteria optimization procedure remains incomplete. In this paper, we present a constraint-handling scheme. We inherit the model developed in an earlier study [9] for unconstrained multi-criteria optimization. We show that the inclusion of constraints results in a much greater shift of robust frontier, as the solutions on a constraint boundary (albeit optimal) are not robust. The proposed constraint handling strategy has been incorporated with both the approaches described in [9]. Some test problems having constraints are developed and the results are shown. Differences between unconstrained and constrained robust multi criteria optimization are clearly outlined.

Rest of the paper is designed as followed. Section 2 introduces some definitions and introduces constraint handling strategy for robust multi-criteria optimization. Section 3 describes the test-problems. Section 4 and 5 show the simulation results. Section 6 concludes with the future research directions.

2 Robust Optimization

Consider a multi-objective optimization problem as:

$$\left.\begin{array}{l}\text{Minimize } (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})),\\ \text{subject to } \mathbf{x} \in \mathcal{S},\end{array}\right\} \quad (1)$$

Deb and Gupta[9] defined following two approaches for robust optimization:

Definition 1. (Multi-objective Robust Solution of Type I): *A solution $\mathbf{x}^*$ is called a multi-objective robust solution of type I if it is the global feasible Pareto-optimal solution to the following multi-objective minimization problem (defined with respect to a δ-neighborhood ($\mathcal{B}_\delta(\mathbf{x})$ of a solution $\mathbf{x}$):*

$$\left.\begin{array}{l}\textit{Minimize } (f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), \ldots, f_M^{\text{eff}}(\mathbf{x})),\\ \textit{subject to } \mathbf{x} \in \mathcal{S},\end{array}\right\} \quad (2)$$

where $f_j^{\text{eff}}(\mathbf{x})$ is defined as follows:

$$f_j^{\text{eff}}(\mathbf{x}) = \frac{1}{|\mathcal{B}_\delta(\mathbf{x})|}\int_{\mathbf{y}\in\mathcal{B}_\delta(\mathbf{x})} f_j(\mathbf{y})d\mathbf{y}. \quad (3)$$

Definition 2. (Multi-objective Robust Solution of Type II): *A solution* $\mathbf{x}^*$ *is called a multi-objective robust solution of type II if it is the global feasible Pareto-optimal solution to the following multi-objective minimization problem:*

$$\left.\begin{array}{l} \textit{Minimize}\ \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \textit{subject to}\ \frac{\|\mathbf{f}^{\text{eff}}(\mathbf{x})-\mathbf{f}(\mathbf{x})\|}{\|\mathbf{f}(\mathbf{x})\|} \leq \eta, \\ \mathbf{x} \in \mathcal{S}. \end{array}\right\} \quad (4)$$

Definition 1 requires an optimization of effective objective function values computed as a mean of the function values in the vicinity of a solution. The Pareto-optimal frontier thus obtained is called a robust Pareto-frontier. Definition 2 requires original objectives to be optimized, but makes a solution feasible only when the extent of change in function values among neighboring solutions is limited to a user-defined parameter η. That study discussed various pros and cons of the two approaches. Here, we use both definitions of robustness, but modify them for handling constrained optimization problems.

3 Constrained Robust Optimization

Definition 3. (Robust Feasible Optimal Solution of Type I) *A solution* $\mathbf{x}^*$ *is called a robust feasible optimal solution of type I if it is a feasible global optimal solution and all solutions in its* δ*-neighborhood is also feasible:*

$$\left.\begin{array}{l} \textit{Minimize}\ (f_1^{\text{eff}}(\mathbf{x}), f_2^{\text{eff}}(\mathbf{x}), \ldots, f_M^{\text{eff}}(\mathbf{x})), \\ \textit{subject to}\ \mathbf{y} \in \mathcal{S}, \quad \textit{for all}\ \mathbf{y} \in \mathcal{B}_\delta(\mathbf{x}), \end{array}\right\} \quad (5)$$

where $\mathcal{B}_\delta$ *is the* δ*-neighborhood of the solution.*

The inclusion of above constraint ensures that the solutions lying on an active constraint boundary will not be robust, as a perturbed solution is likely to be infeasible. In a true sense, solutions lying on a constraint boundary may be optimal, but is most likely to be a non-robust solution, as some minor perturbation in the solution will make the solution infeasible. In our implementation, instead of checking all neighboring solutions to be feasible, we shall check the feasibility of each solution used to compute effective function values.

Similarly, we modify the second definition as follows:

Definition 4. Robust Feasible Optimal Solution of Type II: *For the minimization of a multi-objective problem, a solution* $\mathbf{x}^*$ *is called a robust feasible optimal solution of type II, if it is the Pareto-optimal solution to the following problem:*

$$\left.\begin{array}{l} \textit{Minimize}\ (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \\ \textit{subject to}\ \frac{\|\mathbf{f}^{\text{eff}}(\mathbf{x})-\mathbf{f}(\mathbf{x})\|}{\|\mathbf{f}(\mathbf{x})\|} \leq \eta, \\ \mathbf{y} \in \mathcal{S}, \quad \textit{for all}\ \mathbf{y} \in \mathcal{B}_\delta(\mathbf{x}). \end{array}\right\} \quad (6)$$

The inclusion of the additional constraint ensures that all solutions in the vicinity of a robust feasible optimal solution are feasible.

4 Test-Problems

To investigate the effect of constraints on robust solutions, we suggest the following test problem:

$$\begin{aligned}
\text{Minimize } & f_1(\mathbf{x}) = x_1, \\
\text{Minimize } & f_2(\mathbf{x}) = h(x_1) + G(\mathbf{x})S(x_1), \\
\text{Subject to } & g(x) \geq 0, \\
& 0 \leq x_1 \leq 1, \\
& -1 \leq x_i \leq 1, \quad i = 2, 3, \ldots, n, \\
\text{where } & h(x_1) = 1 - x_1^2, \\
& G(\mathbf{x}) = \textstyle\sum_{i=2}^{n} 50x_i^2, \\
& S(x_1) = \frac{\alpha}{0.2+x_1} + \beta x_1^2.
\end{aligned} \tag{7}$$

We construct three test problems by using the following functions for $g(\mathbf{x})$ and for $\alpha = \beta = 1$:

Test Problem 1: $g(x) = 0.2x_1 + x_2 - 0.1$,
Test Problem 2: $g(x) = \sin(32x_1)$,
Test Problem 3: $g(x) = 4{f_1}^2 + \frac{{f_2}^2}{2}$.

5 Simulation Results

First, we use definition 1 of making a solution a robust feasible optimal solution. We incorporate the above mentioned constraint handling strategy with both the approaches in NSGA-II [10]. The additional constraint is handled using the constrained-domination principle [11]. Here, the main design parameters are the extent of neighborhood (δ) and the number of neighboring points (H) used to compute the mean effective objectives and the feasibility of a solution. In the following subsections, we describe the effect of these parameters. Throughout this paper, we have used the following nomenclatures:

- **Original Front** The front obtained without any robust consideration, but with considering the original constraints.
- **Simple Effective Front** The front obtained with robustness consideration, but without caring the original constraints.
- **Constrained Effective Front** The front obtained with robustness definitions and with consideration of original constraints.

5.1 Effect of Number of Neighboring Points (H) with Definition 1

As mentioned earlier, definition 1 requires optimizing effective objective values. To compute effective objective function values and check the feasibility of a solution $\mathbf{x}$, H different points in the δ-neighborhood of $\mathbf{x}$ are computed and their average is taken. These H points are chosen as described in a systematic manner, as described in the earlier study [9]. To create a pattern systematically, perturbation domain of each variable (around $[-\delta_i, \delta_i]$) is divided into exactly H equal grids, thereby dividing the δ-neighborhood into n^H small hyperboxes. Thereafter, exactly H hyperboxes are picked randomly from n^H hyperboxes so that in each dimension all H distinct grids are represented. In all simulations here, we use the simulated binary crossover (SBX) and the polynomial mutation operator with distribution indices of 10 and 50, respectively [11]. A population size of 100 is run for a long enough (1,000) generations to have confidence in the location of the robust optimal front.

For the test problem without any constraint, expressions for the effective objective functions can be written as:

$$f_1^{\text{eff}}(\mathbf{x}) = x_1, \tag{8}$$

$$f_2^{\text{eff}}(\mathbf{x}) = (1-x_1^2) - \frac{1}{3}\delta_1^2 + \left[\alpha \frac{1}{2\delta_1}\log\left(\frac{0.2+x_1+\delta_1}{0.2+x_1-\delta_1}\right) + \beta\left(x_1^2 + \frac{1}{3}\delta_1^2\right)\right]\sum_{i=2}^{n}\left(\frac{50}{3}\delta_i^2\right). \tag{9}$$

Test Problem 1 The constraint $g(\mathbf{x})$ suggests that for $x_1 \geq 0.5$ any value of x_2 would make the solution feasible. For $x_1 < 0.5$, however, the following relationship must be true for a feasible solution:

$$x_2 \geq 0.1 - 0.2x_1. \tag{10}$$

A closer look at the test problem and the constraint function will reveal that $x_i = 0$ for $i \geq 3$ and the constrained original front can be written in following parametric form:

$$f_1 = x_1,$$
$$f_2 = \begin{cases} 1 - x_1^2 + 50(0.1-0.2x_1)^2(1/(0.2+x_1)+x_1^2), & \text{if } x_1 < 0.5, \\ 1 - x_1^2, & \text{if } x_1 \geq 1. \end{cases} \tag{11}$$

This front is marked as 'constrained original front' in Figure 1. Now, for robustness consideration, we perturb variables x_i with δ_i, thereby making the following relationship among x_2 and x_1 for robust feasible optimal solutions having $x_1 < 0.5$:

$$x_2 \geq 0.1 - 0.2(x_1 - \delta_1) + \delta_2 \tag{12}$$

So the solutions corresponding to constrained effective front should satisfy equation 12, making x_2 not a free variable. Thus, the effective objective functions for test problem 1 in the range $x_1 < 0.5$ can be written as follows:

$$f_1^{\text{eff}}(\mathbf{x}) = x_1, \tag{13}$$

$$f_2^{\text{eff}}(\mathbf{x}) = (1-x_1^2) - \frac{1}{3}\delta_1^2 + \left[\alpha \frac{1}{2\delta_1}\log\left(\frac{0.2+x_1+\delta_1}{0.2+x_1-\delta_1}\right) + \beta\left(x_1^2 + \frac{1}{3}\delta_1^2\right)\right]\sum_{i=3}^{n}\left(\frac{100}{3}\delta_i^2\right)$$

$$+\frac{50}{2\delta_1}\int_{x_1-\delta_1}^{x_1+\delta_1}(\frac{\alpha}{0.2+x_1}+\beta x_1^2)(0.1-0.2x_1+0.2\delta_1+\delta_2)^2 dx_1. \quad (14)$$

Equation 8, 9, 13 and 14 together define the constrained robust optimal frontier. In order to obtain the simple effective frontier (effective function values are optimized, but solutions themselves are checked for feasibility), the term $(0.1-0.2x_1+0.2\delta_1+\delta_2)$ must be replaced by $(0.1-0.2x_1)$.

These theoretical corresponding frontiers (constrained original front, simple effective front and constrained effective front) of test problem 1 are shown in Figure 1. Here, we have chosen $\delta_1 = 0.01$ and $\delta_i = 2\delta_1$ for $i \geq 2$. In Figure 2, we show the obtained NSGA-II solutions with $H = 5$ and $H = 100$ neighboring points. The effect of H is clear from the plot. As the number of neighboring points are increased, the obtained solutions get closer to the theoretical frontier (which can be viewed as a robust optimal frontier with $H = \infty$).

To investigate the effect of robustness due to constraints, we compute the robust constraint violation (RCV) of each solution $\mathbf{x}$, defined as follows:

$$\mathrm{RCV}(\mathbf{x}) = \sum_{\mathbf{y}\in\mathcal{B}_\delta(\mathbf{x})} \mathrm{CV}(\mathbf{y}), \quad (15)$$

where, the constraint violation of a solution $\mathbf{y}$ is defined as follows:

$$\mathrm{CV}(\mathbf{y}) = \sum_j \langle g_j(\mathbf{y})\rangle, \quad (16)$$

where the operator $\langle\gamma\rangle$ is defined as follows:

$$\langle\gamma\rangle = \begin{cases} -\gamma \text{ if } \gamma < 0, \\ 0 \quad \text{otherwise.} \end{cases}$$

Thus, if the robust constraint violation is negative, some neighboring solutions used for mean effective objective computation is infeasible. The RCV values (computed using new 1,000 neighboring points) are plotted for all obtained solutions in Figure 3. It is interesting to note that for a fewer neighboring points (H), constraint violation is more. For $H = 100$ points, the constraint violation is zero. The effect of robustness is also clear from Figure 4, which shows the corresponding x_1-x_2 variation. The solutions of the simple effective frontier satisfies the original constraint ($x_2 + 0.2x_1 - 0.1 \geq 0$). With more neighboring points, the solutions move away from these boundary solutions and fall near the theoretical constraint effective solutions. It is intuitive that the extent of movement from simple effective front to the constrained effective front will depend on the chosen δ_i values.

Test Problem 2 Test Problem 2 presents an example of search space which is piece-wise feasible. We have the constraint: $sin(32x_1) \geq 0$. So the feasible region corresponds to

$$2n\pi/32 < x_1 < (2n+1)\pi/32, \quad n = 0, 1, 2, 3, 4. \quad (17)$$

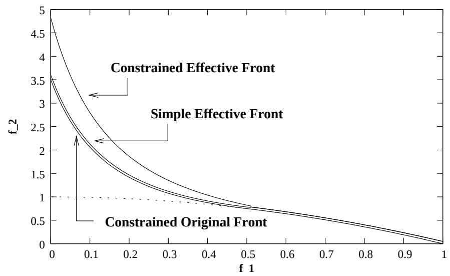

Fig. 1. Theoretical simple and constrained effective fronts on test problem 1.

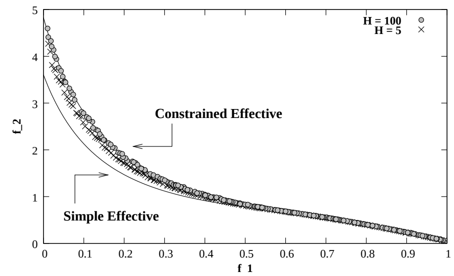

Fig. 2. Constrained effective fronts showing the effect of number of points H on test problem 1.

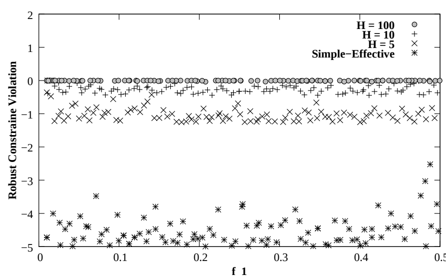

Fig. 3. Robust constraint violation for the points obtained on test problem 1 with different values of H.

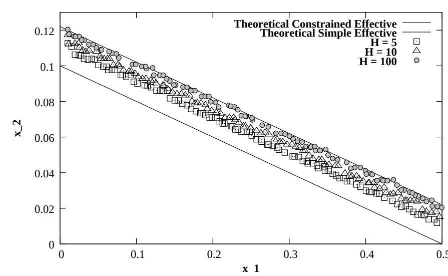

Fig. 4. x_1 vs x_2 relationship for different values of H on test problem 1.

Since for robustness x_1 would be evaluated in a neighborhood δ_1, we must have:

$$2n\pi/32 + \delta_1 < x_1 < (2n+1)\pi/32 - \delta_1, \quad n = 0, 1, 2, 3, 4. \tag{18}$$

The effect of variation of H is shown in Figure 5 and 6. A close look at Figure 5 reveals that the constraint makes the frontier disjointed. In this problem, the simple effective frontier and the constraint effective frontier are more or less same. As in test problem 1, we again compute the robust constraint violation of the solutions obtained for constrained effective fronts using 1,000 points. Variation of robust constraint violation is shown in Figure 6. It can be seen that for a large enough value of H (= 50) the robust constraint violation is very close to zero. As in this problem the constraint is only in one variable (x_1) only corner points in each strip are violating the constraint. It can be seen that the corner solutions in simple effective fronts are highly infeasible with respect to robust constrain violation. With $H = 10$ robust-constrain-violation is very small when compared with solutions in simple effective frontier solutions. It shows that the above mentioned constraint handling strategy is indeed able to find robust feasible solutions.

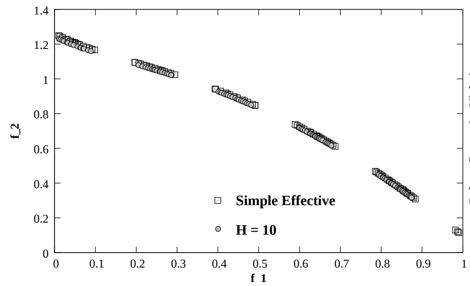

Fig. 5. Constrained Effective fronts showing the effect of number of points H on test problem 2.

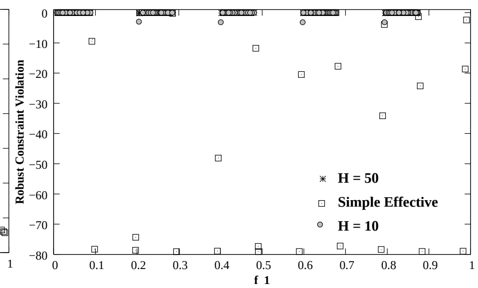

Fig. 6. Robust Constraint Violation for the points obtained on test problem 2 with different values of H.

Test Problem 3 Test problem 3 represents a case where the constraint is formed using objective values directly. Effect of variation of H on the obtained front is shown in Figure 7. Figure 8. The solid line represents the original front. Simple effective front (computed with $H = 50$) is also shown. It can be seen that constrained effective front with $H = 10$ lies above than simple effective front with $H = 50$. The constrained effective front with $H = 50$ lies even above and approaches the theoretical effective front. The robust constraint violation for the solutions is shown in Figure 8. It is observed that with increasing value of H, the RCV value decreases.

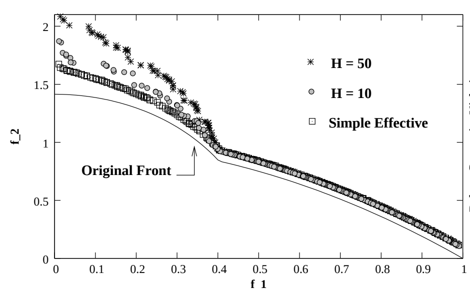

Fig. 7. Constrained effective fronts showing the effect of number of points H on test problem 3.

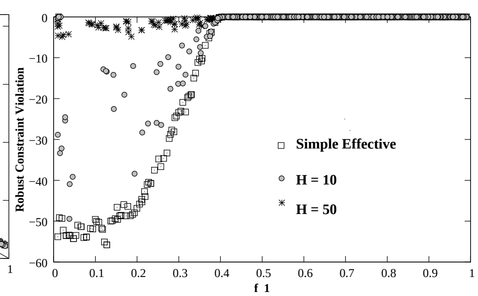

Fig. 8. Robust constraint violation for the points obtained on test problem 3 with different values of H.

5.2 Effect of δ with Approach 1

The variation of the simple effective front with δ has been extensively discussed in an earlier study [9]. Variation of constrained effective front with H is discussed above. Here, we discuss the effect for a change in δ_i values. With the increase in δ_i, the shift in constrained effective front from the original front increases as shown in Figure 9 for test problem 2. Theoretical variation of boundary of constrained effective front with δ is also shown and is found to be matching with the theoretical results.

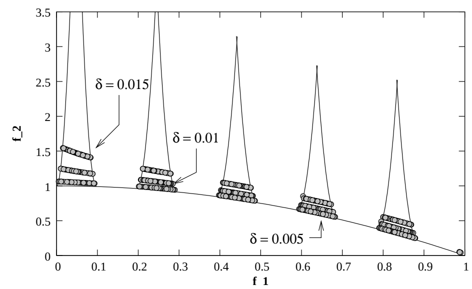

Fig. 9. Constrained effective fronts showing the effect of delta δ on test problem 2.

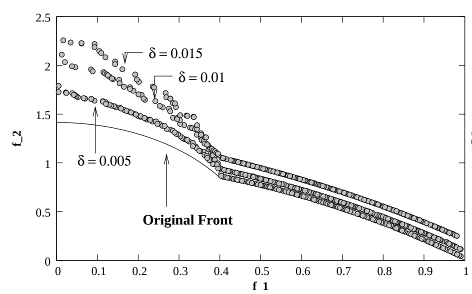

Fig. 10. Constrained effective fronts showing the effect of delta δ on test problem 2.

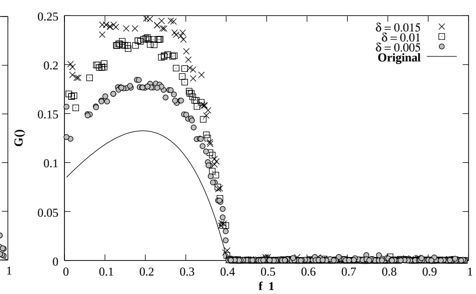

Fig. 11. Effect on Variable Space for different values of δ on test problem 3.

It is clear that the solutions for small values of f_1 are more sensitive to neighborhood size. The boundary lines indicate that with an increase of δ_i, the range of robust frontier decreases. After a certain δ vector, the perturbation is so large that no solution is found to be robust.

Figure 10 shows the effect of neighborhood size on test problem 3. With the increase in δ_i, the effective frontier moves away from the simple effective front. All the above results indicate that NSGA-II with a simple constraint handling procedure is able to find the resulting effective robust frontier in different scenarios.

6 Constrained Robust Multi-criteria optimization with Definition 2

As discussed earlier, in definition 2, the original objectives are optimized but a constraint is added to limit the extent of change in function values. This approach requires a user-defined parameter η. Various advantages of this approach over the previous approach are described in [9].

Incorporation of the above mentioned constraint handling strategy with definition 2 is straightforward. Figure 12 and Figure 14 show the effect of variation of η on test problems 2 and 3, respectively. With a decreasing value of η (tighter requirement for a solution to be defined as robust), the resulting robust frontier becomes more different than the original constrained front.

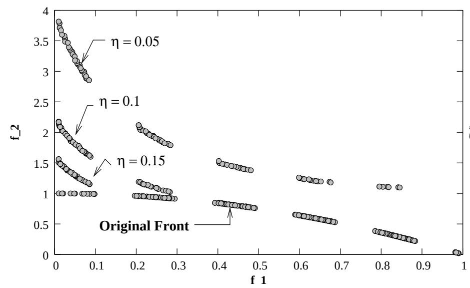

Fig. 12. Constrained effective fronts showing the effect of η on test problem 2.

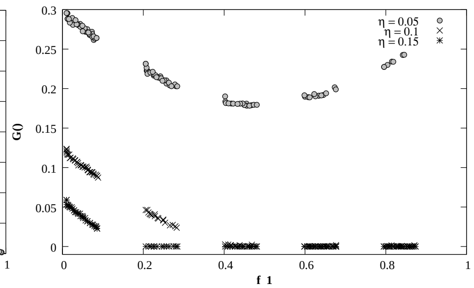

Fig. 13. Effect of η on G() function for test problem 2.

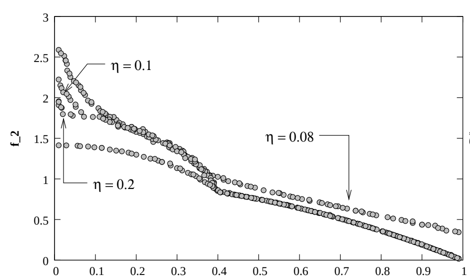

Fig. 14. Constrained effective fronts showing the effect of η on test problem 3.

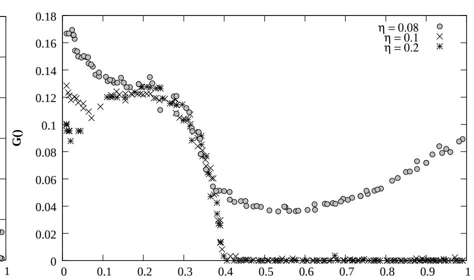

Fig. 15. Effect of η on G() function for test problem 3.

7 Conclusions and Future Work

This paper suggests a couple of constraint-handling strategies for robust multi-criteria optimization. This strategies are extensions of an earlier robust multi-criteria optimization strategies [9]. It is argued that solutions on simple effective front (without robustness consideration) are no longer feasible, because a perturbation to these solutions may produce an infeasible solution. Thus, ideally a constrained effective robust front would lie somewhat away from the simple effective solutions.

The efficacy of the two procedures have been demonstrated on three test problems. In some cases, the NSGA-II robust solutions are found to lie close to the theoretical robust frontiers. The effect of three parameters (neighborhood size, number of neighboring solutions, and limiting robustness) on the extent of movement of the frontiers has also been shown.

Constraints are inevitable in real-world optimization problems. The techniques of this paper, along with the original robust multi-objective optimization study, should now stand as a complete procedure for finding robust Pareto-optimal frontiers in real-world problems.

As an extension to this study, we are currently pursuing a study to find the effect of location of H neighboring points in computing the mean effective objective values. Although in this study we have used a systematic procedure of spreading the solutions across the entire δ-neighborhood, there is some merit in choosing the solutions exactly on the δ-boundary for constraint violation computation. In constraint robust optimization, the latter makes sense, as if a δ-boundary point is feasible, it can be assumed that the interior of the δ-neighborhood is also feasible in most problems. Also, the use of an archive to store already-computed solutions would be another avenue for future research.

References

1. Jurgen Branke. Efficient evolutionary algorithms for searching robust solutions. *ACDM*, pages 275–286, 2000.
2. Jurgen Branke and C. Schmidt. Faster convergence by means of fitness estimation. In *Soft Computing*, 2000.
3. Jurgen Branke. Creating robust solutions by means of an evolutionary algorithm. *Parallel Problem Solving from Nature*, pages 119–128, 1998.
4. Yaochu Jin and Bernhard Sendhoff. Trade-off between performance and robustness: An evolutionary multiobjective approach. In *EMO2003*, pages 237–251, 2003.
5. S. Tsutsui and A. Ghosh. Genetic algorithms with a robust solution searching scheme. *IEEE transactions on Evolutionary Computation*, pages 201–219, 1997.
6. I.C. Parmee. The maintenance of search diversity for effective design space decomposition using cluster-oriented genetic algorithms(cogas) and multi-agent strategies(gaant). In *ACEDC*, 1996.
7. Jurgen Teich. Pareto-front exploration with uncertain objectives. In *EMO2001*, 2001.
8. Hughes E. Evolutionary multi-objective ranking with uncertainty and noise. In *EMO2001*, 2001.

9. Kalyanmoy Deb and Himanshu Gupta. Searching for robust pareto-optimal solutions in multi objective optimization. In *Evolutionary Multi-Criteria Optimization*, 2005.
10. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm:nsga-2. *IEEE transactions on Evolutionary Computation*, pages 182–197, 2000.
11. K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.