Handling Constraints In Robust Multi-Objective Optimization

Himanshu Gupta and Kalyanmoy Deb
Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208016, India
Email: {himg,deb}@iitk.ac.in
URL: http://www.iitk.ac.in/kangal/pub.htm

Abstract- Robust multi-objective 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 constraint handling strategies makes the multi-objective robust optimization procedure more pragmatic and the procedure is now ready to be applied to real-world problems.

1 Introduction

Research in Evolutionary Multi-Criteria Optimization has concentrated on searching a set of diverse and non-dominated Pareto Optimal 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, 2, 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 we [9] discussed various issues in robust multi-objective optimization. We also described two strategies for finding robust solutions. We discussed 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-objective 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-objective 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 objective optimization are clearly outlined.

Rest of the paper is designed as followed. Section 2 and 3 introduce some definitions and constraint handling strategies for robust multi-objective optimization. Section 4 describes the test-problems. Section 5 and 6 show the simulation results. Section 7 concludes with the future research directions.

2 Robust Optimization

Consider a multi-objective optimization problem as:

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

For solving the above multi-objective optimization problem, an EMO procedure attempts to find a finite number of Pareto-optimal solutions, instead of a single optimum. Since Pareto-optimal solutions collectively *dominate* any other feasible solution in the search space, they all are considered to be better than any other solution [10] in the search space.

In Figure 1, two Pareto-optimal solutions (A and B) are checked for their sensitivity in the decision variable space. Since the local perturbation of point B causes a large change in objective values, this solution may not be a robust solution, whereas solution A which does not cause a large change in objective values due to a local perturbation in its vicinity, is a robust solution. To qualify as a robust solution, each Pareto-optimal solution now has to demonstrate its insensitivity towards small perturbations in its decision variable values. The main differences with a single-objective robust solution is that (i) the sensitivity now has to be established with respect to all M objectives. That is, a combined

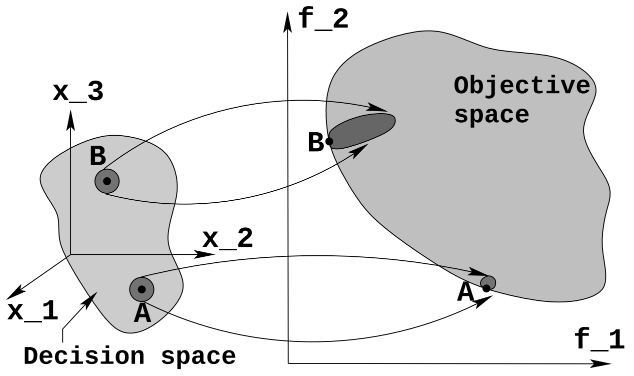

Figure 1: Point A is less sensitive to variable perturbation than point B.

effect of variations in all M objectives has to be used as a *measure* of sensitivity to variable perturbation, and (ii) there are many solutions to be checked for robustness.

We [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* $\mathbf{x}$*):*

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

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

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

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:*

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

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 relative 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.

In certain problem scenarios, the η-constraint can be modified to one of the following constraints:

- $\|\mathbf{f}^{\rm eff}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| \leq \eta$. The decision-maker may be interested in limiting the absolute difference between the perturbed and original objective vectors.
- $\frac{1}{M}\sum_{i=1}^{M}\frac{f_i^{\rm eff}(\mathbf{x})-f_i(\mathbf{x})}{f_i(\mathbf{x})} \leq \eta$. The decision-maker may be interested in limiting an average objective-wise normalized difference.
- $\frac{f_i^{\rm eff}(\mathbf{x})-f_i(\mathbf{x})}{f_i(\mathbf{x})} \leq \eta_i, \quad i = 1, 2, \ldots, M$. The decision-maker may be interested in limiting the normalized difference in each objective. However, this will require the decision-maker to supply a η-vector, instead of a single η value.
- $\max_{i=1,\ldots,M}\frac{f_i^{\rm eff}(\mathbf{x})-f_i(\mathbf{x})}{f_i(\mathbf{x})} \leq \eta$. This will allow the maximum normalized perturbation to be within limits.

3 Constrained Robust Optimization

Figure 2 explains the necessary for finding the robust constrained solution in a multi-objective optimization problem. On most interesting problems, a Pareto-optimal solution is likely to lie on the constraint boundary, as shown by the filled circles on the figure. Since the solutions lie on the constraint boundary, they are also precarious to be used in practice, particularly if the solutions are expected to be uncertain. In such cases, feasible solutions which are somewhat away from the constraint boundary turns out to be robust (like the solutions marked with open circles). The extent of movement of solutions from the constraint boundary will depend on the neighborhood size, chosen for performing the robust optimization. Similar to the definitions of the unconstrained robust solutions, we define in the following two definitions for a constrained robust solution.

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 are also feasible:*

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

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*

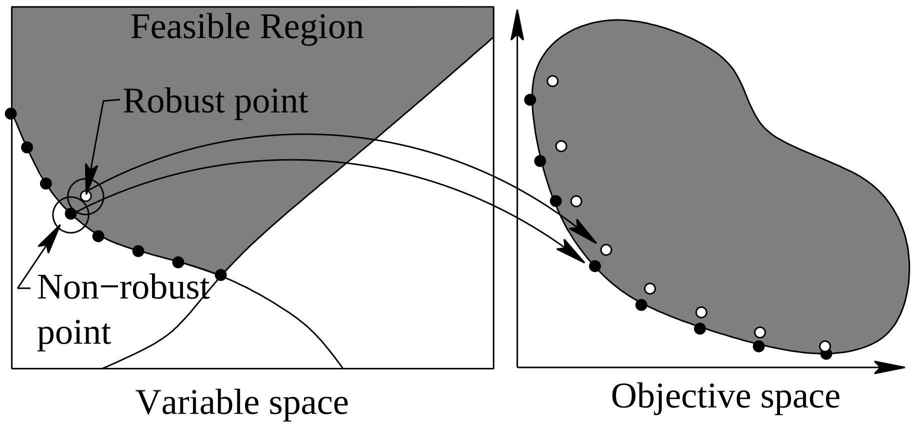

Figure 2: Illustration of a constrained robust solution.

solution $\mathbf{x}^*$ *is called a robust feasible optimal solution of type II, if it is the Pareto-optimal solution to the following problem:*

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

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 propose the following test problem and its variants:

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

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

Test Problem 1: $g(\mathbf{x}) = 0.2x_1 + x_2 - 0.1$,

Test Problem 2: $g(\mathbf{x}) = \sin(32x_1)$,

Test Problem 3: $g(\mathbf{x}) = 4{f_1}^2 + \frac{{f_2}^2}{2} - 1$.

Like ZDT test problems [10], the convergence towards the Pareto-optimal front is governed by $G()$ function and the shape of the Pareto-optimal front is governed by $S()$ function. However, the function $g()$ does not allow the complete original Pareto-optimal front to be feasible, thereby governing the shape and sensitivity of the Pareto-optimal front.

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 [11]. The additional constraint is handled using the constrained-domination principle [10]. 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 when multi-objective problem 1 is optimized. Robust criteria is neither applied on objective function values nor on the constraints.
- **Simple Effective Front** The front obtained with robustness consideration applied only on objectives. This front is obtained when the optimization is done using definition 1 and 2.
- **Constrained Effective Front** The front obtained with robustness considerations applied on both objectives as well as constraints. This type of front is obtained when the optimization is done using definition 3 and 4.

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 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 hyper-boxes. Thereafter, exactly H hyper-boxes are picked randomly from n^H hyper-boxes 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 [10]. 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:

$$f_1^{\rm eff}(\mathbf{x}) = x_1, \quad (8)$$

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

5.1.1 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:

$$x_2 \geq 0.1 - 0.2x_1. \quad (10)$$

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:

$$f_1 = x_1,$$
$$f_2 = \begin{cases} 1-x_1^2 \quad if \quad x_1 \geq 0.5 \quad else \\ 1-x_1^2+50(0.1-0.2x_1)^2(1/(0.2+x_1)+x_1^2) \end{cases} \quad (11)$$

This front is marked as 'constrained original front' in Figure 3. 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 $x_1 < 0.5$:

$$x_2 \geq 0.1 - 0.2(x_1-\delta_1) + \delta_2 \quad (12)$$

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:

$$f_1^{\rm eff}(\mathbf{x}) = x_1, \quad (13)$$

$$f_2^{\rm eff}(\mathbf{x}) = (1-x_1^2) - \frac{1}{3}\delta_1^2 + \left[\alpha\frac{1}{2\delta_1}\log\left(\frac{0.2+x_1+\delta_1}{0.2+x_1-\delta_1}\right) + \beta\left(x_1^2+\frac{1}{3}\delta_1^2\right)\right]\sum_{i=3}^{n}\left(\frac{100}{3}\delta_i^2\right) + \frac{50}{2\delta_1}\int_{x_1-\delta_1}^{x_1+\delta_1}\left(\frac{\alpha}{0.2+x_1}+\beta x_1^2\right)(0.1-0.2x_1+0.2\delta_1+\delta_2)^2 dx_1. \quad (14)$$

Equation 8, 9, 13 and 14 together define the constrained effective frontier. In order to obtain the simple effective frontier (effective function values are optimized, but only 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 3. Here, we have chosen $\delta_1 = 0.01$ and $\delta_i = 2\delta_1$ for $i \geq 2$. In Figure 4, 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:

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

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

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

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

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

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 5. It is interesting to note that for a fewer neighboring points (H), constraint violation is more significant. For $H = 100$ points, the constraint violation is zero. The effect of robustness is also clear from Figure 6, 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 frontier. 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. Effect of constraints on variable space is depicted in Figure 6. Relationship between x_1 and x_2 is given by equation 10 and 12 respectively for simple effective and constrained effective case. It can be observed that as H increases relationship between x_1 and x_2 moves towards the theoretical constrained effective relationship.

5.1.2 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

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

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

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

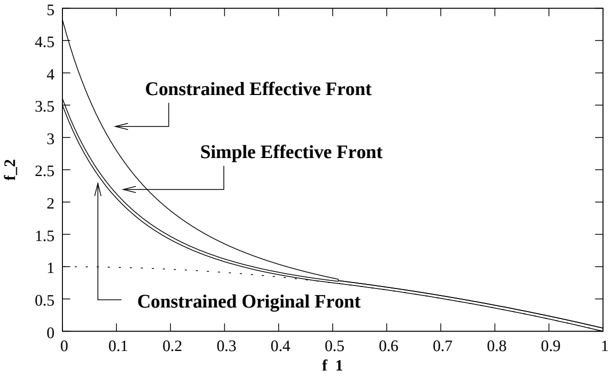

Figure 3: Theoretical simple and constrained effective fronts on test problem 1.

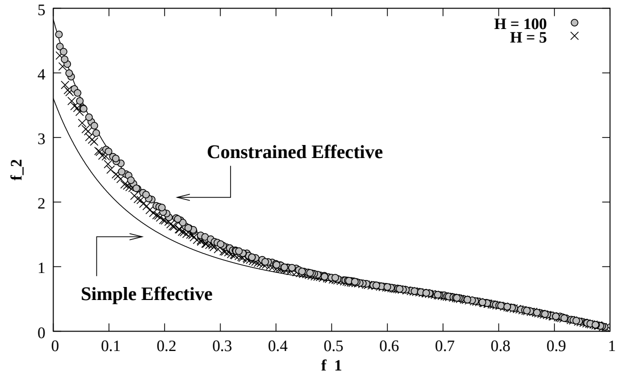

Figure 4: Constrained effective fronts showing the effect of number of points H on test problem 1.

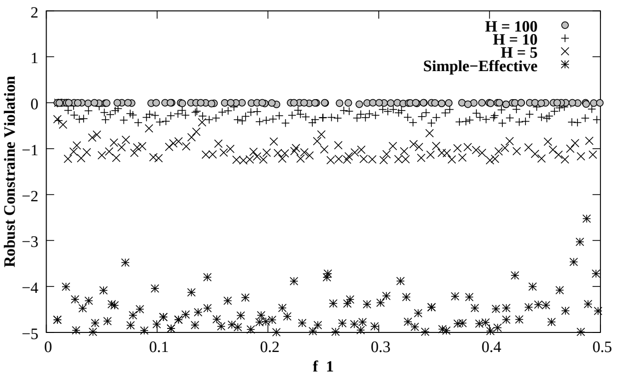

Figure 5: Robust constraint violation for the points obtained on test problem 1 with different values of H.

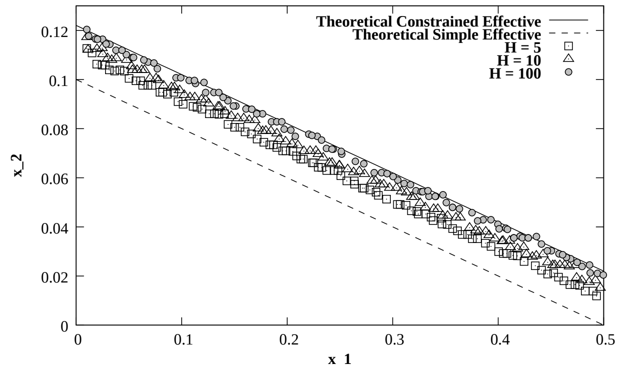

Figure 6: x_1 vs x_2 relationship for different values of H on test problem 1.

The effect of variation of H is shown in Figure 7 and 8. As can be seen from Figure 7, the constraint makes the frontier disjointed. However a close look at Figure 7 reveals that each segment of constrained effective frontier corresponding to H=10, is shorter than its simple effective counterpart. Here the effect of constraint is to contract each segment rather than lifting the frontier as in test problem 1. 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 8. 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 constraint 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.

5.1.3 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 9 and Figure 10. 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(computed 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 10. It is observed that with increasing value of H, the RCV value decreases.

5.2 Effect of δ with Approach 1 (Definition 3)

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 11 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 experimental results.

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 12 shows the effect of neighborhood size on test

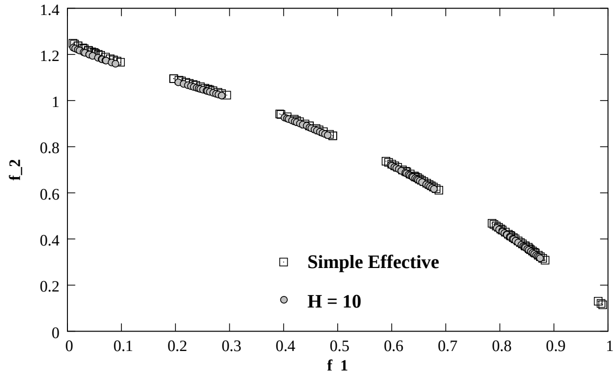

Figure 7: Constrained Effective fronts showing the effect of number of points H on test problem 2.

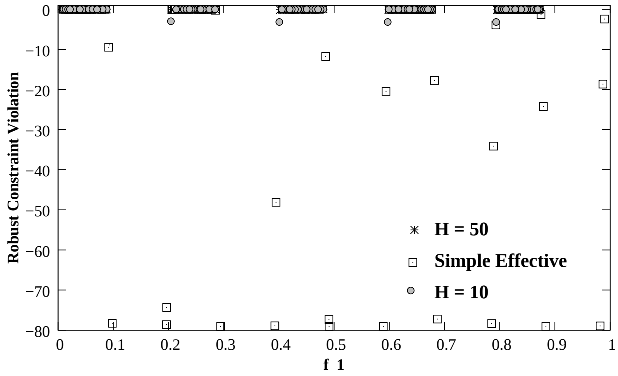

Figure 8: Robust Constraint Violation for the points obtained on test problem 2 with different values of H.

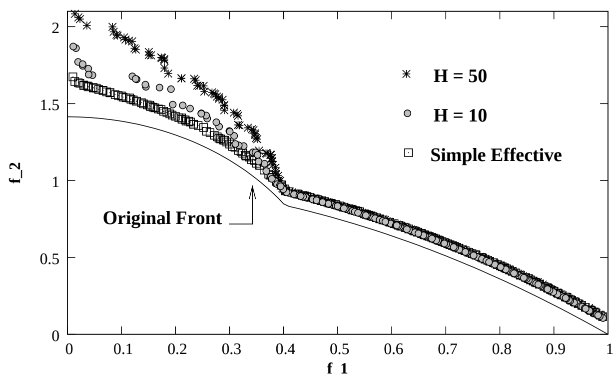

Figure 9: Constrained effective fronts showing the effect of number of points H on test problem 3.

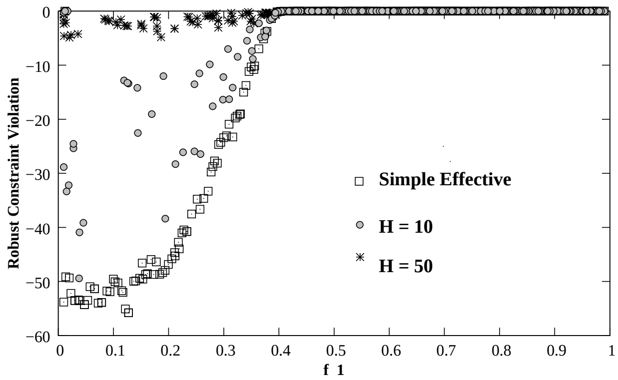

Figure 10: Robust constraint violation for the points obtained on test problem 3 with different values of H.

problem 3. With the increase in δ_i, the effective frontier moves away from the simple effective front. Figure 13 plots the corresponding G() function values. 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-objective optimization with Approach 2 (Definition 4)

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 14 and Figure 16 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. Figures 15 and 17 show the corresponding variations in the decision variable space.

7 Conclusions and Future Work

This paper suggests a couple of constraint-handling strategies for robust multi-objective optimization. These strategies are extensions of an earlier robust multi-objective optimization strategies [9]. It has been 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 should lie somewhat away from the simple effective front.

The efficacy of the two procedures has 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

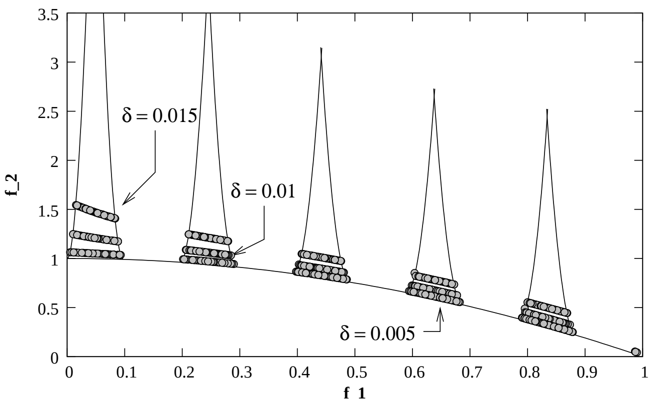

Figure 11: Constrained effective fronts showing the effect of delta δ on test problem 2.

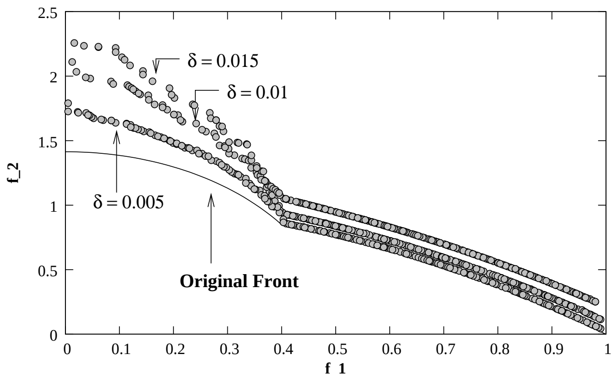

Figure 12: Constrained effective fronts showing the effect of delta δ on test problem 3.

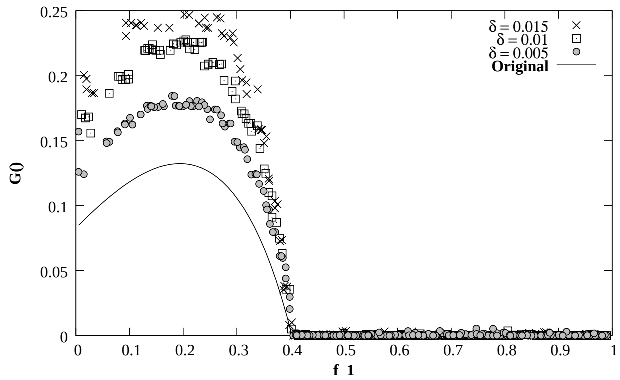

Figure 13: Effect on Variable Space for different values of δ on test problem 3.

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.

Bibliography

[1] J. Branke. Efficient evolutionary algorithms for searching robust solutions. *ACDM*, pages 275–286, 2000.

[2] J. Branke and C. Schmidt. Faster convergence by means of fitness estimation. In *Soft Computing*, 2000.

[3] J. Branke. Creating robust solutions by means of an evolutionary algorithm. *Parallel Problem Solving from Nature*, pages 119–128, 1998.

[4] Y. Jin and B. 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] J. Teich. Pareto-front exploration with uncertain objectives. In *Evolutionary Multi-Criteria Optimization*, pages 314–328, 2001.

[8] Hughes E. Evolutionary multi-objective ranking with uncertainty and noise. In *Evolutionary Multi-Criteria Optimization*, pages 329–343, 2001.

[9] K. Deb and H. Gupta. Searching for robust pareto-optimal solutions in multi objective optimization. In *Evolutionary Multi-Criteria Optimization*, pages 150–164, 2005.

[10] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. First Edition, Chichester, Wiley, 2001.

[11] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm:

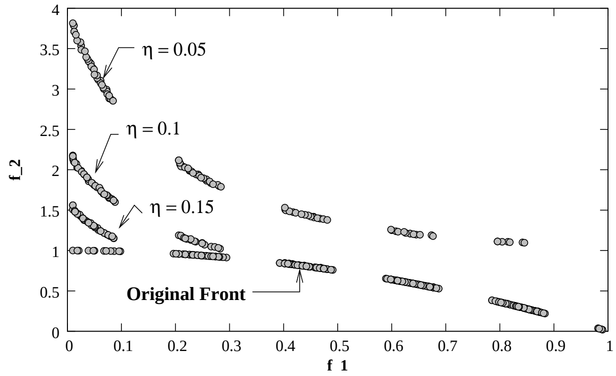

Figure 14: Constrained effective fronts showing the effect of η on test problem 2.

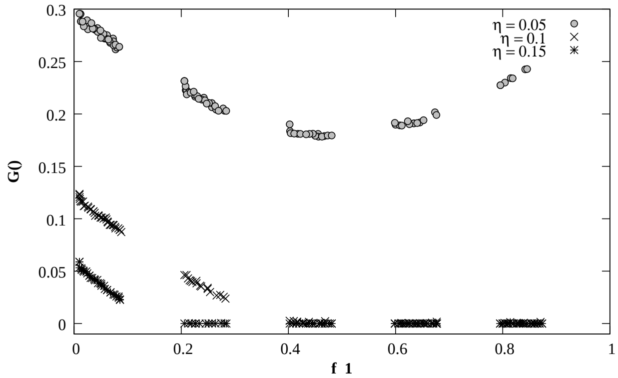

Figure 15: Effect of η on G() function for test problem 2.

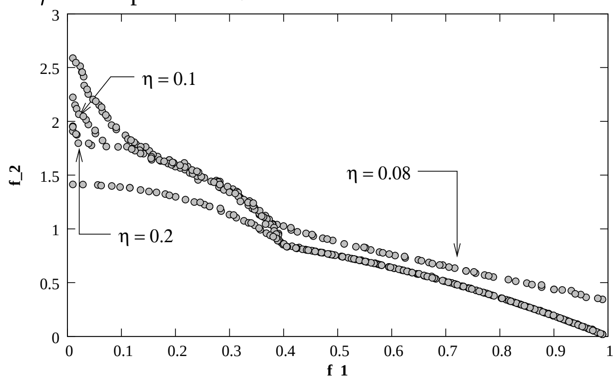

Figure 16: Constrained effective fronts showing the effect of η on test problem 3.

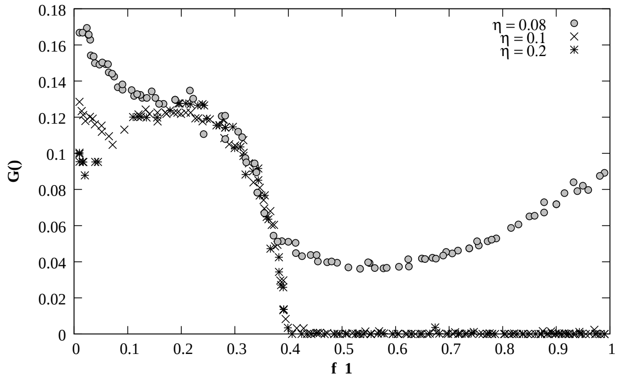

Figure 17: Effect of η on G() function for test problem 3.

Nsga-2. *IEEE transactions on Evolutionary Computation*, pages 182–197, 2000.