Searching for Robust Pareto-Optimal Solutions in Multi-Objective 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

Abstract. In optimization studies including multi-objective optimization, the main focus is usually placed in finding the global optimum or global Pareto-optimal frontier, representing the best possible objective values. However, in practice, users may not always be interested in finding the global best solutions, particularly if these solutions are quite sensitive to the variable perturbations which cannot be avoided in practice. In such cases, practitioners are interested in finding the so-called *robust* solutions which are less sensitive to small changes in variables. Although robust optimization has been dealt in detail in single-objective optimization studies, in this paper, we present two different robust multi-objective optimization procedures, where the emphasis is to find the robust optimal frontier, instead of the global Pareto-optimal front. The first procedure is a straightforward extension of a technique used for single-objective robust optimization and the second procedure is a more practical approach enabling a user to control the extent of robustness desired in a problem. To demonstrate the subtle differences between global and robust multi-objective optimization and the differences between the two robust optimization procedures, we define four test problems and show simulation results using NSGA-II. The results are useful and should encourage further studies considering robustness in multi-objective optimization.

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 any of these solutions. From a theoretical point of view, such solutions are of utmost importance in a multi-objective optimization problem. However, in practice, often a solution cannot be implemented with arbitrary precision for various reasons and the implemented solution may be slightly different from the desired solution. If a global optimal solution is *sensitive* to variable perturbation in its vicinity, the implemented solution may correspond to different objective values than that of the theoretical optimal solution. Thus,

from a practical standpoint, such solutions are of not much importance and the emphasis must be made in finding *robust* solutions, which are less sensitive to variable perturbations in their vicinity.

In single-objective optimization, a number of studies have been devoted for finding such robust solutions. Branke [1] suggested a number of heuristics for searching robust solutions. In another study, Branke [2] suggested a number of methods for alternate fitness estimation. Later, Branke [1] also pointed out key differences between searching optimal solutions in a noisy environment and searching for robust solutions. Jin and Sendhof [3] posed the issue of finding robust solutions in single-objective optimization problem as a multi-objective optimization problem with the objectives being maximizing robustness and performance. Tsutsui and Ghosh [4] presented a mathematical model for obtaining robust solutions using schema theorem for single-objective genetic algorithms. Parmee [5] suggested a hierarchical strategy of searching several high performance regions in a fitness landscape simultaneously. Teich [6] extended Pareto-dominance for handling uncertain objectives and Hughes [7] computed the error estimate for using deterministic Pareto-dominance in noisy functions. However, to our knowledge, there does not exist a systematic study introducing robustness in evolutionary multi-objective optimization.

In this paper, we make an effort to extend an existing approach for finding robust solutions in single-objective optimization for multi-objective optimization. Essentially, in this approach, instead of optimizing the original objective functions, we optimize the mean effective objective values computed at a point by averaging the function values of a few solutions in its vicinity. The solutions which are less sensitive to local perturbations will fair well in terms of the mean effective objective values and the resulting Pareto-optimal front will be the robust frontier. To illustrate the working of this approach, we first suggest four different controllable test problems and then employ NSGA-II. We also present a new definition of robustness in which original objectives are optimized but a constraint limiting the change in function values due to local perturbations is added. The latter approach is more pragmatic and a user has a control on the desired level of robustness on the obtained solutions. The differences between these two robust procedures and fundamental differences between global and robust optimization in the context of multi-objective optimization are clearly demonstrated.

Rest of the paper is designed as followed. Section 2 introduces the concept of robustness in multi-objective optimization and stresses its importance. Sections 3 and 4 discuss the two robust optimization schemes and results obtained using NSGA-II. Finally, a conclusion of this study is presented in Section 5.

2 Robustness in Optimization

We consider a multi-objective optimization problem of the following type:

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

where $\mathcal{S}$ is the feasible search space. A robust solution is defined as the one which is less sensitive to the perturbation of the decision variables in its neighborhood. Let us consider the single-objective function shown in Figure 1. Of the two

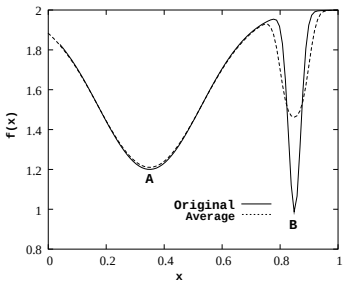

Fig. 1. Illustration of global versus robust solutions in a single-objective optimization problem.

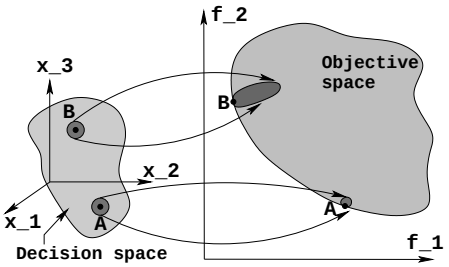

Fig. 2. Point A is less sensitive to variable perturbation than point B.

optimal solutions, solution A is considered robust as a small variation in the decision variables does not alter the objective function value of the solution. On the other hand, solution B is quite sensitive to the variable perturbation and often cannot be recommended in practice, despite having a better function value than solution A. Several EA researchers suggested different procedures of defining and finding such robust solutions in a single-objective optimization problem [2, 8, 3–5].

For solving multi-objective optimization problems, 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 [9]. The concept of robustness discussed above for single-objective optimization can be extended for multi-objective optimization as well and is worth from a practical standpoint. In Figure 2, 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 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.

2.1 Robust Optimization Approaches

One of the main approaches portrayed in the single-objective literature is to use a *mean effective* objective function ($f^{\text{eff}}(\mathbf{x})$) for optimization, instead of the original objective function ($f(\mathbf{x})$) itself. Here, we give a definition for a generic M-objective optimization problem:

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 Pareto-optimal solution to the following multi-objective minimization problem defined with respect to a* δ*-neighborhood* ($\mathcal{B}_\delta$)*:*

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

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

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

where $|\mathcal{B}_\delta|$ *is the hypervolume of the chosen neighborhood.*

To use it in practice, a finite set of H solutions ($\mathbf{y}$) can be randomly (or in some structured manner) chosen around a δ-neighborhood ($\mathbf{y} \in \mathbf{x} + \mathcal{B}_\delta$, where $\mathcal{B}_\delta = \{\mathbf{z} | z_i \in [-\delta_i, \delta_i]\}$) of a solution $\mathbf{x}$ in the variable space and the mean effective objectives (f_j^{eff}) are optimized by an EMO procedure. This way, instead of an individual's own function value (f_j), an agglomerate objective value in its vicinity is used as the objective for optimization.

Another approach would be to restrict a normalized change in perturbed objective vector from its original objective vector by a user-specified limit η:

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

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

The perturbed objective vector $\mathbf{f}^{\text{p}}$ can be chosen as the mean effective function value ($\mathbf{f}^{\text{eff}}$) or the worst function value (among H chosen solutions) in the neighborhood. The operator $\|\cdot\|$ can be any norm measure.

Both definitions for robustness in multi-objective optimization raise some interesting issues. For example, due to the variable sensitivities, a part of the original global Pareto-optimal front may not qualify as a robust front. In another scenario, the original global Pareto-optimal front (given in Equation 1) may be completely non-robust and a original local Pareto-optimal front may become robust. Depending on how robust the original global Pareto-optimal front is with respect to the above definition, there can be the four different scenarios, which we discuss next.

2.2 Case 1: Original Pareto-optimal front remains robust

This is the simplest case in which the original Pareto-optimal front remains Pareto-optimal with respect to the mean effective function values. Figure 3 illustrates such a problem. The complete original front remains robust in this type of problems.

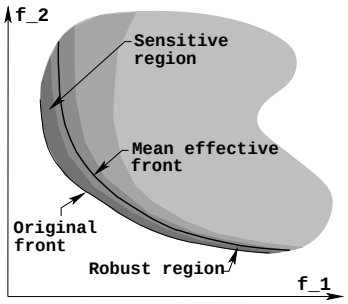

Fig. 3. Case 1: Complete Pareto-optimal front is robust.

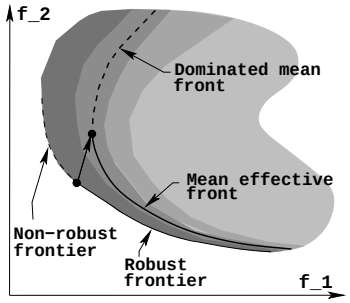

Fig. 4. Case 2: A part of the Pareto-optimal front is robust.

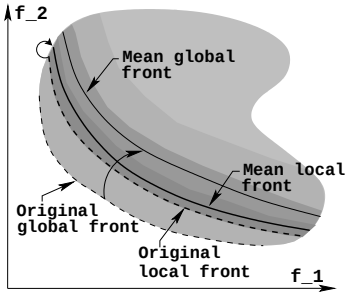

Fig. 5. Case 3: A local Pareto-optimal front is now robust.

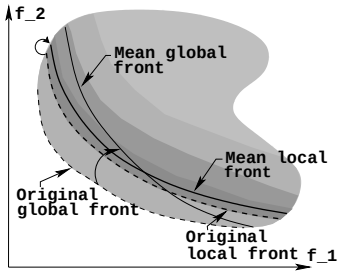

Fig. 6. Case 4: A part of the global Pareto-optimal front is not robust.

2.3 Case 2: Only a part of original front remains robust

Here, the complete original Pareto-optimal front is not robust with respect to the above definition of robustness of type I. In most real-world scenarios such a problem is expected, as some portion of the Pareto-optimal front may lie in a sensitive region in the decision variable space. In such a problem, the task of a multi-objective robust optimizer would be to find only that part of the Pareto-optimal front which is robust.Figure 4 shows that the Pareto-optimal front corresponding to the mean effective objectives does not span over the entire original Pareto-optimal region.

2.4 Case 3: Original local front is robust

Cases 3 and 4 correspond to more difficult problems in which the original problem may have more than one Pareto-optimal fronts (global and local) [9]. In Case 3, the mean effective front constructed using the original global Pareto-optimal solutions is completely dominated by that constructed using the local Pareto-optimal front, thereby meaning that the original global Pareto-optimal front is not a robust one. Figure 5 demonstrates such a problem. This type of problems, if encountered, must be solved for finding the robust Pareto-optimal front, instead of the sensitive global Pareto-optimal front.

2.5 Case 4: Only a part of original global front is robust

Instead of the complete original global Pareto-optimal front being sensitive to the variable perturbation, Case 4 problems cause a part of it to be adequately robust. In the remaining part, a new front appears to be robust. Figure 6 illustrates this problem.

Certainly, other scenarios are possible, where instead of an original local Pareto-optimal front becoming robust, a completely new frontier emerges to be robust. However, we argue that the above four scenarios most likely cover different types of robust multi-objective optimization problems which can be encountered in practice and an algorithm capable of solving these scenarios would be adequate to solve other simpler kinds.

2.6 Test Problems

In this section, we now construct a mathematical two-objective test problem for each of the above four cases.

Test Problem 1 This problem is an illustration to Case 1 discussed above:

$$\begin{aligned} &\text{Minimize } (f_1(\mathbf{x}), f_2(\mathbf{x})) = (x_1, h(x_1) + g(\mathbf{x})S(x_1)), \\ &\text{Subject to } 0 \le x_1 \le 1, -1 \le x_i \le 1, \quad i = 2, 3, \ldots, n, \\ &\quad \text{where } h(x_1) = 1 - x_1^2, \\ &\qquad g(\mathbf{x}) = \sum_{i=2}^{n} \left(10 + x_i^2 - 10\cos(4\pi x_i)\right), \quad S(x_1) = \frac{\alpha}{0.2+x_1} + \beta x_1^2. \end{aligned} \tag{5}$$

Here, we use $\alpha = 1$ and $\beta = 1$. The Pareto-optimal front corresponds to $x_i^* = 0$ for $i = 2, 3, \ldots, n$ and for any value of x_1 in the prescribed domain $[0, 1]$. At these solutions, $g(\mathbf{x}) = 0$, thereby making the following relationship between original objectives:

$$f_2^* = 1 - f_1^{*2}. \tag{6}$$

The mean effective objectives in a δ-neighborhood for a Pareto-optimal solution $\mathbf{x}$ (for $x_1 \in [0, 1]$) are given as follows:

$$\begin{aligned} f_1^{\text{eff}}(\mathbf{x}) &= x_1, \\ f_2^{\text{eff}}(\mathbf{x}) &= (1 - x_1^2) - \tfrac{1}{3}\delta_1^2 + \left[\alpha \tfrac{1}{2\delta_1} \log\left(\tfrac{0.2+x_1+\delta_1}{0.2+x_1-\delta_1}\right)\right. \\ &\quad \left. +\beta\left(x_1^2 + \tfrac{1}{3}\delta_1^2\right)\right] \sum_{i=2}^{n} \left(10 + \tfrac{1}{3}\delta_i^2 - \tfrac{10}{4\pi\delta_i}\sin 4\pi\delta_i\right). \end{aligned} \tag{7}$$

Test Problem 2 This problem is an illustration of Case 2. The mathematical formulation of this problem is identical to that in test problem 1, except that here we use $\alpha = 1$ and $\beta = 10$. The corresponding Pareto-optimal frontier for the original problem and for the mean effective objectives can be obtained from Equation 6 and Equation 7, respectively, by substituting the above parameter values.

Test Problem 3 This problem is an instantiation of Case 3. Since, this problem requires the concept of local and global Pareto-optimal front, we construct a multi-modal multi-objective optimization problem:

$$\begin{array}{l} \text{Minimize } (f_1(\mathbf{x}), f_2(\mathbf{x})) = (x_1, h(x_2)\,(g(\mathbf{x}) + S(x_1))), \\ \text{Subject to } 0 \le x_1, x_2 \le 1, -1 \le x_i \le 1, \quad i = 3, 4, \ldots, n, \\ \quad \text{where } h(x_2) = 2 - 0.8\exp\left(-\left(\frac{x_2 - 0.35}{0.25}\right)^2\right) - \exp\left(-\left(\frac{x_2 - 0.85}{0.03}\right)^2\right), \\ \quad\quad g(\mathbf{x}) = \sum_{i=3}^{n} 50 x_i^2, \quad S(x_1) = 1 - \sqrt{x_1}. \end{array} \tag{8}$$

Once again, the Pareto-optimal front corresponds to $x_i = 0$ for $i = 3, 4, \ldots, n$. Thus, at this front, $f_2(x_1, x_2) = h(x_2)S(x_1)$. Since, $f_1(\mathbf{x}) = x_1$, the local and global Pareto-optimal frontiers will correspond to the local and global minima of $h(x_2)$, respectively. A careful look at $h()$ function (shown in Figure 1) will reveal that there are two minima, of which the global minimum is at $x_2^* = 0.85$ ($h(x_2^*) \approx 1.0$). Similarly, the local Pareto-optimal front corresponds to $x_2^* = 0.35$ (with $h(x_2^* \approx 1.2$). Approximate relationships between f_1 and f_2 at these two fronts are as follows:

$$f_2 = 1 - \sqrt{f_1} \quad \text{(global)}, \qquad f_2 = 1.2(1 - \sqrt{f_1}) \quad \text{(local)}.$$

The mean effective objective values at these two fronts are given as follows:

$$f_1^{\text{eff}}(\mathbf{x}) = x_1, \tag{9}$$

$$f_2^{\text{eff}}(\mathbf{x}) = H(x_2^*, \delta_2) \sum_{i=3}^{n} \left[\frac{50}{3}\delta_i^2 + \left(1 - \frac{1}{3\delta_1}\left((x_1 + \delta_i)^{1.5} - (x_1 - \delta_i)^{1.5}\right)\right)\right], \tag{10}$$

where $H(x_2^*, \delta_2) = \frac{1}{2\delta_2}\int_{x_2^* - \delta_2}^{x_2^* + \delta_2} h(y)dy$ and is equal to 1.154 and 1.237 for local and global Pareto-optimal solutions, respectively, with $\delta_2 = 0.03$.

Test Problem 4 To represent Case 4, we construct a problem which is the same as test problem 3, with a couple of modifications: (i) the function is $h()$ is dependent on two variables:

$$h(x_1, x_2) = 2 - x_1 - 0.8\exp\left(-\left(\frac{x_1 + x_2 - 0.35}{0.25}\right)^2\right) - \exp\left(-\left(\frac{x_1 + x_2 - 0.85}{0.03}\right)^2\right).$$

and (ii) the variable bound on x_2 is different: $-0.15 < x_2 < 1$. The problem has its global Pareto-optimal front somewhere near $x_1 + x_2 = 0.85$ and the local Pareto-optimal front near $x_1 + x_2 = 0.35$, as before. However, the global Pareto-optimal front for the mean effective objectives corresponds to a mix of these two sets of values of x_1 and x_2. For f_1 smaller than about 0.5, the front corresponds to $x_1 + x_2 \approx 0.35$ and for $f_1 \ge 0.6$ values the front corresponds to $x_1 + x_2 \approx 0.85$.

3 Simulation Results

Here, we use NSGA-II [10] procedure to obtain the robust Pareto-optimal front, although any other EMO algorithm can also be used. Various parameters which would determine the extent and nature of shift of the mean effective front from the original front are as follows:

- The extent of the neighborhood (δ vector) considered to each variable.
- Number of neighboring points (H) used to compute the mean effective objectives.

We discuss the effect of these two parameters in detail for the first two test problems. However, before we discuss the results, there is an important matter which we discuss next.

There can be a number of ways of generating H neighboring points in the vicinity of a solution to compute the mean effective objective values [8]. The simplest strategy can be to randomly create H points in the neighborhood of every solution. However, this introduces additional randomness in evaluating the same solution more than once and it was suggested that a random pattern of points around a solution be created in the beginning of a generation and the same pattern be used for evaluating all population members. To create a pattern systematically, we divide the perturbation domain of each variable (around $[-\delta, \delta]$) into exactly H equal grids, thereby dividing the δ-neighborhood into n^H small hyperboxes. Thereafter, we pick exactly H hyperboxes randomly from n^H hyperboxes so that in each variable exactly one hyperbox for each of H grids is picked. Once the hyperboxes are identified, a random point within each hyperbox is chosen and is used for the computation of the mean effective objective values.

In all simulations, we have used the simulated binary crossover (SBX) and the polynomial mutation operator with distribution indices of 10 and 50, respectively. A population size of 100 is run for 10,000 generations to have confidence in the location of the robust optimal front, although the final effective frontier appears well within 1,000 generations.

3.1 Test Problems 1 and 2

Effect of neighborhood size, δ: To not have a significant effect due to finite neighboring points and variation in problem size, we use $H = 50$ and $n = 5$. To have an identical normalized neighborhood size for each variable, we use $\delta_1 = \delta$ and $\delta_i = 2\delta$ for $i \geq 2$. Figure 7 shows the theoretical mean effective front obtained using Equation 7 for four different values of neighborhood size, δ. It is clear from the figure that as δ increases, the mean effective front moves away from the original Pareto-optimal front (marked as the 'original front'). Although for this test problem, all solutions corresponding to the mean effective front are identical to those lying on the original Pareto-optimal front for any neighborhood size, the change in shape of the front is interesting. For the four δ used here,

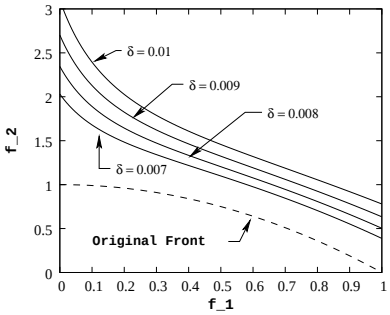

Fig. 7. Theoretical mean effective fronts showing the effect of δ on test problem 1.

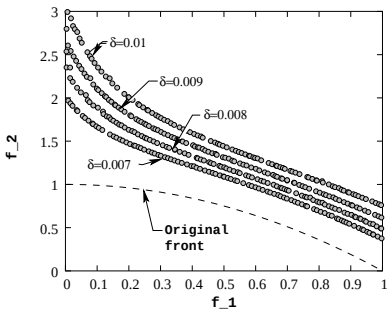

Fig. 8. Robust NSGA-II solutions show the effect of δ on test problem 1.

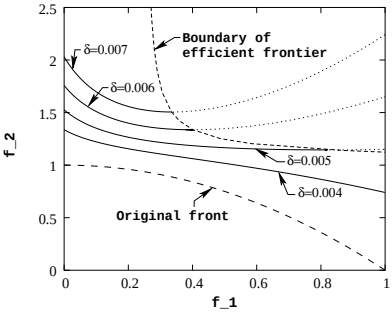

Fig. 9. Theoretical mean effective fronts showing the effect of δ on test problem 2.

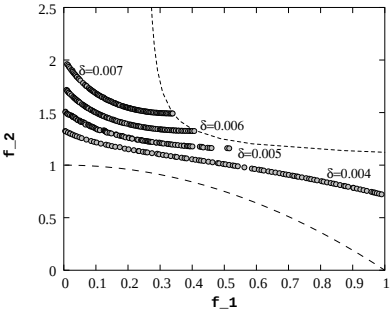

Fig. 10. Robust NSGA-II solutions show the effect of δ on test problem 2.

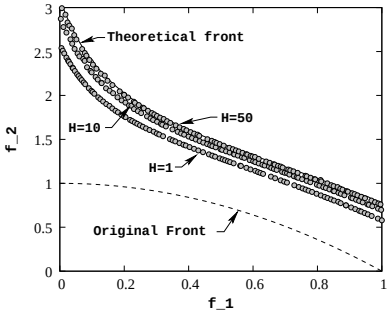

Fig. 11. Effect of H (theoretical and NSGA-II) on test problem 1.

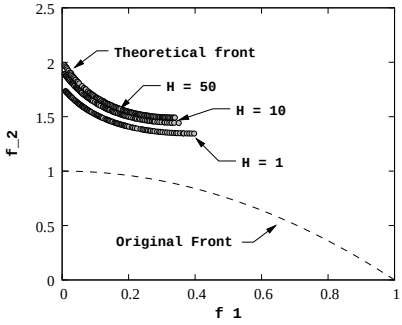

Fig. 12. Effect of H (theoretical and NSGA-II) on test problem 2.

the mean effective front is non-convex, whereas the original front was convex. It is important to highlight here that for robust optimization an EMO algorithm works on the mean effective objectives and thus may have difficulty in solving the robust optimization problem of handling a non-convex problem compared to the original convex problem. Figure 8 shows the obtained NSGA-II solutions

for the same four δ values. A close investigation will reveal that the obtained front is exactly the same as that obtained using the exact mathematical analysis (Figure 7).

Figures 9 and 10 show theoretical and NSGA-II results on test problem 2. In this problem, not only the shape of the mean effective front is different from the original one, some original Pareto-optimal solutions are no more robust. It is clear from Figure 9 that for $\delta = 0.006$, original Pareto-optimal solutions having x_1^* greater than about 0.4 now get dominated. This simply means that these Pareto-optimal solutions are quite sensitive to variable perturbation and are not robust. When performing a robust multi-objective optimization, an algorithm should then find only those Pareto-optimal solutions which are robust. Figure 10 shows that NSGA-II finds only the robust portion of the original Pareto-optimal front.

Effect of neighboring points, H: It is intuitive that if more neighboring points are chosen for computing the mean effective objectives, the objective values will be closer to the theoretical average values; however, the computation time will be more. Figure 11 shows the effect of using different values of H on test problem 1. Here, we use $\delta = 0.01$ and $n = 5$. The theoretical mean effective front (ideally for $H = \infty$) is also shown with a solid line in the figure. It is clear that as H is increased, the mean effective front shifts away from the original front and asymptotically approaches the theoretical front. Figure 12 shows the effect of H on test problem 2 (with $n = 5$ and $\delta = 0.007$). The front obtained using a small H overestimates the true robust front, but at a much smaller computational time.

3.2 Test Problems 3 and 4

For problems 3 and 4, we show the effect of local and global fronts of the original problem in deciding on the true robust front. For both problems, we use $\delta = 0.03$, $H = 50$, and $n = 5$. Figure 13 shows the theoretical results obtained using Equations 9 and 10. The original local and global fronts are shown in dashed lines. The mean effective local and global fronts are also shown in the figure with solid lines. It is clear that the mean effective local front is the robust frontier of this problem, meaning that the original local Pareto-optimal solutions are robust solutions and original global Pareto-optimal solutions are sensitive to the variable perturbation and are not robust solutions. Figure 14 shows NSGA-II solutions applied to mean objective values obtained by averaging H function values in the δ-neighborhood of a solution. The NSGA-II front is very close to the theoretical local mean effective front.

To show the difference between original Pareto-optimal front and robust front, we show all 100 obtained NSGA-II solutions for two cases. In Figure 15, we show the solutions obtained for optimizing the original problem (without robustness consideration). It is clear that for all solutions, x_2 is close to 0.85. Variables x_3 to x_5 are all settled to a value zero and the variation in the front

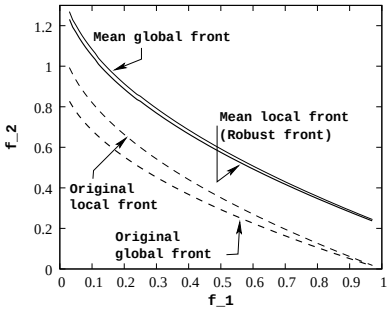

Fig. 13. Theoretical robust front for test problem 3.

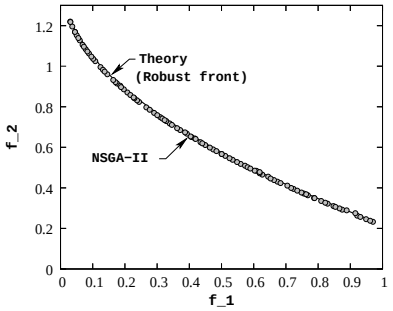

Fig. 14. NSGA-II robust front for test problem 3.

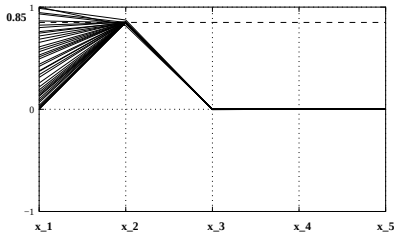

Fig. 15. NSGA-II solutions of the original test problem 3.

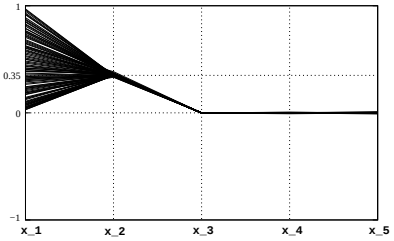

Fig. 16. NSGA-II robust solutions for test problem 3.

appears due to the variation in x_1. Figure 16 shows all solutions for the robust optimization. Here, all solutions take a value close to $x_2 = 0.35$.

Next, we consider test problem 4. The theoretical fronts for the original problem are shown in Figure 17 in dashed lines and corresponding mean effective fronts are shown in solid lines. It is clear from the figure that the robust frontier is constituted with a part of the local Pareto-optimal solutions and a part of the global Pareto-optimal solutions. Figure 18 shows the robust solutions obtained using NSGA-II. The deviation in the global part of the robust frontier from theory is due to the choice of a finite H (50 here). The original function landscape at the global frontier is quite sensitive to parameter changes, and it becomes difficult for an optimization algorithm to converge to the exact global frontier. When we rerun the problem with $H = 500$, the obtained NSGA-II solutions lie on the theoretical frontier. Figures 19 and 20 show the relationship between x_1 and x_2 in the solutions obtained for the original problem and that obtained for the mean effective objectives, respectively. It is clear that for solutions $f_1 \leq 0.5$ the relationship more or less follows $x_1 + x_2 = 0.35$ and for $f_1 > 0.6$ the relationship is $x_1 + x_2 = 0.85$. The latter condition corresponds to the original global Pareto-optimal front, as shown in Figure 19.

The above discussion on simulation results amply demonstrates that by optimizing the mean effective objectives (instead of the original objective functions)

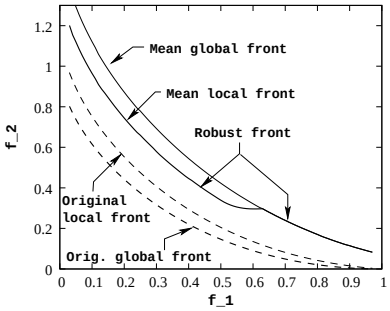

Fig. 17. Theoretical robust front for test problem 4.

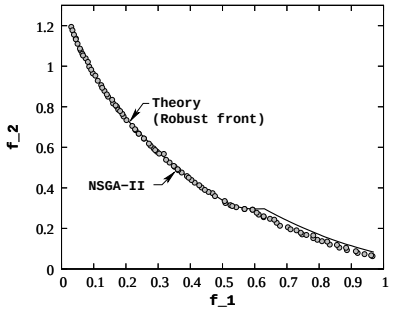

Fig. 18. NSGA-II robust front for test problem 4.

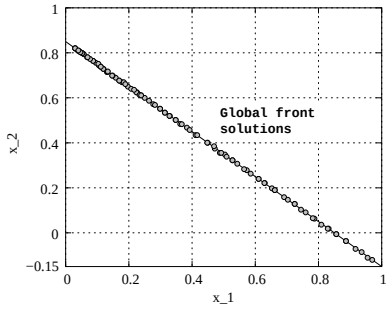

Fig. 19. NSGA-II solutions of the original test problem 4.

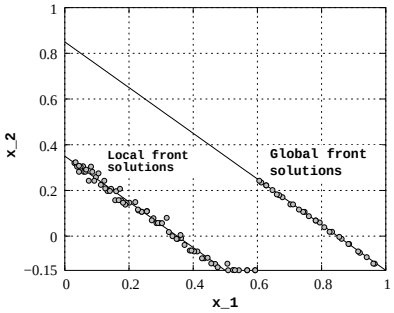

Fig. 20. NSGA-II robust solutions for test problem 4.

computed by averaging a few neighboring solutions, the robust frontier of type I can be found by using an EMO procedure. In a problem, the computation of the robust front is more useful and provides a user with the information about robust solutions directly. It has been also found that the neighborhood size and the number of neighboring points used to compute the mean objective values are important parameters in obtaining the true robust frontier. However, the type I definition of robustness is somewhat less practical and yields in a robust frontier which cannot be controlled. For a given problem, the above definition constitutes a particular front as a robust front, mainly from the consideration of mean objective values. However, a user may like a preferred limiting change in function values for defining robustness and would be interested in knowing the corresponding robust frontier. For this purpose, we have defined the robust solutions of type II earlier and discuss it in the next section.

4 Multi-Objective Robust Solutions of Type II

The robust solution of type II were defined earlier (Definition 2). Here, we use f_j^{eff} for f_j^p and the Euclidean norm for $\|\cdot\|$ operator. The limiting parameter η

is considered constant in a simulation run and is a user-defined parameter. We simply employ NSGA-II to solve the corresponding constrained optimization problem by using the constrained-domination principle [9].

To demonstrate the nature of robust solutions of type II, here we consider test problems 1 and 2, for brevity. We use $\delta = 0.007$ and 0.006 for problems 1 and 2, respectively. All other parameters are the same as before.

4.1 Test Problem 1

Figure 21 shows NSGA-II solutions obtained for different pre-defined η values on test problem 1. Here, the mathematical mean effective objective functions

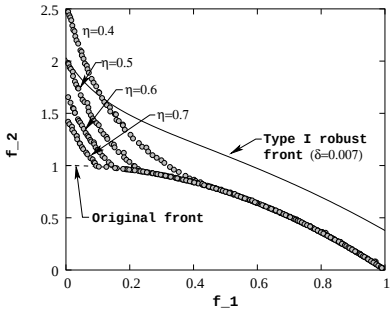

Fig. 21. Robust fronts with different η obtained using exact f^{eff} for problem 1.

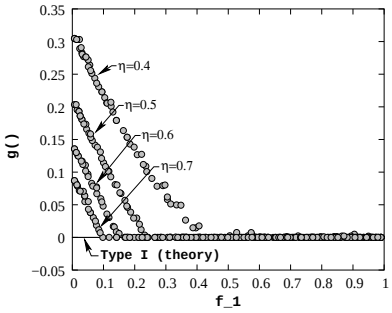

Fig. 22. Function $g()$ of the robust solutions shown in Figure 21 for problem 1.

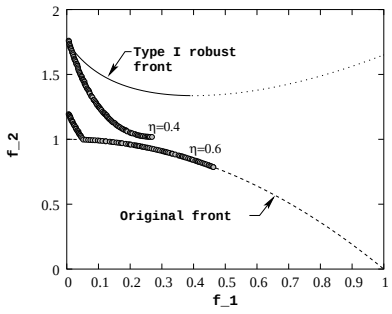

Fig. 23. Robust fronts with different η obtained using exact f^{eff} for problem 2.

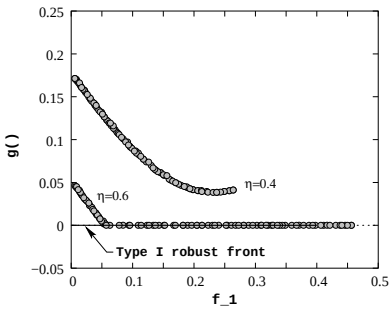

Fig. 24. Function $g()$ of the robust solutions shown in Figure 23 for problem 2.

(Equation 7) are optimized with the additional η constraint by using NSGA-II. On separate NSGA-II runs, similar fronts are obtained when the mean effective objective values are computed using $H = 50$ neighboring solutions. The figure demonstrates that the sensitive region of the original Pareto-optimal front is vulnerable to the chosen value of η. For a more tight (smaller) limiting η, the

corresponding front is further away from the original front in the sensitive region. As η is increased, the robust frontier gets closer to the original front. However, on the less sensitive portion of the original frontier, there is no change.

For comparison, the robust front obtained with type I robustness is also shown for identical δ and H parameter values in Figure 21. Recall that in the case of type I robustness, the robust solutions for this problem corresponds to $x_i = 0$ for $i > 1$ (thereby making the $g(\mathbf{x}^*) = 0$). However, with type II robustness, different solutions appear in the sensitive portion of the robust frontier having $g(\mathbf{x}^*) \geq 0$. To demonstrate this aspect, we plot $g(\mathbf{x}^*)$ values in Figure 22 for two cases: type I robust frontier (theoretical) and type II robust frontier. Although solutions having $g(\mathbf{x}^*) \geq 0$ were not the Pareto-optimal solutions of the original problem, the definition of robustness of type II causes them to be robust optimal solutions with respect to a particular η.

4.2 Test Problem 2

Figure 23 shows the NSGA-II solutions obtained using $H = 100$ neighboring points and with $\eta = 0.4$ and $\eta = 0.6$. As discussed earlier and as shown in the figure, the complete Pareto-optimal front was not robust of type I in this problem. For both η values, the robust frontiers of type II also do not cover the entire range of the original Pareto-optimal front. However, as η is increased the robust frontier comes closer to the original front. Figure 24 compares the $g(\mathbf{x}^*)$ values for all robust solutions of type I (theoretical) and type II ($\eta = 0.4$ and $\eta = 0.6$). The theoretical type I robust solutions correspond to $f_1 \leq 0.4$ and the corresponding $g()$ value for all solutions is zero. However, for the robust solutions of type II, we observe that the $g()$ values are nonzero in the most sensitive region. The NSGA-II procedure finds solutions which were non-optimal before but are robust with respect to the chosen η parameter.

5 Conclusions

This paper takes the first step towards defining robust multi-objective solutions. First, a straightforward extension of a mean effective objective approach suggested for single-objective optimization is defined for multiple objectives. In this approach (we redefined it as a robust optimization of type I), an EMO methodology can be applied to the mean effective objective values obtained by averaging a finite set of neighboring solutions. Second, we have suggested robust optimization of type II, in which the original objectives are optimized, but an additional constraint restricting a pre-defined limiting change in objective values is considered. We have argued that such a procedure is more practical, as it allows a user to find robust solutions with a user-defined limit to the extent of change in objective values with respect to local perturbations.

Additionally, we have identified four different scenarios which can happen to a robust frontier in real-world problems and suggested variable-wise scalable two-objective test problems. Simulation results of NSGA-II on these test problems

have been illustrated and explained to understand the differences between two robust optimization procedures.

A number of salient issues remains. In this research, we have considered $H = 50$ neighboring solutions to compute the mean effective objectives. Thus, in principle, this method is 50 times computationally more expensive than the regular non-robust optimization methods. This issue needs an immediate attention before such a method becomes really practical. We are currently pursuing the use of an updatable archive to store a large number of previously-computed solutions as a reservoir for neighboring solutions. Such a technique has been successfully tried for single-objective robust optimization [8], but, new insertion and deletion rules honoring the two distinct goals of multi-objective optimization – convergence and distribution – may have to be considered. Other efficient statistical techniques involving evaluations of a fewer number of neighboring solutions may also be tried. It is also important to extend the study for handling active constraints while finding robust Pareto-optimal solutions. Nevertheless, this initial study should motivate more detailed studies in the future and may encourage interested readers to understand and apply robust optimization procedures to real-world multi-objective optimization problems.

Acknowledgements

The first author wishes to acknowledge and thank Juergen Branke, Hartmut Schmeck, and Marco Farina, for some initial discussions.

References

1. Jurgen Branke. Creating robust solutions by means of an evolutionary algorithm. *Parallel Problem Solving from Nature*, pages 119–128, 1998.
2. Jurgen Branke and C. Schmidt. Faster convergence by means of fitness estimation. In *Soft Computing*, 2000.
3. Yaochu Jin and Bernhard Sendhoff. Trade-off between performance and robustness: An evolutionary multiobjective approach. In *EMO2003*, pages 237–251, 2003.
4. S. Tsutsui and A. Ghosh. Genetic algorithms with a robust solution searching scheme. *IEEE transactions on Evolutionary Computation*, pages 201–219, 1997.
5. 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.
6. J. Teich. Pareto-front exploration with uncertain objectives. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, pages 314–328, 2001.
7. E. J. Hughes. Evolutionary multi-objective ranking with uncertainty and noise. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-01)*, pages 329–343, 2001.
8. Jurgen Branke. Efficient evolutionary algorithms for searching robust solutions. *ACDM*, pages 275–286, 2000.
9. K. Deb. *Multi-objective optimization using evolutionary algorithms.* Chichester, UK: Wiley, 2001.
10. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.