Highly Reliable Optimal Solutions to Multi Objective Problems and their Evolution by Means of Worst-case Analysis

[1]Gideon Avigad
Mechanical Engineering Department
ORT Braude College of Engineering
Karmiel, Israel
gideona@braude.ac.il

C.A. Coello Coello
Computer Science Department
CINVESTAV-IPN
México
ccoello@cs.cinvestav.mx

Abstract

In the current paper, high reliability in the presence of uncertainty is of interest. Therefore, no violation of constraints, by any solution, although uncertainty exists, is mandatory. In the paper uncertainties, in which, the boundaries of uncertainty are known, are treated.

To allow a high reliability, the notion of worst-violation set is introduced. Moreover, two possible measures to assess the extent of the violation of the constraints by a solution, which is subjected to uncertainty, are suggested. One of these measures is then introduced into a multi-objective evolutionary algorithm (MOEA) in order to search for optimal reliable solutions.

It is shown that the approach applies a search towards solutions with optimal performances while taking into account high reliability. The suggested approach is the only one available so far (to the authors' best knowledge), which treats reliability through evolutionary multi objective search, while not assuming any probability distribution of the uncertainty.

Keywords: Reliability, Multi-objective optimization, Worst-case optimization.

[1] Corresponding author

1 Introduction

Multi-criteria decision making concerns the selection of solution(s) for multi-objective problems. Such a decision commonly involves comparing the solutions' representations in the objective space. The representation of a solution in the objective space may involve its performances as calculated by utilizing the problem's objective functions. If there is no uncertainty involved with the problem, the mapping between decision variable space and objective function space is a one-to-one mapping with the solution being represented by a single point in objective space. The solutions' representations are utilized for the evolutionary process, which aims to produce the problem's Pareto front (Pareto, 1896). The obtained Pareto front is associated with the optimal solutions set, which is termed the Pareto optimal set. It is noted that the notion of Pareto optimality inherently comprises the uncertainty of the designers towards their preferences of the objectives. Such preferences are sometimes referred to as range-based preferences (e.g., Deb, 2001).

According to a recent review by Coello (2005a), Evolutionary Multi-objective Optimization (EMO) has reached a mature stage. Its development has consistently been followed by applications in engineering, product development, management, and science. The development of Pareto-based evolutionary algorithms has been initiated by the procedure suggested by Goldberg (1989). Surveys and descriptions of such algorithms can be found in several references (e.g., Kicinger, 2005, Deb, 2001). According to Zitzler *et al.* (2004), advanced Pareto-based algorithms, such as SPEA2, in Zitzler *et al.* (2001), and NSGA-II, in Deb *et al*. (2000) involve three major elements. The first element concerns the creation of a search pressure towards the Pareto optimal set. This is commonly achieved by one of the known Pareto-based fitness assignment (dominance-based) techniques. The second element is set to avoid convergence to a single solution and preserve diversity. The third element is elitism, which helps to prevent losing non-dominated solutions, which are diversified.

When the problem includes uncertainties, which are associated with the design variables/environmental parameters, the mapping between the design space and the objective space for each solution is a many-to-many mapping. This is due to the solution being associated with set of (finite or infinite) scenarios, forming a cluster of performances in the objective space. In such a case, the solution may be represented in the objective space by the mean value of the cluster (e.g., Deb *et al*., 2007) or by a set of its worst scenarios (e.g., Avigad and Branke, 2008). For the evolutionary process, the approaches utilizing the mean value apply some extra knowledge on the cluster (e.g., standard deviation). Thus, the evolution is influenced both by the level of non-dominance of the mean performances point value and by the standard deviation. As a result of considering the uncertainty, the nominal front may be shifted towards a less (or partially less) optimal set. Considering the shifted set, some of the solutions may be removed from being represented to the decision makers as they become dominated in the new setting (e.g. Deb and Gupta, 2005).

In many engineering related multi-objective optimization problems (MOPs), the MOP involves not only uncertainty, but also constraints. In such a case, it is required to consider reliability, where the cluster has to be checked for constraint violation by considering the extent of the required reliability (e.g., Deb *et al.*, 2007). In Deb *et al.* (2007), two approaches amalgamating classical reliability optimization techniques with EMO are suggested. It is noted that both deal with problems where the uncertainties are not bounded, which is in contrast to the current paper.

According to Demopoulos[2], "*a major component of success involves avoiding making any major mistakes. Instead of focusing exclusively on implementing “Best Practices,” I suggest avoiding “Worst Practices.” You can do almost everything perfectly, but if you do one thing horribly wrong you can negate everything. A soldier greatly increases his chances in a firefight by doing things right, but one serious mistake and his odds of surviving plummet. Fatal flaws and mistakes are often exactly that – FATAL!*" In such cases where fatality is involved, a reliability of 100% is desired. Similar declarations might be found as also related to engineering (Anderson 2008) and as related to software design (Viega and McGraw,)Relying on an assumption of a statistical distribution is problematic. According to a well-known practitioner, Thomas Pyzdek (Pyzdek 2007): "*After nearly two decades of research involving thousands of real- world manufacturing and non-manufacturing operations, I have an announcement to make: Normal Distributions are Not the Norm!*" Some

[2] (http://www.demop.com/security-newsletter/secure-software-part1.html)

engineering examples of non Gaussian distributions may be found in Rouillard, (2007) and in Choi *et al*. (2004). This means that if a variance is not a reliable statistical measure of the distribution, or that an uncertainty function can not represent the distribution, then, counting on an estimation of them to assess reliability might be dangerous. This makes the search for the worst obligatory, in order to ensure such a high reliability. It is therefore vital to ensure that the selected solutions do not involve any scenario that might violate the constraints even on the cost of extended computational time. It is important to note that, although reliability is important, the motivation to find optimal solutions still exists.

The current paper is motivated by the requirement to find highly reliable optimal solutions to constraint-uncertain MOPs. This paper is inspired by the worst-case optimization approach of Branke *et al*. (2008), and Avigad and Branke (2008) to ensure worst–case reliability. It should be emphasized that the worst-case Evolutionary Computation (EC) treated to date, does not deal with constrained problems. To consider worst-case for constrained problems by utilizing EMO, two main issues are discussed in the current paper. The first issue is how to assess the worst-case violation of a solution's scenarios. Naturally, if there are more than two, possibly contradicting, constraints, the worst might be a set of worst cases. The second issue is how to amalgamate this assessment within an EMO search for an optimal reliable front. Considering this issue, Avigad and Coello (2008) highlighted the fact that there is no correlation between the constraints space and the objective space. This means that a solution's scenarios worst-case performances are not necessarily the same scenarios associated with the worst violation of the constraints. Therefore, the performances of a solution might be represented by one set of scenario(s) while its violation of constraints is represented by another set. These two sets are then used for developing a MOEA that will increase its selection pressure towards the optimal reliable set and front.

The following section briefly surveys the state-of-the-art as related to the solution of constrained MOPs and worst-case EMO. These two issues are in the core of the paper and will be fundamental factors in the construction of the introduced MOEA.

2. BACKGROUND

2.1 Solving Constrained MOPs by MOEAs

Although multi-objective optimization and constraint handling have received a lot of attention separately, relatively few studies have been conducted regarding the solution of constrained multi-objective optimization problems.

One main approach, which is related to several algorithms that handle constraints in MOPs, is to amalgamate dominance relations with a violation of constraints in order to prefer one solution over the other. It is reiterated here that a solution *i* is said to dominate a solution *j* if both of the following conditions are true:

1. Solution *i* is no worse than solution *j* in all objectives, *i.e.* (assuming minimization),

$$\forall f_m(\vec{x}_i) \leq f_m(\vec{x}_j)$$

2. Solution *i* is strictly better than solution *j* in at least one objective, *i.e.*,

$$\exists f_m(\vec{x}_i) < f_m(\vec{x}_j) \ .$$

Deb *et al*. (2002) proposed a constrained dominance relation as follows: A solution *i* is said to constraint-dominate a solution *j* if any of the following conditions is true:

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

Coello (2000) suggested a somewhat different definition of domination as follows:

A solution *i* is said to constraint-dominate a solution *j* if any of the following conditions is true:

1. Solutions *i* and *j* are feasible and solution *i* dominates solution *j*.
2. Solution *i* is feasible and solution *j* is not.
3. Solutions *i* and *j* are both infeasible and solution *i* violates less constraints than solution *j*.
4. Solutions *i* and *j* are both infeasible and solutions *i* and *j* violate the same number of constraints, but solution *i* has a total amount of constraint violation smaller than the constraint violation of solution *j* where the total amount of constraint violation for an individual $\vec{x}$ is given by:

$$\mathrm{coef}(\vec{x}) = \sum_{n=1}^{n_{max}} \max(g_n(\vec{x}),0)\,.$$

A closely related approach to the latter has been suggested in Jiménez and Verdegay (1998). In this case, instead of comparing two infeasible solutions based on the total sum of their constraints' violations, the solution closer to the constraint boundary is chosen. According to Oyama (2005), a solution *i* is said to constraint-dominate a solution *j* if any of the following conditions are true:

1. Solutions *i* and *j* are both feasible and solution *i* dominates solution *j* in objective function space.
2. Solution *i* is feasible and solution *j* is not.
3. Solutions *i* and *j* are both infeasible, but solution *i* dominates solution *j* in constraint space.

where dominance in constraint space is defined according the following definition:

A solution *i* is said to dominate a solution *j* in constraint space if both of the following conditions are true:

1. Solutions *i* is no worse than solution *j* in all constraints, *i.e.*,

$$\forall G_n(\vec{x}_i) \le G_n(\vec{x}_j)$$

2. Solution *i* is strictly better than solution *j* in at least one constraint, *i.e.*,

$$\exists G_n(\vec{x}_i) \le G_n(\vec{x}_j)$$

where

$$G_n(\vec{x}_i) = \max(0, g_n(\vec{x}_i))$$

The proposed method simply introduces the idea of non-dominance adopted in objective function space into the constraint function space.

In Binh and Korn, (1997) a Multi-Objective Evolutionary Strategy (MOBES) taking into account the objective functions as well as the degree of constraint violation of the infeasible solutions has been suggested. Infeasible solutions are divided into different classes according to their "nearness" to the feasible region. Thereafter, each is assigned a rank based on their class. Moreover, a mechanism to maintain a feasible Pareto set is employed within the suggested algorithm. In all of the above, the search pressure is solely applied towards the feasible front. There are quite a few studies in which the pressure is not just applied towards the feasible front, but also towards feasibility. This means that feasible solutions are further pressurized to improve in the direction of increased optimality whereas infeasible solutions are pressurized to become feasible. This is done by taking different approaches. One such approach is presented in Jimenez *et al.* (2002). There, they proposed an evolutionary algorithm with non-dominated sorting and radial slots that employs the min-max formulation for constraint handling. In addition, they introduced a diversity technique based on the partitioning of the search space into a set of radial slots along which the successive populations generated by the algorithm are positioned. Another attempt to incorporate the knowledge of constraint satisfaction during mating was proposed by Hinterding and Michalewicz (1998), in which "constraint matching" was employed during partner selection. The mating individuals are chosen according to this approach by utilizing the sum of squares of the constraints violation. An improvement to the work of Hinterding and Michalewicz (1998) has been suggested in Ray *et al*. (2001). Ray *et al*. (2001) suggested the use of three different non-dominated rankings of the population. The first ranking is performed using the objective function values; the second is performed using the different constraints; and the third is a ranking which is based on the combination of all the objective functions and constraints. Depending on these rankings, the algorithm evolves towards the feasible front. The mating process within the proposed evolutionary algorithm incorporates the knowledge of every individual's constraint satisfaction/violation and objective performance.

Another approach to drive solutions towards feasibility has been recently suggested in Yomas *et al.* (2007), who introduced an algorithm in which the value of the objective function is influenced not just by the related individual dominance relations, but also by a distance measure and a penalty that takes into account the amount of constraint violation. In that work, the preference of one individual over the other depends on consideration of the objective performances and the constraints violation as being in one space. Therefore, it is possible that an infeasible solution is assigned a higher fitness than a feasible solution. The work seems to introduce an approach that outperforms other existing approaches when considering performances, which are computed based on the so-called performance matrix (see e.g., Deb, 2001).

Repairing the infeasible solutions is yet another approach to direct the search towards the feasible region. Harada *et al*. (2007) suggested a hybrid approach to constraint handling. In this approach, a repair of infeasible solutions is carried out such that they become feasible. This is done by monotonically reducing the number of constraints violations utilizing a local search in the vicinity of the infeasible solutions (i.e., by utilizing the Pareto descent Method –PDM of e.g., Harada *et al*. 2006).

2.2 Searching reliable solutions to MOPs by EMO

In many cases, the MOP involves not only constraints, but also uncertainties. In such a case, it is required to consider reliability, for which the cluster has to be checked for constraint violation by considering the extent of the required reliability (e.g., Deb *et al*. 2007, Daum *et al*. 2007). In Deb *et al*. (2007), two approaches amalgamating classical reliability optimization techniques with EMO are suggested in order to solve the reliability problem. It is noted that both deal with problems where the uncertainties are not bounded and a statistically based distribution of the scenarios around the mean is assumed, which is in contrast to the current paper. In Daum *et al*. (2007), the concept of structural reliability has been discussed. In that discussion it has been stated that: "if the desired reliability is large, the number of samples must also increase to find at least one infeasible solution". In that paper, estimation for the failure probability through the use of the most probable point (Hasofer and Lind, 1974) has been found by utilizing the reliability index approach. This was then utilized for an evolutionary search of reliable solutions. In the current paper, worst-case optimization is considered.

The worst-case paradigm in EMOs was initiated in a work by Avigad *et al.* (2005). That work has been involved with choosing concepts to multi-objective problems. The approach presented in (Avigad *et al.* 2005) has been adopted and generalized in (Branke et al 2008) to allow a worst-case optimization using evolutionary algorithms. An embedded algorithm that formulates some related issues and allows the solution of continuous problems has been suggested by Avigad and Branke (2008). A further development aimed at reducing the computational complexity has been presented in Branke and Rosenbusch (2008) and further developments are reported in Stuermer *et* *al.* (2009). The two latter papers have treated worst-case in MOPs through the use of a coevolutionary approach.

In worst-case EMO, each possible solution is associated with a set of possible realizations. The comparison of two solutions x and y is based on their worst cases W(x) and W(y), respectively. The comparison utilizes the non-dominated representatives of $W(x) \cup W(y)$ with respect to the inverted problem (where $\min \rightarrow \max$ and vice versa). If all non-dominated representatives (in the inverted problem) belong to W(x), then solution y (worst-case-) dominates x (denoted as y $\succ$ wc x). If all non-dominated representatives belong to W(y), then solution x (worst-case-) dominates y (x $\succ$ wc y). Otherwise, the two solutions are non-dominated. In the worst-case approach, the search for optimal solutions results in a set termed as the best of the worst, and its related best of the worst front. There exist some other works within the framework of EMO which deal with robust design of solutions for multi-objective problems by taking a worst-case approach (e.g., Ong *et al*. 2006 and Lim *et al*. 2006). In those works, a combination of a max-min optimization strategy with a Baldwinian trust-region framework is employed together with a local surrogate model for designing robust solutions.

2.3 Observations and focus

The following may be stated about the works which were revised above.

a. All works assume a normal distribution of the parameters in design space. This is so at least for all the examples, which were found in those works. As surveyed by us, this is far than being correct in many engineering problems (e.g., dynamic loads, climate changes, etc.) In fact, assuming an uncertainty probability function makes some of the possible cases improbable!

b. Only one point represents a solution in determining reliability. This means that there is one worst case. Considering the notion of optimality, if the constraints are contradicting, then there might be a set of worst cases (a Pareto set in constraints' space, which is a solution of the reversed constrained violation problem). Simply saying, considering just one point for the worst, contradicts the notion of optimality in multi-objective problems. With that respect, think of a Pareto front, would the solution associated with the minimal (for a min-min problem) distance to the origin, at objective space, be chosen as the best solution over all other Pareto set's solutions? Why?

c. In all of the revised works some algorithmic parameters should be tuned (e.g., epsilon in Daum *et al*. 2007).

d. In order to assess the worst, the approaches use non evolutionary search approaches (e.g., gradient-based in Daum *et al.* 2007, and radial based neural net in Ong *et al.* 2006).

Based on the above observations, the focus of the research, which is reported in this paper, is:

a. The distribution of uncertainty associated with the design parameters is not an issue and any distribution might exist. This issue also means that any situation might occur with no probability of occurrence. This makes a major difference between the former approaches and the current one. Assuming a normal distribution, the boundary of a parameter will never be realized, whereas in the current paper it may be realized like any other case. This means that, comparing between the hereby suggested approach and former approaches is somewhat unfair. This is due to the fact that here, the worse might be inherently worse than could be found by formerly suggested approaches. Therefore, in the current paper the focus is on the introduction of a novel approach and the highlighting of some limitations of former works, avoiding artificial test comparisons.

b. A search for the worst in a multi constraints space by confining to the well defined notion of optimality in multi objective problems. This means that there are cases where a set of points represent the worst rather than a single point.

c. The introduced algorithm has no tunable parameters.

d. No hybridization of an evolutionary search is considered.

3 Methodology

In this section, the constrained worst-case optimization problem and an approach to solve it are introduced and formulated. The approach is comprised of several steps, which lead ultimately to a set of optimal and worse case reliable solutions.

3.1 Problem definition

The problem of optimizing solutions to a MOP associated with uncertainties of the environmental parameters can be formalized as:

$$\underset{(x)}{\text{Minimize}} F(x,d,p) \tag{1}$$

$$\text{where } F(x,d,p) = [f_1(x,d,p), f_2(x,d,p),.....f_K(x,d,p)]^T \text{ ; } K \geq 2$$

$$x \in X \subseteq \Omega \subseteq R^n \text{ , } x = [x_1, x_2,.....x_n]^T$$

$$\text{and } d = \{d_1, d_2,.....d_m\} \text{ , where:}$$

$$x_i \in \Omega | x_i^{(L)} \leq x_i \leq x_i^{(U)} \text{ and } d_j \in \Gamma | d_j^{(L)} \leq d_j \leq d_j^{(U)}$$

$$\text{and } \begin{array}{l} g(x,d,p) < 0 \quad \text{where } g(x,d,p) = [g_1(x,d,p), g_2(x,d,p),....g_L(x,d,p)] \\ R \cong 1 \end{array}$$

where Ω is the design parameters space (parameters that are to be chosen) and Γ is the model's environmental parameters space (which are not chosen but might be uncertain). $x_1^{(U)}, x_{n'}^{(U)}$, $d_1^{(U)}, d_{m'}^{(U)}$ are the uncertainty upper and lower boundaries of the design and environmental parameters, respectively. The term p is a vector of constants, which are not to be chosen. The last two expressions are related to the addition of the constraints and the demand for high reliability.

3.2 Solution

As a result of the uncertainty, each nominal solution, $x = [x_1, x_2,.....x_n]^T$, in which $x \in X \subseteq \Omega \subseteq R^n$, may be realized by a possible set of realizations, $r_x = \{r_x^1,..r_x^s,.....,r_x^z\}$. Each of these realizations is defined within given boundaries as follows: $r_x^{i(L)} \leq r_x^i \leq r_x^{i(U)}$, $i = 1,2,...,\ n$. Each such realization may be subjected to different environmental conditions that are expressed by the set of the environmental parameters, $d = \{d_1,..,d_j,.....,d_D\}$, where $d \subseteq \Gamma \subseteq R^D$, and d_j is bounded such that $d_j^{(L)} \leq d_j \leq d_j^{(U)}$. The combination between a possible realization of a solution x and an environmental condition is designated as a scenario of x, s^x. The set of all possible scenarios associated with a solution x is designated by S_x, where $s^x \in S_x \subseteq \Phi = \Omega \times \Gamma$. Each scenario s^x has its related performances, $y_{s^x} = F(s^x)$, in objective space. The corresponding set of all the scenarios' performances of the solution x is designated as: Y_x , $Y_x \subseteq T \subseteq R^K$.

It is desired to search for solutions to equation (1) that are optimal and highly reliable (R=1). When high reliability is sought, it is a requirement that a scenario does not violate the constraints, whereas, optimality might be considered based on the performances of the nominal solution's scenario. Considering equation (1), the nominal scenario of a solution x, is s_N^x . Here, a reliable solution will be designated by x^R. In that case, the solution to the problem is the optimal reliable set and optimal reliable front, RX, RX_f respectively:

$$\begin{gathered} RX := \{x^R \in X \mid \neg\exists\ x' \in X : F(s_N^{x'}) \preceq F(s_N^x) \wedge \neg s^x \in S_X : g(s^x) > 0\} \\ RX_f := \{y_{s_x^n} \in Y \mid y_{s_x^n} = F(s_N^x) : s_N^x \in S_x\} \end{gathered} \qquad (2)$$

Where $F(s_N^{x'}) \preceq F(s_N^x)$ means that $F(s_N^{x'})$ worst-case dominates $F(s_N^x)$.

The formulation of the solution as given in equation (2) includes demands for both optimality and reliability. That is, a solution is not dominated by any other solution, which might have worst scenarios that do not violate any of the constraints. In equation (2), the optimality is associated with domination and the reliability of a solution x is ensured by demanding that no scenario of x violates the constraints. When considering the worst-case paradigm, instead of ensuring that there is no scenario that violates the constraints, it is ensured that the worst cases will not violate the constraints. The word "worst" implies precisely that, a search for the worst, i.e., optimization. In the following sub-section, the notion of worst violating set is formulated.

3.3 Worst set and front

When there is more than one constraint and the constraints are contradicting, what is considered as the worst? The answer is straightforward: it might also be a set of the solution's scenarios. In the following, the worst-case violation is considered.

Each scenario has its related performances in the constraints space: $v_{s^x} = g(s^x)$. The set of all of a solution's scenarios' performances in the constraints space is designated as:

$$V_x\,,\, V_x \subseteq G \subseteq R^L$$

The set V_x may be sorted to find the solution x related to the worst scenarios set, $S_{wv}^x \subseteq S_x$ and its related worst scenarios front $F_{wv}^x \subseteq G_s$. These are defined as follows:

$$\begin{gathered} S_{wv}^x := \{ s^x \in S_x \mid \neg \exists\, s^{x'} \in S_x : g(s^{x'}) \preceq^R g(s^x) \} \\ F_{wv}^x := \{ v_{s^x} \in G \mid v_{s^x} = g(s^x) : s^x \in S_x^W \} \end{gathered} \tag{3}$$

where $\preceq^R$ means that $g(s^{x'})$ dominates $g(s^x)$ in the reversed problem (here; maximization), within the constraint space. More simply, the worst scenarios set of a solution are the scenarios that have the worst performances, which are found by optimizing the scenarios in the reversed constrained violation problem.

Finding the sets of equation (3) may be done by solving the following optimization problem which is defined for constraints that are given without loss of generality as:

$g(x, d, p) < 0$. The problem is in fact a reverse constrained space problem where the maximal violation is searched for.

$$\underset{(s^x)}{\text{maximize}}\, g(x, d, p) \tag{4}$$

where;

$$g(x,d,p) = [g_1(x,d,p), g_2(x,d,p), \ldots g_L(x,d,p)]$$

In the current paper, it is suggested to embedd an NSGA-II algorithm (Deb *et al.* 2002) to search for each solution's worst violating front. This is explained in Section 3.4

3.4 Comparing solutions

A solution does not violate the constraints if $\neg s^x \in S_x : g(s^x) > 0$. In the current approach, non-violation of the constraints possesses a supplemental meaning: specifically, that the solution is worst-case reliable. The demand for worst-case reliability is automatically ensured if all worst cases do not violate the constraints. It is noted that attaining all worst scenarios for a continuous space is impossible/impractical, and therefore a sample of them is the best one could do. Clearly, as the number of samples gets higher the validity of the estimation gets higher. The question is how to compare the two solutions, which are both involved with the violation of the constraints by way of their related scenarios. It is suggested that the comparison should utilize the worst violating set for this comparison. As surveyed in the introduction, comparison between two solutions when no uncertainty is involved may be accomplished by comparing the solutions' nominal performances in the objective space. Here, such approaches are not valid because, in the current case, one set is compared to another, with these sets being generally unequal in terms of size. The question is how to compare between two solutions associated with a set of scenarios. Several measures for this comparison, two of which are explained next, have been considered. In all cases, it is suggest considering only solution's scenarios that violate the constraints. This means that the worst set may be associated with scenarios which violate the constraints while others do not. Considering reliability, only the violating scenarios are of any interest. Therefore, it is suggest to dilute the set of all worst scenarios of a solution x (see equation (3)) in order to extract the worst violating set of the solution, DS^x_{wv} out of S^x_{wv}. The diluted worst scenarios set and the related diluted worst front, Df^x_{wv} are formalized as follows:

$$\begin{aligned} DS^{x}_{wv} &:= \{s^{x} \in S_{x} \mid \neg\exists\, s^{x'} \in S_{x} : g(s^{x'}) \preceq^{R} g(s^{x}) \wedge \neg\exists s^{x} : g(s^{x}) > 0\} \\ Df^{\,x}_{wv} &:= \{v_{s^{x}} \in G \mid v_{s^{x}} = g(s^{x}) : s^{x} \in DS^{W}_{x}\} \end{aligned} \quad (5)$$

Figure 1 depicts two solutions' clusters of all performances in a bi-constraints space. Each solution is designated by a different symbol (squares and circles). The solutions' worst sets (F^{x}_{wv}) are designated by gray filling and the diluted set ($Df^{\,x}_{wv}$) is designated by thick borders.

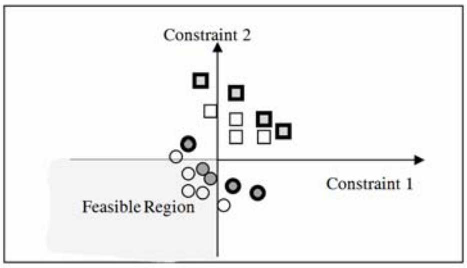

Figure 1: Two solutions' performances (designated by circles and squares) in constraints' space. The worst set of each solution is designated by gray filling and the diluted set by a bold boundary.

3.4.1 Constraint domination measure

As a first suggested measure, let's consider adopting and adapting the measure used by Avigad and Branke 2008 (initially introduced by Branke *et al.* 2008) for the comparison between two worst-case sets of a solution x and a solution x' based on their diluted worst cases as follows:

$$I_{\vartheta^{+}}(DS^{x}_{WV}, DS^{x'}_{WV}) = \min_{\varepsilon}(x)\,\{\forall x \in DS^{x}_{WV} \exists y \in DS^{x'}_{WV} : g_{i}(x') - \varepsilon \le g_{i} \ \text{for}\, i \in \{1,\ldots,G\}\} \quad (6)$$

where G, is the number of constraints of the MOP in hand. In practice, this measure may be computed by:

$$I_{\vartheta^{+}}(DS^{x}_{WV}) = \max_{x \in W_{1}} \min_{x' \in W_{2}} \max_{1 \le i \le G}(g_{i}(x) - g_{i}(x')) \quad (7)$$

The ϑ^{+}, which is termed here as the constraint domination measure, gives a numerical value to a solution x when compared to solution x'. In fact, this value is a measure of the minimal distance the diluted worst set of x' should be moved in each direction along the constraints axis in order for the diluted set of x to become the worst non-dominated case (in the reversed problem).

In order to elucidate the characteristics of the measure, which is aimed at allowing a comparison between two solutions, one should refer to the following hand calculation examples, which are related to Figure 2.

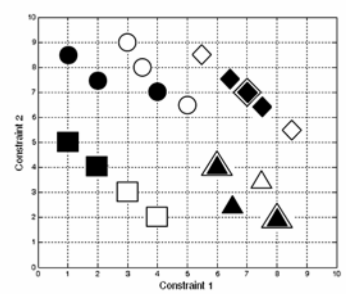

Figure 2: Four example cases for comparing worst scenarios.

In the figure, four different cases of comparing between solutions' worst sets, in constraints' space, are depicted. Each case is designated by a different symbol. In each case, two solutions are compared, designating one from the other by bold and blank worst scenarios fronts. Utilizing the measure, which is defined in equation (7), values of 1.5 and 0.5 correspond to the bold and blank circles, respectively; 0.0 and 1.0 to the bold and blank triangles, respectively; 1 and 0.5 to the bold and blank diamonds, respectively; 2 and 2 to the bold and blank squares, respectively. The first three comparisons show the following:

a. The measure gives preference to dominating scenarios (see circles).
b. The measure gives preference to concave over convex worst sets (see triangles).
c. The measure gives preference to less spread scenarios' constraints violation (see diamonds).

A comparison of the black and white squares shows equality of the measure values. Comparing between two solutions based on the constrained domination measure is fine, but the manner of ordering them within the set of infeasible solutions is a problematic issue. It calls for sorting the solutions' worst sets by non-dominance levels, which will further increase the complexity. Moreover, the way of ordering them within a domination level is unclear (here crowding is not an issue).

3.4.2 Constraint violation measure

In contrast to the relative measure introduced in the previous section, here an absolute measure, is suggested. It is also based on the worst violating set DS^{x}_{wv} and is termed here as the constrained violation measure. It is computed as follows:

$$I_{\varphi^{+}}(S^{x}_{WV}) = \frac{\sum_{i=1}^{\left|S^{x}_{WV}\right|}\sqrt{\sum_{j=1}^{G} u\left(\frac{g_j(s^{x_i})}{g_j^{\max}(s^{x_i})+\varepsilon}\right)^2}}{\left|S^{x}_{WV}\right|} \qquad (8)$$

where u is a step function and $\varepsilon << 1$. The smaller the measure is, the smaller the constraints violation is. In the numerator, the normalized distances of the solution diluted set performances from the feasible region are summed up. The step function ensures that only violations (positive values) are accounted for. The denominator normalizes the measure to account for different diluted set sizes. Considering the example of Figure 2, the following values for the constraint violation measure are obtained: black circles; 0.90, white circles; 0.987, black squares; 0.53, white squares; 0.501, black triangles; 0.869, white triangles; 0.92, black diamonds; 1.18, white diamonds;1.16. Viewing these values, it may be seen that generally the measure coincides with the constrained domination measure with the proviso that it does not prefer any of the diamond related solutions over the other. This is reasonable because the current indicator measures the distance the solution scenarios have to be moved in order for the solution to become feasible and not in order to dominate another solution. The current measure has a major advantage over the former. Being an absolute measure, it allows its direct utilization in ordering the solutions based on their violation. In the current example, the white square has the smallest violation whereas the black diamond is associated with the highest constraints violation. Thus, the constraints violation measure may serve the following purposes:

1. To assess the reliability of a solution, that is, if the measure equals zero, then the solution is worst-case reliable.

2. To compare two solutions that are violating the constraints such that if $I_{\varphi^+}(S^x_{WV}) < I_{\varphi^+}(S^{x'}_{WV})$, then, x violates to a smaller degree the constraints than x'.

3. To sort the infeasible solutions according to their constraints violation i.e., $I_N = \mathrm{sort}(I_{\varphi^+}, <)$.

Based on the advantages of the constraints violation measure, it is suggested here, to adopt it for the evolutionary search.

Considering the general case for comparing between two solutions i and j:

A solution *i* is said to constraint-dominate a solution *j* if any of the following conditions is true:

1. Solutions *i* and *j* are feasible and solution *i* dominates solution *j*, *i.e.*

$$I_{\varphi^+}(S^x_{WV}) = 0 \wedge I_{\varphi^+}(S^x_{WV}) = 0 \wedge i \succ j.$$

2. Solution *i* is feasible and solution *j* is not *i.e.*

$$I_{\varphi^+}(S^i_{WV}) = 0 \wedge I_{\varphi^+}(S^j_{WV}) > 0.$$

3. Solutions *i* and *j* are both infeasible, but solution *i* has lower value of the constraint violation measure (I_{φ^+}),

$$I_{\varphi^+}(S^i_{WV}) > 0 \wedge I_{\varphi^+}(S^j_{WV}) > 0 \wedge I_{\varphi^+}(S^i_{WV}) < I_{\varphi^+}(S^j_{WV})$$

The suggested measure and comparison procedure allow the direct use of the NSGA-II algorithm. The dominance relations are utilized both for comparing solutions in the tournament selections as well as for constructing the elite population. The evolutionary algorithm is provided and explained in the following.

3.5 The Evolutionary Algorithm

In the current paper, it is suggest using the proposed methodology within an NSGA-II framework. This has been motivated by the popularity of the algorithm, which is highlighted in Deb (2008). In fact, the procedure contains two different versions of NSGA-II. One is the main algorithm in which solutions are evolved while the other is an embedded NSGA-II in which scenarios are evolved for each solution in order to find the solution's worst set. In the suggested algorithm there is a clear aspiration for both optimality and worst-case reliability. The algorithm is introduced and explained in the following sub-section.

The MOEA

a. Initialize a population P_t of size n= $|P_t|$ which decodes the nominal scenario of a solution x (s_N^x). Also, set $Q_t = P_t$.

b. Combine parent and offspring populations and create $R_t = P_t \cup Q_t$.

c For each individual of R_t :

c.1 Initialize a population G_t of size n'=$|G_t|$ which decodes all uncertain parameters (design and environmental) within their range of search.

c.2 Run NSGA-II on the reversed constrained space problem, see equation (4)) to find for each x its worst scenarios set, S_w^x (see equation (3)).

c.3 Find for each solution x its diluted worst scenarios set DS_w^x (equation (5)).

c.4 Find for each solution x its constraint violation measure (equation (8)).

d. Sort all feasible solutions (for which $I_{\varphi^+}(S_{WV}^x) = 0$) to levels of non dominance (see Deb *et al.*, 2002) utilizing the nominal solution performances.

e. Sort all non-feasible solutions to a list I_N, $I_N = \mathrm{sort}(I_{\varphi^+}, <)$.

f. While not all individuals of R_t are sorted to a list:

f.1 Fill up a list I_L, containing all feasible solutions (solutions for which $I_{\varphi^+}(S_{WV}^x) = 0$) according to the procedure of NSGA-II (taking into account level of non dominance, boundary solutions and crowding distances).

f.2 Continue to fill up the list I_L this time, with infeasible solutions from the list I_N

g. Initialize a new parent population $P_{t+1} = \varnothing$ of size n and include the first n solutions of I_L in the new parent population: P_{t+1} , to form an elite population.

h. Create a population Q_{t+1}^{*} from P_{t+1} by a tournament selection, where the comparison between the competing individuals is based on the four conditions, which were presented in sub-section 3.3.

i. Perform Crossover on Q_{t+1}^{*} to obtain Q_{t+1}^{**}.

j. Perform mutation to obtain Q_{t+1}.

k. If the last generation has not been arrived at, go-to 'b'.

l. Present to the designers the first level of non dominance for all solutions with $I_{\varphi^{+}} = 0$.

The difference between the well-known NSGA-II and the hereby introduced algorithm is in the embedding of Step 'c' as well as the sorting of infeasible solutions in Step 'e' 2. In section 4, the algorithm is applied to both academic and engineering problems.

3.6 Computational complexity

The embedded MOEA is computationally expensive. The need to perform a complete evolutionary run for each individual of a population is a major drawback of the approach. Without considering the complexity of the non-embedded algorithm, the embedded algorithm's complexity is considerably higher than that of a common MOEA. The complexity of NSGA-II is $O(JKn^2)$, in which J is the number of generations, K is the number of objectives, and n is the population size. The n^2 term is due to the fitness assignment process. Here, for each solution, the complexity is: $O(EK(n')^2)$ where E is the number of generations of the embedded algorithm. Therefore, the overall complexity becomes $O(JKn^2 + JnEK(n')^2)$. Although computationally tractable, the complexity increases rapidly as the number of realizations (n') and the number of generations of the embedded algorithm are increased. It is further noted that, as the number of realizations used within the non-embedded part are increased, the complexity associated with the set-based comparisons is correspondingly increased. In order to reduce complexity, the following recommendations should be considered: Modification 1 - Run a basic constrained MOEA using the nominal scenarios and then discard solutions with positive values of their constrained violation measure; modification 2 - Run a basic constrained MOEA using the nominal scenarios until all solutions are feasible and then activate the suggested procedure.

4 Examples

4.1 Academic Examples

In this section, the introduced approach, is tested through utilizing adapted versions of well known constrained MOP test cases, namely the SRN (Deb et al., 2000) and the CEP1, CEP2 (Deb et al., 2001). The adaptation includes the addition of uncertainty to the design variables. The SRN test-case in its adapted version, distinguished as SRN*, is as

$$\text{follows: SRN}^*: \begin{cases} \text{Minimize } f_1(x) = 2 + (x_1^{\pm\Delta} - 2)^2 + (x_2^{\pm\Delta} - 1)^2 \\ \text{Minimize } f_2(x) = 9x_1^{\pm\Delta} - (x_2^{\pm\Delta} - 1)^2, \\ \text{Subject to } c1(x) \equiv (x_1^{\pm\Delta})^2 + (x_2^{\pm\Delta})^2 - 225 \le 0 \\ \qquad c2(x) \equiv (x_1^{\pm\Delta})^2 - 3x_2^{\pm\Delta} + 10 \le 0 \\ \qquad -20 \le x_1 \le 20 \\ \qquad -20 \le x_2 \le 20 \end{cases}$$

The difference between the original SRN and the hereby adapted one is the addition of the tolerance Δ associated with the design parameters x_1 and x_2. In the current example, Δ =0.2. The population is of size 20 and with 30 embedded individuals. The initial population (R^t) depicted by its nominal parameters values is shown in the left panel of Figure 3.

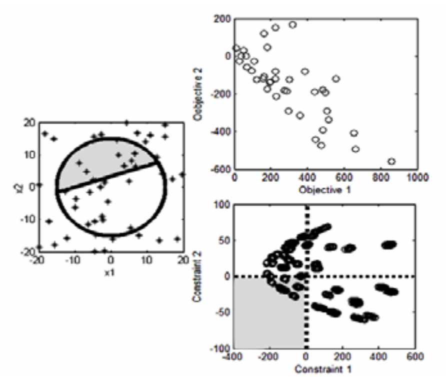

Figure 3: An initial population of solutions is depicted by its related nominal design values (left panel), its nominal solutions' performances (upper right panel) and its solutions' clusters of performances in constraint space (lower right panel). The feasible regions in design and constraints spaces are designated by gray areas.

Also shown in the figure are the constraints. The feasible region is highlighted in gray. The nominal performances of these solutions are shown in the upper right panel of Figure 3. The lower panel on the right hand side of Figure 3 depicts the solutions' scenarios performances in the constraints' space. Running the embedded algorithm for such an initial population, results in a representative front for each of the solutions. These fronts (just for the elite 20 solutions) are depicted in Figure 4.

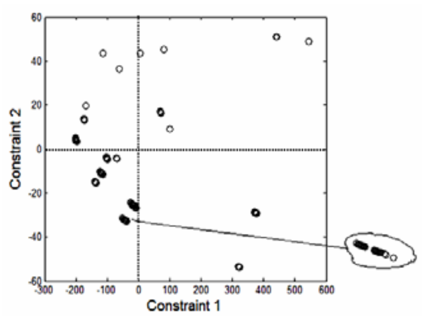

Figure 4: The initial elite population worst Pareto sets in constraints space

It is observed that some fronts are associated with a set of performances (e.g., see the enlarged circled set) while others are associated with single representing scenarios (see e.g., the points in the vicinity of [40.0, 0.0]). When executing the suggested algorithm (see Section 3.5), the number of worst case infeasible solutions decays rapidly as depicted in the left panel of Figure 5 by plotting the average constraint violation in a population versus the generation number.

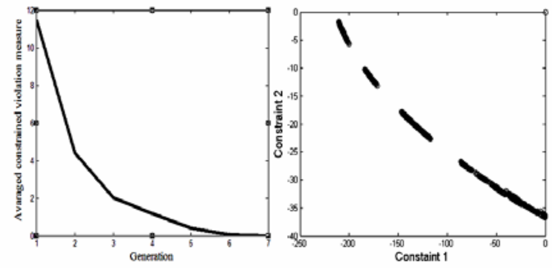

Figure 5: The decay of the number of worst-case unfeasible solutions (left panel) and the final elite population worst Pareto sets in constraints space

When the run is over, all solutions are optimal and reliable. Their constraint violation measure is non positive, as depicted in the right panel of Figure 5. It is noted that the reliable optimal set constitutes a front also in constraints space. This might be an interesting point that might be considered by multi-criteria decision makers.

Such a front does not always develop. For example, consider the following example, which is an adapted CEP1 problem (see Deb et al., 2001). With the addition of the uncertainties, the adapted constrained MOP, distinguished as CEP1*, is as follows:

$$\text{CEP1*}:\left\{\begin{array}{c} \text{Minimize } f_1(x) = x_1^{\pm\Delta} \\ \text{Minimize } f_2(x) = g(x)\exp^{-f_1(x)/g(x)} \\ \text{Here}: g(x) = (1 - x_1^{\pm\Delta})^2 + (1 - x_2^{\pm\Delta})^2 \\ \text{Subject to}: c1(x) \equiv -(f_2(x) - 0.858e^{-0.541x_1}) \le 0 \\ c2(x) \equiv -(f_2(x) - 0.728e^{-0..295x_1}) \le 0 \\ -1 \le x_1 \le 1 \\ -1 \le x_2 \le 1 \end{array}\right.$$

The left panel of Figure 6 depicts 100 solutions' representing sets in the constraints space.

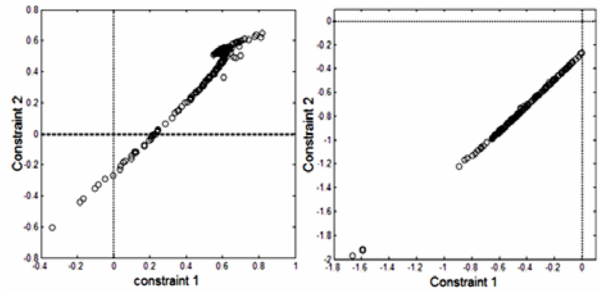

Figure 6: Initial population initial (left panel) and final violations (right panel) in constraints space.

It is shown that some solutions possess scenarios that violate both constraints, others just one, while some do not violate any of the constraints. The unique arrangement of the scenarios performances in the constraints space indicates that, in most cases, the worst is not a set but rather a single scenario performance vector. Running the suggested algorithm (of Section 3.4) for 100 generations results in the solutions worst sets performances in constraints space, which are depicted in the right panel of Figure 6. It is observed that at the end of the run, all of the evolved solutions are worst-case reliable. Changing the multipliers of the exponent in the constraints expressions shifts the apparent line to the left/right, whereas changing the multipliers in the exponents changes the angle of this apparent line.

In order to reduce the computational time associated with the introduced algorithm, two different modifications have been suggested (see Section 3.5). In order to assess the impact of these modifications with respect to the original algorithm, they are statistically compared based on the number of reliable solutions, within the elite population, after the final generation has been evolved. This has been repeated for three different test problems including SRN*, CEP1*, and CEP2*. This has been tested for Δ =0.2. The statistics are based on 40 individuals evolved for 100 generations and run 50 times for each approach and test problem. The results are summarized in Figure 7 and discussed thereafter.

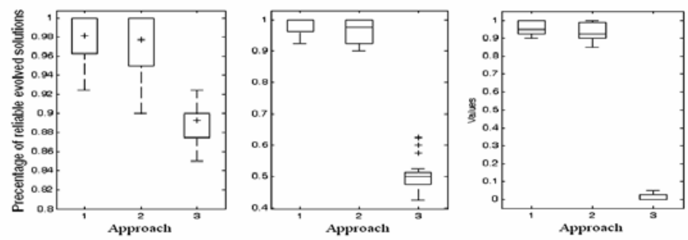

Figure 7: Comparing different search approaches.

The following might be inferred from the results. The original algorithm performs the best, although modification 1 provides competitive results. It is noted that, if fewer generations are used, modification 1 is not as competitive since it finds it harder to recover from the sudden demand for worst-case reliability. Modification 2 is clearly not an option, especially in cases in which the constraints are in the vicinity of the front, as depicted in the CEP1, and more distinctly, in CEP2. These conclusions receive further support when the uncertainty is increased.

4.2 Real life application

In order to demonstrate the applicability of the approach to real life engineering problems, the well-known cantilever problem, which has been used in Deb *et al*., 2001, is modified and utilized. The original problem, depicted in the left panel of Figure 8, involves the design of a 14 inches long beam that needs to be welded on another beam and must carry a load of 6000 lb at its end. The objectives of the design are to minimize the cost of fabrication and the end deflection. The design parameters are h, b, l, t, which are shown in the left panel of Figure 8 and are searched within the design space limits as follows: $0.25 \le h, b \le 5$ and $0.1 \le l, t \le 10$. The details of the model might be found in the Appendix.

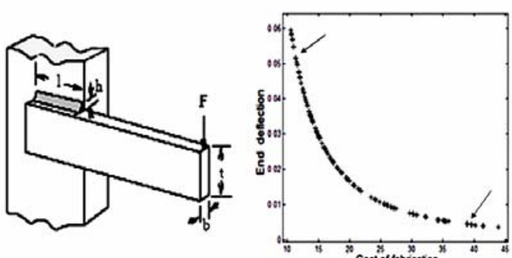

Figure 8: The welded beam design problem (left panel) and the related Pareto front, which has been evolved by setting the uncertainty to zero (right panel).

In Avigad and Branke, 2008, the problem has been utilized for the worst-case optimization by inserting uncertainties to the parameters b and t such that $\Delta t = \pm 0.2$ and $\Delta b = \pm 0.09$. In the current paper, the uncertainties of the values of the parameters t and b, which are associated with the values of the parameters h and L, has been added. The adapted problem is therefore altered such that, for each design variable, the scenarios are associated with: $t = t_{nominal} \pm \Delta t, b = b_{nominal} \pm \Delta b, h = h_{nominal} \pm \Delta h, \ell = \ell_{nominal} \pm \Delta \ell$, where, $\Delta t = \pm 0.2$ $\Delta b = \pm 0.09$, $\Delta h = \pm 0.12$ $\Delta \ell = \pm 0.08$. It is noted that in the current problem the uncertainty is involved only in the design parameters. Nevertheless this should not obstruct the validity of the example as the approach is not influenced whatsoever, from the source of the uncertainty, and the difference between the uncertainties lies only in the problem's formulation. A typical initial population solution scenarios' performance in the constraints space is depicted in Figure 9.

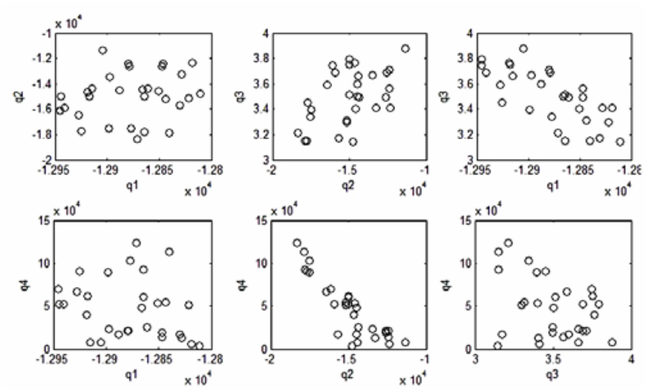

Figure 9: A typical initial solution clusters of performances in constraints space.

Each panel in the figure depicts the performances in a different bi-constraint space (out of 6 such bi-constraints combinations). It can be observed that some of the constraints are violated (e.g., g_3), while others are not (e.g., g_1). According to the suggested procedure, each solution is associated with a worst front, which has to be evolved (by the embedded algorithm).

For the solution of Figure 9, the projections of this front on the g2-g4 and g1-g3 are depicted in Figure 10.

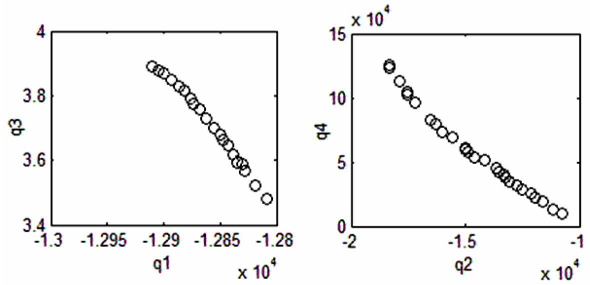

Figure 10: A solution worst violating set projection in some of the constraints sub-spaces.

The fact that in both cases a worst Pareto front is formed indicates the contradicting nature of these constraints. Nevertheless, in case of g2 versus g3, the Pareto set includes a single worst-case scenario, implying the non-contradictory nature of this couple of constraints.

When the suggested procedure is applied to the problem, the result is a Pareto set with all its scenarios' performances within the feasible region. An example of a Pareto solution set of scenarios is depicted in Figure 11, showing that all are feasible (g(x)<0). In the current example, the optimal reliable front is collocated with the original front (with no uncertainty), nevertheless it is shorter. The front is limited to the front between the two arose in Figure 8.

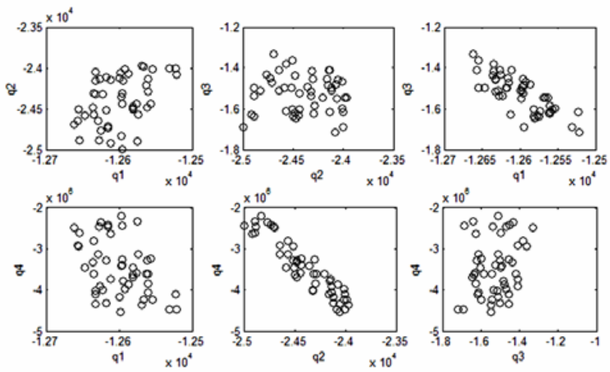

Figure 11: A typical Pareto set solution clusters of performances in constraints space.

Although the approach no doubt works well for the current problem, it highlights the main weakness of the suggested approach, namely, its computational complexity. In the current case, just two objectives are considered, but with four constraints. Naturally, if there are more constraints and more problem objectives, the complexity might be too high. Moreover, if there are many constraints, the problem of producing a good Pareto approximation is analogous to treating many objectives.

4. Summary and conclusions

In this paper, the notion of worst-case reliability, has been introduced. Solving such a reliability problem is aimed at generating solutions for which no scenario violating any of the constraints exists. To ensure such reliability, assumptions concerning the statistical distribution might not suffice. This means that here the requirement of obtaining knowledge about the boundaries of the uncertainties is vital. In order to ensure such a high reliability, the worst possible cases should be checked for their constraints violation. If the problem is associated with more than one constraint, the worst might be a set of scenarios' performances in constraints space. Finding the worst might be achieved by searching for it by way of optimization within the constraints' space. Such a search has been formulated and suggested in the current paper. It involves an evolutionary search for the solutions' worst scenarios performances in the constraints space. These scenarios are the solution worst set with their performances constituting the solution’s worst front. These solutions' fronts are utilized in order to compute a measure to the extent of the constraints violation by their worst cases. This measure is then introduced into a constrained multi-objective evolutionary search in order to evolve a set of optimal reliable solutions. Several test cases were utilized in order to demonstrate the suggested approach and its applicability to real life applications.

Considering the study reported hereby, the following may be stated:

- A multi-objective worst-case reliability problem has been formulated.

- A worst-case multi-objective multi-constraints evolutionary algorithm has been suggested and tested.
- In cases where the boundaries of the uncertainties are known or might be assessed, the suggested worst-case algorithm may ensure finding highly reliable optimal solutions.

Although the approach has been demonstrated to work well on the investigated test cases, a major problem of the approach has also been highlighted. It concerns the computational complexity, which seems to raise doubts about the applicability of the approach. When comparing the achievements of the current paper with respect to the computational issue, the following is stated:

- The proposed approach is the only one suggested till date that may guarantee worst-case reliability. At this stage it is important to emphasize again that utilizing statistical data and related measures is not to be compared with the hereby taken approach for the following reasons: a. If the distribution is not normal, then sampling this space may give samples with no relation to the normal distribution measures. It is possible to consider a transformation from one kind of distribution to the other, alas; it is not always possible to achieve a transformation. b. Till date all approaches did not assume that the boundaries of uncertainty are given or might be assessed. This means the direct evaluations may give an estimation of the boundary performances.
- If the computational time involved with the assessment of the constraints functions is within the time limit given for the reliability design, the approach should perform well. In that respect, it is pointed out that computational speed will probably increase rapidly in the future.
- When new suggested search approaches are investigated in the future, probably considering statistical distributions, the current approach may serve as a benchmark for comparison.
- Less expensive computational approaches should be sought. Saying that, it is argued again, that in cases, reliability comes to the expense of computational time and moreover what looks like elaborated

computations today, may probably appear like uncomplicated in a not too far future.

Future work should address both computational issues as well as MCDM related aspects. As related to the computational issues, the effect of the number of generations and the size of the embedded algorithm on the results should be studied. Furthermore, more elaborated cases (e.g., more parameters, more objectives/ constraints) should be examined. Some other approaches of worst-case optimization within evolutionary approach might be hybridized to the procedure in order to reduce its complexity (e.g., the coevolutionary approach suggested by Branke and Rosenbusch, 2008). It might be also interesting to search just for the knees of the reversed optimization problem or to hybridize statistical data. Concerning MCDM , an approach to support designers in reaching a decision based on the added knowledge from constraints space might be valuable. In that respect, maybe a worst-case optimization together with a worst-case constraint violation algorithm should be developed, followed by a decision making procedure to prefer worst-case optimal, worst-case reliable solutions. The decision might be posed as an auxiliary MOP where optimality and relative reliability may be contradicting.

Acknowledgements

The first author acknowledges support from OBC project no. 5000.838.2.1
The second author acknowledges support from CONACyT project no. 45863-Y.

Appendix

$$\begin{aligned}
&\text{Minimize } f_1(x) = 1.10471h^2\ell + 0.04811tb(14.0+\ell),\\
&\text{Minimize } f_2(x) = \delta(x),\\
&\text{Subject to } g_1(x) = \tau(x) - 13600 \le 0,\\
&\qquad\qquad g_2(x) = \sigma(x) - 30000 \le 0,\\
&\qquad\qquad g_3(x) = h - b \le 0,\\
&\qquad\qquad g_4(x) = 6000 - P_c(x) \le 0
\end{aligned}$$

where $\delta(x) = \dfrac{2.1952}{t^3 b}$

and:

$$\tau(x) = \sqrt{(\tau')^2 + (\tau'')^2 + (\ell\tau'\tau'')/\sqrt{0.25(\ell^2 + (h+t)^2}}\ ,$$
$$\tau' = \frac{6000}{\sqrt{2}h\ell},$$
$$\tau'' = \frac{6000(14 + 0.5\ell)\sqrt{0.25(\ell^2 + (h+t)^2)}}{2\{0.707h\ell(\ell^2/12 + 0.25(h+t)^2)\}},$$
$$\sigma(x) = \frac{504000}{t^2 b},$$
$$P_c(x) = 64746.022(1 - 0.20282t)tb^3.$$

The first constraint ensures the shear stress developed at the support is smaller than the allowable shear strength (13,600 psi). The second constraint ensures that the normal stress developed at the support is smaller than the allowable yield strength of the material (30,000 psi). The third constraint ensures that the thickness of the beam is not smaller than the weld thickness. The fourth constraint makes sure that the allowable buckling load (along direction) of the beam is more than the applied load F.

5. References

Anderson R., 2008. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Indianapolis: Wiley.

Avigad, G., and Coello C.A.C, 2008. Solving Constrained Multi-Objective Problems by Objective Space Analysis. *In Genetic and Evolutionary Computation Conference (GECCO'2008)*, 7-11, July 2008, Atlanta, USA: ACM Press, 753—754.

Avigad, G., and Branke, J., 2008. Embedded evolutionary multi-objective optimization for worst-case robustness. *In 2008 Genetic and Evolutionary Computation Conference (GECCO'2008)*, 7-11, July 2008, Atlanta, USA: ACM Press, 617-624.

Avigad, G., Moshaiov, A., Brauner, N., 2005. MOEA-based Approach to Delayed Decisions for Robust Conceptual Design. *In Applications of Evolutionary Computation, Lecture Notes in Computer Science, LCNS 3449*, Springer, 584-589.

Binh, T. T., and Korn, U., MOBES, 1997. A Multi-objective Evolution Strategy for Constrained Optimization Problems. *In* Schaffer D., ed., *proceedings of the 3rd International Conference on Genetic Algorithms*, Magdeburg, 176-182.

Branke, J., Avigad, G., Moshaiov, A., Multi-objective Worst-case Optimization by Means of Evolutionary Algorithms. Report Bru2008/12.

Branke J., and Rosenbusch, J., 2008. New Approaches to coevolutionary worst-case optimization. In Rudolf G., ed., *Proceedings of Parallel Problem Solving from Nature (PPSN X)*,13-17, September, 2008, Dortmund. *Stuttgart*, Berlin: Springer, 144-153.

Choi, S-K., Ramana, V., and Canfield, R., 2004. Structural reliability under non-Gaussian stochastic behavior. *Computers & Structures*, 82 (13-14), 1113-1121.

Coello, C.A.C., 2005. Recent Trends in Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain and Robert Goldberg (editors), *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*. London: Springer-Verlag, 7-32.

Coello, C. A. C., 2000. Constraint-Handling Using an Evolutionary Multiobjective Optimization Technique. *Civil Engineering and Environmental Systems*. 17, 9-346.

Daum, D.A., Deb,K., and Branke, J., 2007. Reliability-based optimization for multiple constraints with evolutionary algorithms. *In proceedings of IEEE congress of evolutionary computation, CEC 2007*, 25-28, September, Singapore, 25-28.

Deb, K., Pratap, A., Moitra, S., 2000. Mechanical Component Design for Multiple Objectives Using Elitist Non-dominated Sorting GA. *In the Proceedings of the Parallel Problem Solving from* Nature *VI Conference* , London Springer-Verlag. 849-858.

Deb, K., 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Pratab, A. and Meyarivan, T., 2001. Constrained Test Problems for Multi-objective Evolutionary Optimization. In E. Zitzler et al. ed., First International Conference on Evolutionary Multi-Criterion Optimization (EMO’01), March 7-9, 2001, Zurich, Switzerland. Berlin: Springer-Verlag, 284-298.

Deb, K., 2008. A Robust Evolutionary Framework for Multi-Objective Optimization. *In Genetic and Evolutionary Computation Conference (GECCO'2008)*, 7-11, July 2008, Atlanta, USA: ACM Press, 633-640.

Deb et al., 2007. Reliability-Based Multi-objective Optimization Using Evolutionary Algorithms. In Shigeru Obayashi et al., ed. Proceedings of, *4th International Conference on Evolutionary Multi-Criterion Optimization EMO 2007*, March 5-8, Matshushima, Japan. Berlin, Heidelberg: Springer, 66-80.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T.A., 2000. Fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In*: Proceedings of the Parallel Problem Solving from Nature VI Conference*, Paris, France, 849–858,

Deb, K., Pratap, A., and Moitra, A., Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA. Kangal Technical Report No. 200002, 2002

Deb, K., and Gupta, H., 2005. Searching for robust Pareto-optimal solutions in multi-objective optimization. In Proceedings of the *Third Evolutionary Multi-Criteria Optimization (EMO-05)* Conference. March, 9-11, Guanajuato, Mexico. Berlin: Springer, 150-164.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison Wesley.

Harada, K., Sakuma, J., Ikeda, K. Ono I., and Kobayashi, S., 2006. Local Search for Multi-objective Function Optimization: Pareto Descend Method. *In proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, New York: ACM Press, 659-666.

Harada, K., Sakuma, J., Ono I., and Kobayashi, S., 2007. Constraint-Handling Method for Multi-objective Function Optimization: Pareto Descent Repair Operator. In Shigeru Obayashi et al., ed. Proceedings of, *4th International Conference on Evolutionary Multi-Criterion Optimization EMO 2007*, March, 5-8, Matshushima, Japan. Berlin, Heidelberg: Springer, 156-170.

Hasofer, A, and Lind, N., 1974. An exact and invariant first order reliability format. Journal of the Engineering Mechanics Division, ASCE, 100(1), 111-121.

Hinterding, R. and Michalewicz, Z., 1998. Your brains and my beauty: parent matching for constrained optimization. In *Proceedings of the Fifth international Conference on Evolutionary Computation,* Anchorage, Alaska. San Francisco: Morgan Kaufmann Inc, 810-815.

Jiménez, F. and Verdegay, J.L., 1998. Constraint Multiobjective Optimization by Evolutionary Algorithms, *Proceedings of the International ICSC Symposium on*

Engineering of Intelligent Systems (EIS'98), February 11-13. New York, NY, USA: Cambridge University Press, 266-271.

Jimenez, A., Gomez-Skarmata, A.F., Sanchez G., and Deb, K., , 2002. An Evolutionary Algorithm for Constrained Multi-objective Optimization. *In the Proceedings of the Evolutionary Computation on 2002. CEC '02.*, Washington, DC, USA: IEEE Computer Society. 1133-1138.

Kicinger, R., Arciszewski, T., and De Jong, K. A., 2005. Evolutionary computation and structural design: a survey of the state of the art. *Computers & Structures*, 83(23-24), 1943-1978.

Lim, D. Ong, Y.S. Jin, Y. Sendhoff, B. and Lee, B.S., 2006. Inverse Multi-Objective Robust Evolutionary Design. *Genetic Programming and Evolvable Machines*, 7(4), 383-404.

Ong, Y. S., Nair P. B., and Lum, K. Y., 2006. Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust Design. IEEE Transactions on Evolutionary Computation, 10(4), 392-404.

Oyama A., Shimoyama K., and Fujii K., 2005. New Constraint-Handling Method for Multi-Objective Multi-Constraint Evolutionary Optimization and its Application to Space Plane Design. In R. Schilling et al., ed. *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2005)*, September 12 - 14, Munich, Germany.

Pareto, V. *Cours D'Economic Politique*, *1896 Volume 1 and* 2 Frouge, Lausanne.

Pyzdek, T., and Keller, P., 2009. *The Six Sigma Handbook*. Place:McGraw-Hill. Available from: http://ebooks.ebookmall.com/ebook/135574-ebook.htm.

Ray, T., Tai, K., and S.O., Ceow, 2001. An Evolutionary Algorithm for Multi-objective Optimization. *Engineering Optimization*, 33(3), 399-424.

Rouillard, V., 2007. On the Non-Gaussian Nature of Random Vehicle Vibrations. *In Proceedings of the World Congress on Engineering 2007* July 2 - 4, London, U.K.

Stuermer, P., Bucci, A., Branke, J., Funes, P., and Popovici, Elena., 2009. Analysis of Coevolution for Worst-Case Optimization. *In Proceedings of the 11th Annual conference on Genetic and evolutionary computation GECCO 2009*, July 8-12, Montreal, Canada, 899-906.

Viega, J., and McGraw. G., 2007. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley.

Yomas G. Woldesenbet, Biruk G. Tessema and Gary G. Yen., 2007. Constraint Handling in Multi-Objective Evolutionary Optimization . *In proceedings of IEEE* *congress of evolutionary computation, CEC 2007*, 25-28, September, Singapore. IEEE Press, 3077-3084,

Zitzler, E., Laumanns, M., and Bleuler, S., 2004. A Tutorial on Evolutionary Multiobjective Optimization. In Xavier Gandibleux, Marc Sevaux, Kenneth Srensen and Vincent T'kindt (editors), *Metaheuristics for Multiobjective Optimisation*, Lecture Notes in Economics and Mathematical Systems, 535, 3-37. Berlin: Springer,

Zitzler, E., laumanns, M., and Thiele, L., 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092.