

Embedded Evolutionary Multi-Objective Optimization for Worst Case Robustness

Gideon Avigad

Mechanical Engineering Department
ORT Braude College of Engineering
Karmiel, Israel
gideona@braude.ac.il

Jürgen Branke

Institute AIFB
University of Karlsruhe
76128 Karlsruhe, Germany
Branke@aifb.uni-karlsruhe.de

ABSTRACT

In Multi-Objective Problems (MOPs) involving uncertainty, each solution might be associated with a cluster of performances in the objective space depending on the possible scenarios. Therefore, in MOPs, the worst case might not be a single scenario but rather a set of such worst case scenarios, depending on the user preferences. The evolution of solutions based on their related sets of worst case scenarios has been recently introduced. It has been termed: "worst case evolutionary multi-objective optimization." In the current paper the worst case evolutionary multi-objective optimization is further developed. In contrast to the former work where the number of possible scenarios is small and the set of worst cases can thus be easily determined, here, the number of scenarios is assumed to be large, and the worst cases are searched for by means of an embedded evolutionary search. This means that for each nominal solution, a worst set of scenarios has to be found. In the current study, the resulting front, consisting of sets of solutions' worst cases, is formally defined, and a new approach to support decision making based on it, is suggested. The new decision support poses the selection as an auxiliary MOP, highlighting the tradeoff which might result from the worst being a set and not a single point. An academic example and an engineering design problem are given in order to explain the methodology and to demonstrate its applicability to real life problems.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving - Heuristic Methods
J.6.1 [Computer-aided Engineering]: Computer-aided Design

General Terms: Algorithms, Design.

Keywords: Multi-objective, Robustness, Worst-case

1. INTRODUCTION

WHENEVER the objectives of a MOP problem are contradicting, there are usually multiple solutions with different trade-offs.

Such solutions which can not be improved by either objective without deteriorating at least another objective are termed Pareto optimal.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07...\$5.00.

The selection of a solution out of the set of Pareto optimal solutions is based on the designers' preferences. Solving such problems (i.e., finding the Pareto set), is generally considered as a difficult problem. Evolutionary Algorithms (EAs) in general and Multi-Objective Evolutionary algorithms (MOEAs) in particular possess several characteristics, which make them suitable for solving this type of problems. Often, algorithms provide solutions that may not be Pareto optimal, but may satisfy other criteria, making them significant for practical applications. For example, Parmee [2] introduced Cluster Oriented Genetic Algorithm (COGA), where the result of the search for solutions to the MOP, is a set of solutions that are related to 'interesting regions'. In [4], where the focus is on diversity within the design space on the expanse of optimality in objective space, non-Pareto solutions are consciously searched for.

Searching for robust solutions (e.g., [5]) and reliable solutions (e.g., [6]), may also end up with solutions which are different from those of the deterministic Pareto set.

When a solution is subjected to uncertainties, it is associated with a set of scenarios, resulting in different performances in the objective space. Thus, each solution is associated with a cluster of performances in the objective space. While evolving solutions towards optimal and robust solutions, these solutions' clusters should be represented during the evolution. Several approaches to represent these clusters are present (e.g., using the average and the standard deviation). Recently in [7], a new approach has been introduced in which each solution is represented by its worst set in the objective space. This is motivated by a possible demand to ensure a reliability of 1 (no failure). In [7], the worst set of a solution is specified and a suggestion to embed an algorithm that searches for this worst set has been proposed.

In the current paper this suggestion is embraced and an embedded algorithm is introduced. Moreover the resulting robust front is examined from the point of view of a decision maker and an approach to support decision making is provided.

2. BACKGROUND

A search for robust solutions is aimed at ensuring that performance requirements are met and constraints are not violated due to system's uncertainties and variations (e.g., [8]). Fundamentally, robust design is concerned with minimizing the affect of such variations without eliminating the source of the uncertainty or variation (see [9]). Taguchi, (e.g., [10]) has contributed tremendously to the development of this field of interest by introducing several approaches (e.g., Loss Function, Orthogonal Arrays and Linear Graphs). It is well known that optimality and robustness might be

contradicting demands (e.g., [11]). The importance of arriving at a robust design is well known and the interest at that field is reflected by the vast number of works, which are partially surveyed in the following.

Robust performance approaches involve situations in which either the design variables and/or the environmental parameters are subject to perturbations or changes. There are many possible ways to treat robustness by using Evolutionary Computation (EC), and a few possible heuristics have been suggested in [12]. The existing research work is commonly concerned with robustness as related to single objective problems, and the optimization of the expected fitness, given a probability distribution of the disturbance. Since it is usually not possible to calculate the expected fitness analytically, it has to be estimated. This, in turn, raises the question of how to estimate an expected fitness efficiently, and how to optimize based on such estimates. Evolutionary algorithms have been shown to be quite robust with respect to uncertainty in the fitness values, (e.g., [13]).

Dealing with robustness within MOPs is a relatively new issue. Just recently Deb and Gupta, [14], introduced a formulation for the different aspects of robustness within Evolutionary Multi-objective Optimization (EMO) and suggested an approach to evolve a robust front based on the mean of an effective fitness function. Other EMO algorithms as related to robustness are also surveyed in [7].

Using the worst case within in an EMO search is scarcely found. Luo *et al.* [15] used an EMO approach to evolve robust fronts, which are a result of taking into consideration possible market changes. In their work it has been assumed that, given a design space, the designer typically can specify a target point in terms of aimed design objective values. This target becomes the basis for determining the worst-case objective values and the best-case objective values under the variations in uncontrollable design parameters.

As discussed in the introduction, in MOPs it is expected that there exist several worst cases. Considering a set of worst cases, which are associated with a solution to a MOP, [16] suggested an EMO algorithm. There, it has been related to delayed decisions within a conceptual design. Recently Branke *et al.*, [7], have generalized and improved the algorithm, which has been presented in [16]. The work in [7] considered worst-case multi-objective optimization, where a solution is evaluated by means of a finite set of different worst scenarios. It is assumed that it is impossible to reduce a solution to a single worst-case representative, because different users would consider different representatives as their worst case. This assumption led to the need to compare between solutions based on sets of worst cases. For this purpose, a definition of dominance for worst-case optimization based on sets of representatives has been suggested. Comparing between 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 $y >_{wc} x$). If all non-dominated representatives belong to $W(y)$, then solution x (worst-case-) dominates y ($x >_{wc} y$). Otherwise, the two solutions are non-dominated. This allowed performing non-dominance ranking of the solutions. To allow such set based dominance ranking two measures were considered. One is based on the expected marginal utility [19], while the other is based on the distance a solution needs to be shifted to become non-dominated. In the current paper the latter is used and therefore a short description of it is given in the following.

The general idea is to adapt the $\varepsilon+$ indicator of [3] for the case of worst-case domination. For distinction, it has been termed in [7] as $\delta+$ indicator. It is computed as follows:

$$I_{\delta^+}(W_1, W_2) = \min(x) \{ \forall x \in W_1 \exists y \in W_2 : f_i(y) - \varepsilon \leq f_i(x) \text{ for } i \in \{1, \dots, K\} \} \quad (1)$$

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

$$I_{\delta^+}(W_1, W_2) = \max_{x \in W_1} \min_{y \in W_2} \max_{1 \leq i \leq K} (f_i(x) - f_i(y)) \quad (2)$$

The $\delta+$ indicator allows comparing between two sets of worst-case representatives. It is noted that when comparing between two non-dominating sets based on Equation 2, the measure takes into account three main factors. These are the diversity of the set (when compared to other sets) its spread and whether it is a convex or a concave front. It may be verified by following the example from [7] that the bullets in Figure 1 would be preferred in the evolution over the other two sets, which are designated by squares and triangles.

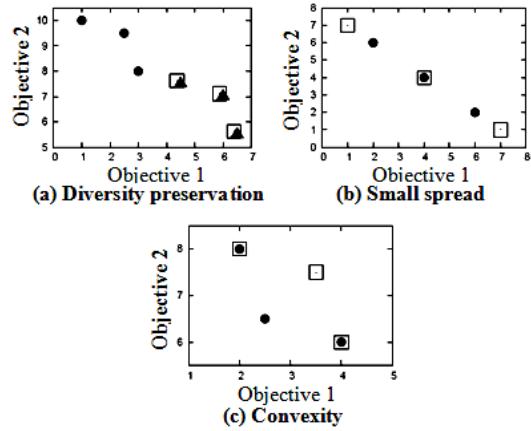


Figure 1: Preferring diverse, small spread convex set.
The Figure appears originally in [7].

However, for selection, there is a need for a definition of how good a solution is with respect to all other solutions. Therefore, in [7] it has been proposed to define a solution's surrogate fitness $Fit(x)$ as the minimum distance a solution has to be moved to become non-dominated, or, if it is already non-dominated, the maximum distance it can be moved until it becomes dominated. Formally, it has been expressed as

$$Fit(x) = \min_{y \in P \setminus x} \{ I_{\delta^+}(x, y) \} \quad (3)$$

where P is the population of all individuals. Furthermore, it has been suggested to assign the boundary solutions (of each non-dominated front) in either objective, a very high value to keep them in the population.

The algorithm of [7] treats problems where the uncertainty can be modeled by a small, known set of scenarios. In other words, the solutions' worst-cases' performances can be easily enumerated. In current paper this limitation is addressed by introducing an embedded EMO, which searches for these worst sets. By inspecting the results of [7], it becomes evident that when the Pareto set includes worst sets of solutions the decision making problem, which is inherently problematic in MOPs, is intensified. Therefore some new decision support approaches are obligatory. For this reason, a possible approach is formulated and discussed.

3. METHODOLOGY

This section is organized as follows: Section 3.1 introduces the definition for worst case MOP. In Section 3.2, the solution to the problem is formulized. Section 3.3 introduces one possible approach to sort the resulting solutions in order to support decision making. Finally Section 3.4 introduces the embedded algorithm.

3.1 Problem definition

An unconstrained uncertain multi-objective problem may be formulated as follows:

$$\underset{(x)}{\text{Minimize}} \quad F(x, d, p) \quad (4)$$

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

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

$$d \in D \subseteq \Gamma \subseteq \mathbb{R}^m, d = [d_1, d_2, \dots, d_m]^T$$

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

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). It is assumed that for all $n' \leq n$ uncertain design parameters, their lower bounds $x_1^{(L)}, \dots, x_n^{(L)}$ of uncertainty as well as $x_1^{(U)}, \dots, x_n^{(U)}$, their uncertainty upper boundaries, are known or may be determined. This is also assumed for all $m' \leq m$ uncertain environmental parameters; $d_1^{(L)}, \dots, d_m^{(L)}$ and $d_1^{(U)}, \dots, d_m^{(U)}$. Each possible solution is associated with a set of possible realizations. As a result of the uncertainty, the nominal solution x may be realized by a possible set of realizations, $r_x = (r_x^1, r_x^2, \dots, r_x^z)$. Each such realization may be tested within any of the possible environmental situations, d out of all possible situations D . The combination between a possible realization of a solution x and an environmental situation is designated as a scenario of x , $s^x = r_x^z \cup d$. The set of all possible scenarios associated with a solution x is designated by S_x , where $S_x \subseteq \Phi = \Omega \times \Gamma$. Each scenario has its related performances in objective space: $y_{s^x} = F(s^x)$. The set of all of a solution's scenarios' performances in the objective space is designated as Y_x , $Y_x \subseteq T \subseteq \mathbb{R}^K$.

In this paper the approach is to consider the worst case and therefore the problem already defined in Equation 4, is hereby restated. It is termed "worst-case MOP" and its definition is:

find x such to

$$\min_x (\max_{s^x} (F(s^x))) \quad (5)$$

where: $F(s^x) = [f_1(s^x), f_2(s^x), \dots, f_K(s^x)]^T$

Equation 5 states that solutions, which posses the best of the worst performances are searched for.

3.2 The solution

In MOPs, finding the max of the transformation (of Equation 5) may be associated with finding a set of scenarios' performances. Thus the maximization in Equation 5 results in a worst set of scenarios for

each nominal solution, x , $S_w^x \subseteq S_x$ and its related worst front $f_w^x \subseteq Y_x$. These are defined as follows:

$$\begin{aligned} S_w^x &:= \{s^x \in S_x \mid \neg \exists s^{x'} \in S_x : F(s^{x'}) \preceq^R F(s^x)\} \\ f_w^x &:= \{y^s \in Y_x \mid y^s = F(s^x) : s^x \in S_w^x\} \end{aligned} \quad (6)$$

where \preceq^R means that $F(s^{x'})$ do min ates $F(s^x)$ in the reversed problem (here; maximization). To elucidate the different notions, which where discussed so far refer to Figure 2. One out of several realizations of solution (encircled by a curvature in Ω space) together with an environmental situation ($d \in D$) are forming a scenario $s^x \in S_x$. This scenario is mapped to a point ($y_{s^x} \in Y_x$) in the objective space by the objective function, $F(s^x, p)$. All the worst scenarios' performances, form the worst Pareto front of solution x , f_w^x (designated by blank circles at space T) and the related scenarios' set S_w^x (designated by blank circles at space Φ).

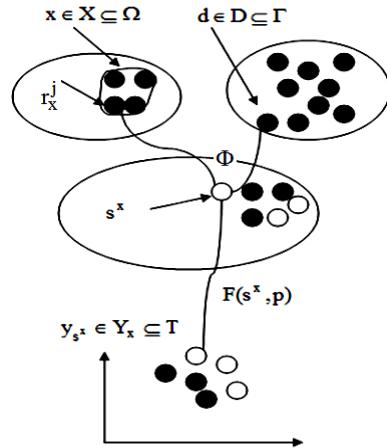


Figure 2: The realization of a solution and its related performances.

The solutions' worst scenarios and their associated worst fronts may now be used as the worst cases to assign fitness to the solution by using the approach introduced in [7]. The result of the worst case EMO of [7] is the Best of the Worst set, BW , of solutions with the best of the worst scenarios' performances, and a related set of Best of the Worst fronts, BW_f , which are here defined as follows:

$$\begin{aligned} BW &:= \{BW \subseteq \bigcup_x S_x \mid \neg \exists S_w^x : S_w^x \succ_{wc} S_w^x\} \\ BW_f &:= \{f_w^x \in BW \mid f_w^x = \bigcup F(s^x) : s^x \in S_w^x\} \end{aligned} \quad (7)$$

The EMO algorithm, which enhances the search for the solution of Equation 7, is described in Section 3.4.

3.3 Supporting decisions

3.3.1 The problem of set based decisions

Pareto optimality is associated with an inherent uncertainty of the designers towards their preferences of the objectives. It is generally assumed that, a-posteriori to the introduction of the Pareto front to the designers they should have that insight that will point the selection towards a specific solution by inspecting the resulting Pareto front.

This is commonly done by a decision on such preferences, which are referred to in [17] as range based preferences. In the following it will be assumed that there are no range based preferences. In the worst case EMO there are some other issues that might influence the decision. To highlight these issues, Figure 3 depicts four BW solutions' performance sets designated by different blank symbols (each symbol designate a set of worst cases). It is depicted that although all solutions belong to the BW set, they differ one from the other by the spread of the performances. For example the triangles related solution seems to be more spread on the objective space with respect to the circles related solution.

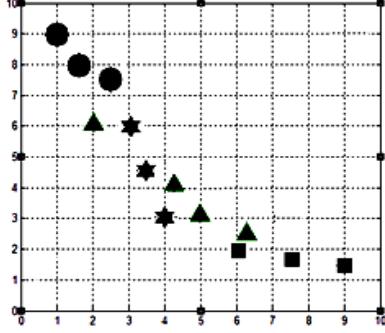


Figure 3: Four BW sets

Another observation is that the stars and the squares related solutions seem to be spread to the same distance, nevertheless, the spread with respect to each axis is different. The stars are spread a distance of 1 along objective 1 and 3 along objective 2 while the squares are spread 3 units along objective 1 and a half along objective 2. Naturally, reducing the spread as much as possible is desirable as it means less uncertainty *with respect to the worst cases*. It is not inherently so, but it is possible that there will not be a single 'best solution' which is involved with a minimal spread along all objectives axis (the probability for a single global optimal solution decreases as the number of the problem objectives increases). Thus it is possible that there will be solutions with less worst-case uncertainty than others in one objective and higher uncertainty in others. If the designers do not have a preference towards the objectives they might gain an understanding in which objective they would like to see less uncertainty influence. In other words they might have preferences as related to the robustness of the solutions to the uncertainties. In order to support the designers, this knowledge should be presented to them as part of a decision making support. In the following it is suggested to support decision making by minimizing the worst case uncertainty of the solutions.

3.3.2 Auxiliary MOP for decision making

The auxiliary MOP, which support decision making is formulized as follows:

(8)

$$\min(\Psi(S_w^x)), S_w^x \in BW,$$

$$\Psi(S_w^x) = [dis_1(S_w^x), dis_2(S_w^x), \dots, dis_K(S_w^x)]^T : S_w^x \rightarrow Z_x \subseteq Z \subseteq \Delta$$

$$\text{where } dis_i(S_w^x) = \max_{s^x \in S_w^x \in W} F_i(s^x) - \min_{s^x \in S_w^x \in W} F_i(s^x)$$

$$\begin{aligned} PS &:= \{x \in PS \mid \exists x' \in X : \psi(S_w^{x'}) \leq \psi(S_w^x)\} \\ FS &:= \{Z_x \in Z \mid Z_x = \psi(S_w^x) : x \in PS\} \end{aligned} \quad (9)$$

The solution to the auxiliary MOP belongs to the Auxiliary MOP's set PS, and front FS. Therefore, such a solution should be considered optimal and considered for selection. To elucidate this support approach, refer to Figure 4. In the Figure the uncertainties associated with the solution sets of Figure 3, are depicted, each along an objective axis. For example the circles related solution is associated with an uncertainty of 1.5 with respect to both objectives and therefore designated as a point [1.5, 1.5] in the auxiliary objective space.

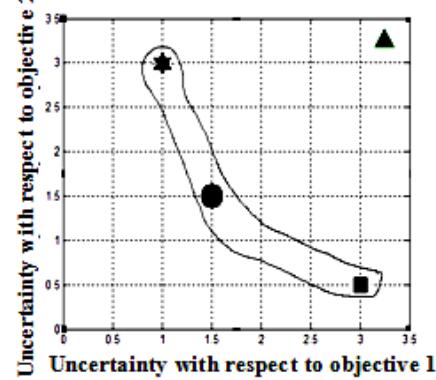


Figure 4: The performances of the sets of figure 3 in the auxiliary objective space.

From comparing Figures 3 and 4 it may be observed that the decision based on inspecting sets is transformed to a decision based on single representatives. Such a transformation reduces the complexity of the decision making by introducing the knowledge in a more compound manner. Figure 4 shows that although the triangle related set is part of the BW set, it is not optimal from the auxiliary MOP demand for optimality (minimizing the uncertainties). It is clear that if the robustness of the solutions to uncertainties associated with objective one is more important, than the star related solution should be chosen.

3.3 The overall procedure

The whole process of searching for best of the worst solutions and decision support as formulized and discussed above may now be summarized as follows. The realizations of a solution x (from Ω) combined with possible environmental situations (from Γ) end up with a set of sets of scenarios (in Φ). The performances of these scenarios are found by mapping them to the objective space (T). The worst cases of each solution (see Equation 6) which are found in the current paper by an embedded algorithm (see Section 3.4) are used by the EMO worst case optimization to find BW and BW_f (see Equation 7). To support decision making the sets of the best of the worst cases are mapped to the auxiliary MOP objective space (Ψ). The optimal set PS of the auxiliary MOP is presented to the designers to support a selection of a worst case robust solution.

It is noted that the suggested decision support approach is just one possible approach. It is performed posteriori to the evolution and therefore has no affect on it. Other presentations may also be considered (see future work in Section 5).

3.4 The embedded EMO algorithm

The procedure of searching the solutions is outlined in the following, where the embedded parts are highlighted in gray.

- a. Initialize a population P_t of size $n = |P_t|$ which decodes the design variables x . Also, set $Q_t = P_t$.
- b. Combine parent and offspring populations and create $R_t = P_t \cup Q_t$.

The Embedded Algorithm

- c. For each individual of R_t :

- c.1 initialize a population G_t of size $n' = |G_t|$ which decodes the possible range of d and of x .
- c.2 Run NSGA-II on the reversed optimization problem to find for each x its S_w^x

- d. While not all individuals of R_t are assigned with fitness:
 - e.1 Sort all worst sets for worst case dominance (Equation 2) to find the non-dominated front (see details in [7]).
 - e.2 Assign all the solutions of the front based on Equation 3, and assign boundary solutions with high fitness. Remove this front from the population.
- f. Initialize a new parent population $P_{t+1} = \emptyset$ of size n . Sort the solutions according to their Fit: $I_{Fit} = \text{sort}(Fit_x, >)$.
- g. Include the first n solutions of I_{Fit} 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.
- 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'.

The embedded part of the algorithm includes the search for the worst set for each of the candidate solutions by running an EMO (e.g., NSGA-II). It is noted again that the embedded MOEA optimizes the solutions for the reversed problem and therefore the algorithmic conditions should be adapted accordingly.

In the following section the algorithm is implemented for an academic example.

4. TEST CASE

The MOP in hand involves the minimization of:

$$F_1 = x_1 \quad (10)$$

$$F_2 = 1 + (x_2)^A - x_1 - 0.2 \cos(\pi x_1)$$

Here it is assumed that the uncertainty as related to the design variables (x_1 and x_2) may be described as:

$$x - 0.05 \leq x \leq x + 0.05$$

and the uncertainty involved with the environmental parameters (here it is on such parameter; A) may be assessed by:

$$1.98 \leq A \leq 2.02$$

To solve this problem by the embedded algorithm, populations of $n=100$ and $n'=30$ were used. A simple binary code with one point cross over (probability of 0.5) and mutation rate of 0.05 were used. Figure 5 depicts 6 out of the 200 sets associated with the individuals of R_t . Just 10 out of the 30 realizations' performances are depicted for each set (for clarity). It is depicted that the sets are forming clusters

of performances within the objective space. An entire set of such performances, of a solution, is depicted in Figure 6, designated by blank squares.

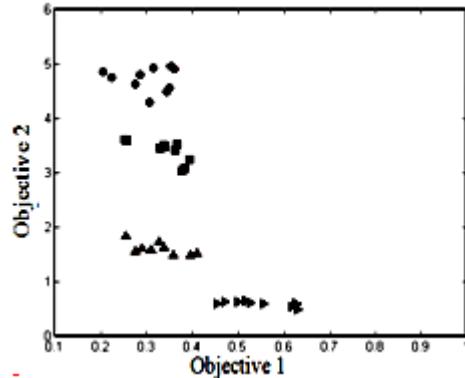


Figure 5: Several clusters (diluted) in the objective space. Each is associated with a solution.

The embedded algorithm evolves each of these clusters to find the worst cases set, which is depicted for the case of the cluster of Figure 6 as pluses in the same figure. Remember that the problem in hand involves a min-min problem and therefore the worst set is the Pareto front of the same problem, nevertheless posed as max-max problem.

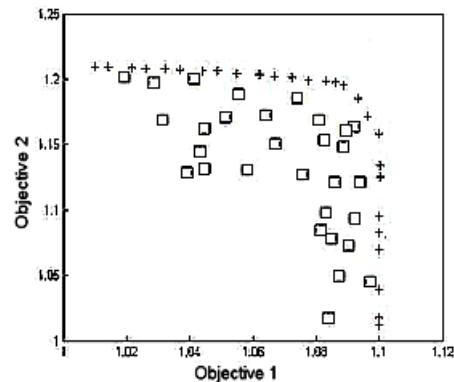


Figure 6: A cluster of performances and their related worst set found by the embedded EMO which is executed to search for the optimum of the reversed problem.

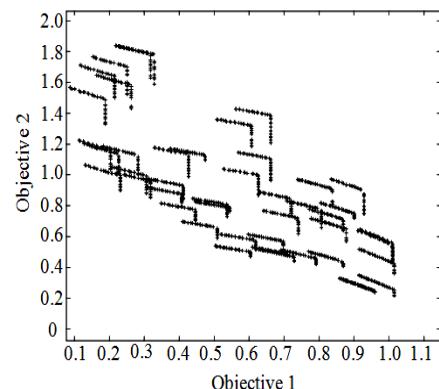


Figure 7: A set of worst case sets performances

Figure 7, depicts the result of the embedded algorithm, which is in the form of worst cases sets. The algorithm of [7], applies a search pressure such that the best of the worst front is evolved. This front for the current example is depicted in Figure 8.

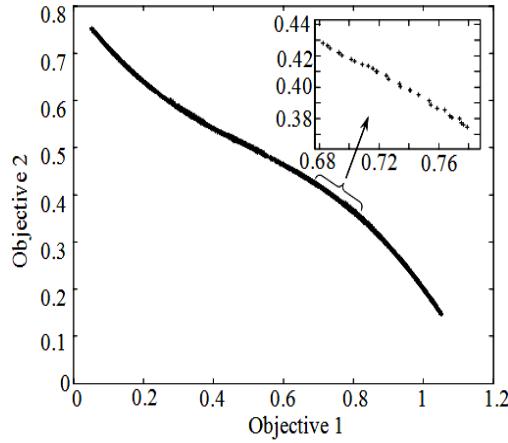


Figure 8: The evolved best of the

The best of the worst front appears as a thick front. This thickness is a result of the front being associated with $n \cdot n$ points (in the current example it is $100*30=3000$ points), which are the solutions' worst sets. One of these sets is depicted in the window within the Figure. It is already fairly clear that taking decisions based on such a representation is not as "simple" as by observing the common Pareto front.

The embedded EMO is computationally expensive. Running 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 complexity is profoundly higher than that of a common EMO. The complexity of NSGA-II is $O(GKn^2)$, where G is the number of generations, K is the number of objectives and n is the population size. The n^2 term is due to fitness assignment. 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 $O(GKn^2 + GnEK(n')^2)$. 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 the number of realizations used within the non-embedded part are increased; the complexity associated with the set-based comparisons is increased. This means that in order to reduce the complexity the following recommendations should be considered with respect to the embedded algorithm: 1. reduce the number of generations of the embedded algorithm (E), 2. reduce the size of the embedded algorithm population (n'). The affects of these recommendations on the results are reported in the following. Reducing E : In the current example the number of generations has been reduced from 60 (for the results of Figure 8) to 10 generations. The impact on the time reduction is clear, but what about the quality of the results? A zoom-in to the same region of the front for two different runs is depicted in Figure 9. In Figure 9, the squares designate the worst set of a solution obtained by using 60 generations for the embedded algorithm. On the other hand, the pluses designate a worst set of a solution obtained by setting the number of generations for the embedded algorithm to 10. It is clear from the figure, that most of the squares related set members

dominate those of the pluses set and not vice versa. This means that when running the embedded algorithm more generations, a higher search pressure is applied to find more optimal solutions in the best of the worst sense. This conclusion has been verified by testing with other sizes of populations and averaging over several runs.

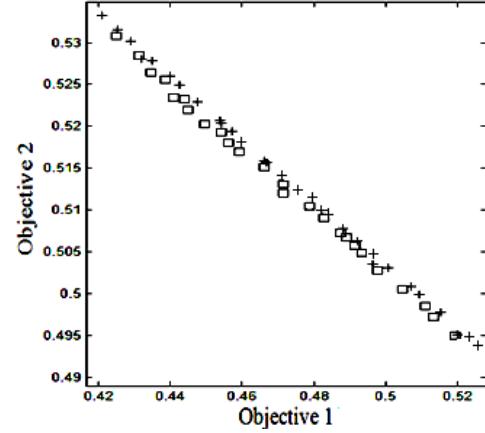


Figure 9: Two worst-case sets resulting from different evolutionary runs; When $E=60$ -squares, when $E=10$ -pluses.

Another interesting outcome of reducing the number of generations of the embedded algorithm has been spotted by depicting Figure 10. The uncertainties for the best of the worst sets (computed based on section 3.3.2), which were evolved by using $n'=10$ and $n'=60$ are depicted in the left and right panels of Figure 10 respectively.

It is observed that running the algorithm fewer generations (10 instead of 60) results in solutions, which their worst sets are more spread. Therefore not running the embedded algorithm long enough is not just associated with a reduced optimality with respect to the MOP but also with less optimality with respect to the auxiliary MOP!

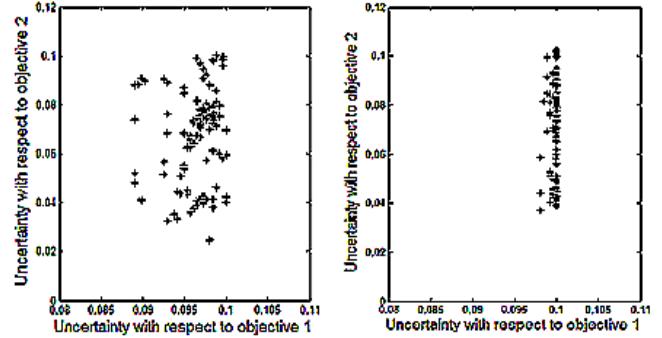


Figure 10: The best of the worst solutions' uncertainties with respect to the two objectives when running the embedded algorithm using $n'=10$ (left panel) and with $n'=60$ (right panel).

The improvement in quality of the front, as the number of generations/population size of the embedded algorithm increase, is evident from the above presented results. The actual affect of these increases on the running time is examined in the following two tests.

In the first test, the size of the embedded algorithm is kept on 30 individuals. These are run to search for the worst set of scenarios of

the solutions by utilizing different number of overall generations (i.e., 10 to 100 with jumps of 10 generations) of the embedded algorithm. The results are statistically evaluated over 20 runs of the entire algorithm for each number of generations and are depicted in the left panel of Figure 11. This panel describes the change of the normalized running time (normalized according to the longest running time) with respect to the number of the generations of the embedded algorithm.

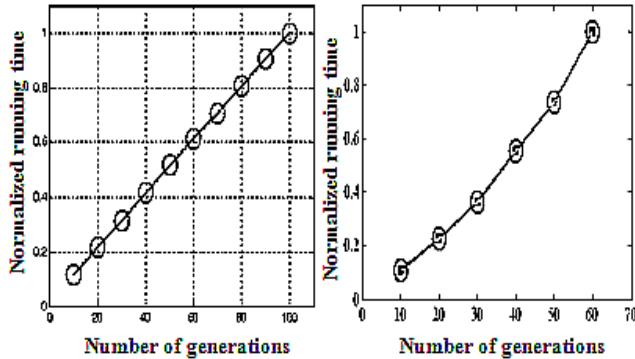


Figure 11: The affect of changing the number of generations of the embedded algorithm on the running time (left panel) and the affect of changing the number of individuals of the embedded algorithm on the running time (right panel).

The results, which are depicted in the figure, clearly indicate the linear relation between the increase in running time, with increase in the number of generations of the embedded algorithm.

In the second test, the number of generations used within the embedded algorithm is kept constant as 30 generations, and the number of individuals is altered from one run to the other. Tests were conducted on embedded algorithm's populations of 10 to 60 with jumps of 10 individuals. The results are statistically evaluated over 20 runs of the entire algorithm for each number of individuals and are depicted in the right panel of Figure 11. The right panel shows the change of the normalized running time (normalized according to the longest running time) with respect to the embedded algorithm's population's size. It may be comprehend from the right panel of Figure 11 that the running time increases with number of generations more rapidly than it does, when the number of generations, are increased.

Based on the above results a compromise between computational complexity and quality of results is considered. It is suggested to dynamically adjust the number of generations of the embedded algorithm. Such a possible change may be computed by:

$$n'_t = n'_0 + \frac{n'_f - n'_0}{n} t \quad (11)$$

where n'_t , n'_0 , n'_f , are the sizes of embedded algorithm populations at generation t (of the worst case EMO), at generation 1 and at generation n , respectively. Running the algorithm with $n'_0 = 5$, $n'_f = 60$ for the problem in hand resulted in reduction of the computation time by 40% (averaged over 20 runs) and with no evident reduction of quality with respect to running the algorithm with $n'_0 = 60$. It is clear that the results are good for the current simple problem and as suggested latter in the summary and conclusion section, these results are to be further investigated.

We have tested the algorithm on more complex cases. For example we have added uncertainties to the welded beam problem, which has been solved using NSGA-II by Deb et al., [18]. The problem 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 12 and are searched within the design space limits as follows: $0.25 \leq h, b \leq 5$ and $0.1 \leq l, t \leq 10$. The details of the problem might be found in [18]. Running the embedded algorithm and setting the uncertainty to zero results in the front depicted in the right panel of Figure 12.

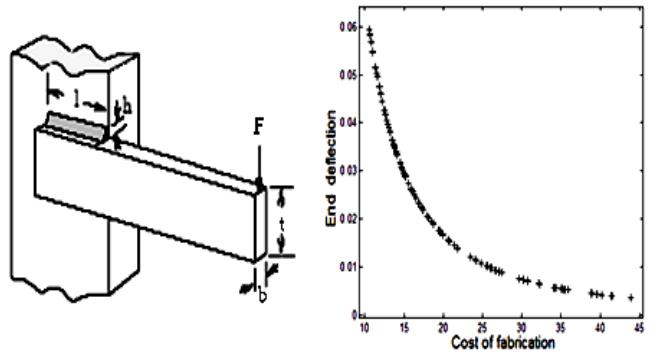


Figure 12: The welded beam design problem (left panel) and the related Pareto front, which has been evolved by setting the uncertainty to zero (right panel)

Here we assume that the parameters b , t are associated with an uncertainty such that $\Delta t = \pm 0.2$ and $\Delta b = \pm 0.09$. The BW front, which resulted from running the algorithm for that uncertainty, has some interesting aspects including: a. The BW distance from the nominal front (shown by dots in the figure) are changing along the design objective space, b. There is a distinct front in the auxiliary MOP, which might be correlated to the problem's objective space (see arrows in the figure), c. Using the two panels of Figure 13, a DM may take decisions on a solution based on his/her preferences towards the objectives supported by an understanding of the accompanying uncertainty.

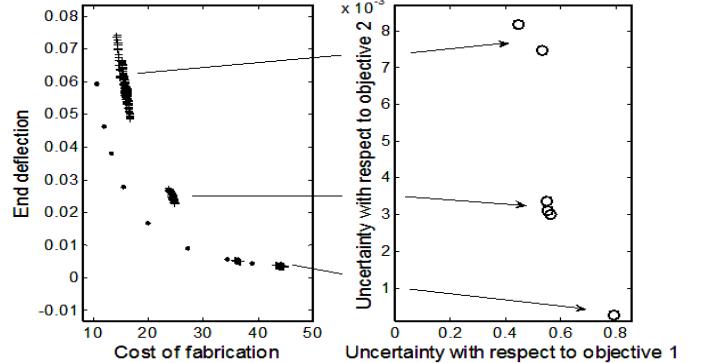


Figure 13: The BW front (left panel) and the auxiliary MOP front (right panel)

5. SUMMARY AND CONCLUSIONS

In this paper the recently introduced worst case EC is extended by embedding an EMO, which evolves the worst sets for each of the candidate solutions. These sets are then utilized by the worst case EMO to evolve the best of the worst front. The different worst sets and their related fronts are formulated in this paper. These fronts are defined such that they are the solution to the problem of finding the best performances under the worst scenarios' performances.

It is shown that the resulting front is associated with sets of solutions' performances and therefore with a set of points in the objective space for each solution. This makes the already not simple task of multi-criteria decision making to even a more complicated task. Instead of deciding on a solution based on a single point, the decision has to be made based on a set of such points. In order to support a decision, a decision support approach is suggested in this paper. It is implemented posteriori to the evolutionary run and therefore should be viewed as just one possible approach. In the suggested approach the decision is supported by posing the decision problem as an MOP with the objectives of minimizing the uncertainty with respect to all objectives. It is clear that the resulting auxiliary front is a support that should be considered together with other knowledge and other preferences of the designers. An alternative approach to support decisions might include displaying the average performance in each objective, plus spread of the solution's performances as additional objective, where the latter could be visualized by radius of a circle around the solution indicating uncertainty.

In the paper the implementation of the embedded algorithm is demonstrated for an academic and engineering example. Future work should test the approach on more elaborate real life engineering problems. Moreover the overall approach of the worst case EC should be further investigated with respect to the following issues: a. Influence of the size of the embedded population (n'). As explained in section 4 this should have a major influence on the computational time and performances of the algorithm, b. Developing measures that will allow the assessment of the quality of the resulting front. Such measures will open the way for conducting comparisons between the suggested algorithm and other approaches (e.g., searching for the worst by Monte Carlo simulations). Naturally such measures will allow assessing numerically the influence of changing the algorithmic parameters (e.g., size of population) on the quality of the resulting front.

6. REFERENCES

[1] Pareto, V. 1906: *Manuale di Economica Politica*, Societa Editrice Libaria. Milan; translated into English by A.S. Schwier as *Manual of Political Economy*, edited by A.S. Schwier and A.N. Page, 1971. New York: A.M. Kelley.

[2] Parmee I.C., The maintenance of search diversity for effective design space decomposition using cluster oriented genetic algorithms (COGAs) and multi-agent strategies (GAANT), *Proceedings of Adaptive Computing in Engineering Design and Control II*, University of Plymouth, pp: 128-138, 1996.

[3] Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2), 117-132.

[4] Avigad, G., and Deb, K. The Sequential Optimization-Constraint Multi-objective Problem and its Applications for Robust Planning of robot Paths. *Proceedings of the Congress on EC*, 25-29 September, Stamford, Singapore, (2007).

[5] Deb, K., and Gupta, H. Searching for robust Pareto-optimal solutions in multi-objective optimization, *Evolutionary Multi-Criterion Optimization*, volume 3410 of LNCS, pp: 150-164. Springer-Verlag, 2005.

[6] Daum, D.; Deb, K.; Branke, J.: "Reliability-based Optimization for multiple constraints with evolutionary algorithms". *Congress on Evolutionary Computation*, IEEE, 2007, S. 911-918

[7] Branke, J., Avigad, G., Moshaiov, A., Multi-objective Worst Case Optimization by Means of Evolutionary Algorithms, Technical Report, University of Karlsruhe, 2007.

[8] Mattson, C. A., and Messac, A. Pareto frontier Based concept selection under uncertainty, with visualization, *Optimization and Engineering*, 6: 85-115, 2005.

[9] Phadke, M.S. *Quality engineering using robust design*, Englewood Cliffs, New Jersey, Prentice Hall, 1989.

[10] Taguchi, G., Taguchi, S., and Tugulum, R., Taguchi robust engineering: worlds' best practice for achieving competitive advantage in the new millennium, McGraw -Hill, 1999.

[11] Andersson, J. Multiobjective optimization in engineering design applications to fluid power systems, *Ph.D. Thesis submitted to Linkoping University*, Linkoping, 2001.

[12] Branke, J. Reducing the sampling variance when searching for robust solutions. In et al., L. S., editor, *proceedings of Genetic and Evolutionary Computation Conference (GECCO'01)*, pp: 235-242. Morgan Kaufmann Publishers, 2001.

[13] Arnold, D.V., and Beyer, H.G. A comparison of evolution strategies with other direct search methods in the presence of noise, *Computational Optimization and Applications*, 24:135-159, 2003.

[14] Deb, K., and Gupta, H. Introducing robustness in multi-objective optimization. *KanGAL Report No. 2004016*, October, 2004.

[15] Luo, P.K Besharati, B.B., and Azarm, S. Design of robust new products under variability: marketing meets design, *Journal of Product innovative Management*, 22: 177-192, 2005

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

[17] Moshaiov, A., Multi-competence Cybernetics: The Study of Multiobjective Artificial Systems and Multi-fitness Natural Systems, In J. Knowles(ed.), D. Corne(ed.), K. Deb(ed.); *Multiobjective Problem Solving from Nature From Concepts to Applications* pp. 285-304, Springer, Berlin, 2008.

[18] Deb, K., Amrit, P., and Subrajyoti, M., Mechanical Component Design for Multiple Objectives Using Elitist Non-Dominated Sorting GA. *KanGAL Technical Report No. 200002*.

[19] Branke, J.; Deb, K.; Dierolf, H.; Osswald, M.: „Finding knees in multi-objective optimization“. In: *Parallel Problem Solving from Nature*, LNCS 3242, Springer, 2004, S. 722-731.