Solving Constrained Multi-Objective Problems by Objective Space Analysis

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

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

***Abstract*— In this paper, a new approach to solve constrained multi-objective problems by way of evolutionary multi-objective optimization is introduced. In contrast to former evolutionary approaches, which amalgamate objective space dominance relations with feasibility of solutions considered in the design spaces, the hereby suggested approach relies solely on objective space based analysis. It is shown in this paper that considering the violation of constraints within the design space is problematic as it may lead to misleading conclusions. Moreover, the current approach is inherently capable of dealing with constraints that are imposed directly in the objective space.**

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, Constraints, Reliability

1. INTRODUCTION

Approaches to solve constrained multi-objective problems by Evolutionary Multi-objective Optimization (EMO) are limited. Apart from the well-known approaches that utilize penalties to handle constraints by EMO (see e.g., [1]), most of the approaches are based on adapting the dominance relation between solutions to take into consideration the feasibility of the solutions. For example Jiménez and Verdegay [2] suggested a procedure which compares two solutions in a tournament selection. If one solution is feasible and the other is not, the feasible solution is chosen. If both solutions are infeasible, the solution closer to the constraints boundary is chosen. We note that here we regard the violation distance as a violation, which is considered in the design space. This is due to the fact that measuring the violation in that design may be done directly by considering the constraint boundary. Deb *et. al*., [3], defined a constrained domination principle, which differentiates infeasible from feasible solutions during the non-dominated sorting procedure.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

Viewing the background for the current research as described above, reveals some drawbacks and shortages of existing approaches as related to the solution of constrained MOPs by using EMO. These include the following:

1. None of the approaches deals directly with constraints, which are imposed within the objective space. An example for such constraints' imposing is the limiting of the budget and the allowed deflection of a truss. Such inherent objective space constraints should be treated during the evolution in order to efficiently use computational resources and to direct the search towards the unconstrained sub-space.
2. It is assumed that a small violation of the constraints based on the original constraints is directly related to a small violation in objective space (this assumption will be referred to as the correspondence relation). ***Such an assumption is wrong!*** Proving this may be done directly by introducing the following example. Let the design space consist of two variables $0<=x_1<5$ and $0<=x_2<5$. The design space is constrained by $g(x) = x_2+2x_1-2<0$, or alternatively $x_2<-2x_1+2$. The objectives of the problem are to minimize $f_2 = x_1$ and $f_2 = x_1 + (x_2 - 1)^2$. The feasible design space and its corresponding feasible objective space are designated by gray areas in the panels of Figure 1.

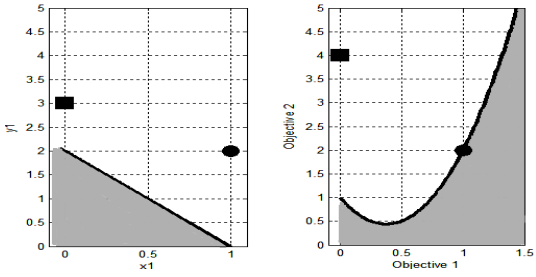

Figure 1: An example showing that the violation in design space (left panel) in not comparable to the violation in the objective space (right panel).

Two infeasible solutions are now considered. These are x_1= [0, 3] and x_2= [1, 2], which are designated in Figure 2 by a square and a circle, respectively. Both of these solutions are a result of changing

one of the variables by one unit. This means that considering the design space, the violation of constraints by both of the solutions is identical. Transforming the constraints and the solutions to the objective space results in the feasible region and the performances, which are depicted in Figure 2 (right panel). It is clear from the figure that the performances of x_2 just barely violate the objective space constraint whereas the performances of x_1 substantially violate the constraints.

The current paper tackles the above drawbacks by introducing a new approach. It involves the consideration of constraints in the objective space allowing an evolutionary search, which applies a pressure towards the evolution of a feasible optimal front within the constraints boundaries.

2. PROBLEM DEFINITION

A constrained MOP may be defined as follows:

$$\text{Find x which minimizes } F(x) = [f_1(x), f_2(x), ..., f_K(x)]^T \qquad (1)$$

Subject to:

$$g_i(x) \le 0, \quad i = 1,...., m$$

In the above definition all equality constraints are transformed into inequality constraints.

In the current paper the interest is on the assessment of performances and of constraint violation within the objective space. Moreover, there is an interest in solving problems where the constraints are directly posed within the objective space. As a result, the problem is reformulated as follows:

$$\text{Find x which minimizes } F(x) = [f_1(x), f_2(x), ..., f_K(x)]^T \qquad (2)$$

Subject to:

$$T(g_i(x)) \le 0, \quad i = 1,...., m$$

$$f_k(x) \le C_k \qquad k=1,\ldots K$$

The first constraint expression refers to the constraints of the design space, which are transformed to objective space. The second constraint expression refers to the limits of performances at each objective (objective space based constraints). A sub-set of all the solutions, which comply with the above constraints, may be defined. This set is termed here as the Constraint Envelope Set, CES, and its related performances set Constraint Envelope, CE, are defined as follows:

$$\begin{aligned} SCE := \{x_{ce} \in X | \forall\ i \in [1,...m] : T_i(x) = 0 \wedge \exists \varepsilon << 1 : T_i(x+\varepsilon) < 0 \\ \wedge \forall\ k \in [1,...K] : f_k(x) = C_k \wedge \exists \varepsilon << 1 : f_k(x+\varepsilon) < C_k \\ CE := \{y^* \in Y | y^* = F\ (x_{ce}) : x_{ce} \in SCE\} \end{aligned} \qquad (3)$$

In fact, equation 3 suggests that the CE is a set of solutions, whose performances lie on the boundary between feasibility and infeasibility, sorted in the objective space.

3. THE EVOLUTIONARY SEARCH

The following summarizes the influences that the performances of a solution should have, on the solution's fitness.

a. *A solution belonging to a lower level of non-dominance should be rewarded.* This demand allows applying a pressure towards the Pareto front.

b. *An un-crowded solution should be awarded.* Such a reward enhances the spreading of solutions along the constrained Pareto front. With that respect, boundary solutions should be maintained.

c. *A solution should be rewarded with relation to its distance from the CE (the closer, the higher the reward).* It is clear that a feasible solution is as close as can be and should be highly rewarded. Such rewarding should enhance a differential pressure towards feasible solutions.

On the basis of these desired influences, an EC algorithm is suggested. The algorithm, apart from the common elements of an EC algorithm (e.g., cross-over) is composed out of the following steps: a. Sort all solutions and than just the feasible solutions according to non-dominance and assign the solutions with fitness between upper and lower values for each level of non-dominance, taking the lower level for each solution. b. Spread the fitness of the solutions within each level, between the upper and lower values of the level, based on crowding distances, assigning the boundary solutions the highest fitness of the level. c. Find the CE of the problem (see equation 3). d. Penalize all non-feasible solutions based on their distance from the CE. This step will allow a higher pressure towards the constrained front as also depicted by the example of figure 2.

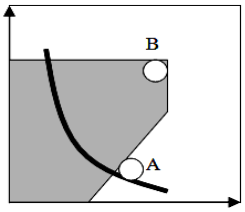

Figure 2: Preferring the solution associated with B (feasible solution) as done till today, over the solution associated with A may hamper the development of the feasible front as it drives the search away from it.

The results of the introduced algorithm have been compared with an algorithm suggested in [3] for several test cases. Averaging the results over three different examples and over 20 runs for both algorithms and using the proximity indicator, it is shown that the hereby introduced approach converges 7.1% faster than that of [5]. If mutation is not used in both algorithms, this grows up to 11.6%.

4. ACKNOWLEDGEMENTS

The second author acknowledges support from project 45863-Y.

REFERENCES

[1] Coello Coello, C.A., Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art, *Computer Methods in Applied Mechanics and Engineering*, **191**(11-12), pp. 1245-1287, January 2002.

[2] Jiménez, F. and Verdegay, J.L., Constraint Multiobjective Optimization by Evolutionary Algorithms, *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98)*, pp:266-271, 1998.

[3] Deb, K., Pratab, A. and Meyarivan, T., Constrained, Test Problems for Multi-objective Evolutionary Optimization, in E. Zitzler et al. (editors), First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01), pp. 284-298, Springer-Verlag, Lecture Notes in Computer Science Vol. 1993, 2001.