A New Approach to Robustness Analysis in Multi-Objective Optimization

Carlos Barrico[1,2] Carlos Henggeler Antunes[1,3]

[1] INESC Coimbra, Rua Antero de Quental 199, 3000-033 Coimbra, Portugal
{cbarrico, cantunes}@inescc.pt

[2] Dept. of Informatics, University of Beira Interior, 6201-001 Covilhã, Portugal

[3] Dept. of Electrical Engineering and Computers, Polo II, University of Coimbra, Portugal

1. Introduction

The main purpose of the study of a multi-objective optimization problem is often to characterize the set of non-dominated (Pareto optimal) solutions. However, some of these solutions, which could be of interest for a decision maker (DM) as acceptable compromise solutions, can be very sensitive to perturbations. Therefore, the need arises to offer the DM solutions that are relatively immune to perturbations in the decision variable space. That is, algorithms must strive for robust solutions. Some studies have been devoted to compute robust solutions both in single-objective [1, 2, 7, 9] as well as in multi-objective optimization [5, 6, 8].

In this paper, a new approach to robustness analysis in multi-objective optimization is proposed, involving the definition of a degree of robustness, which is based on the solution behavior in its neighborhood (see also the concept of robust solution of type II in [5]). This definition of degree of robustness permits that the user exerts a control on the desired level of robustness of the solutions obtained. Users can specify the size of the solution neighborhood, both in the decision variable space and in the objective function space. An evolutionary approach has been developed encompassing the definition of degree of robustness, which is applied to two test problems.

2. The degree of robustness

The assessment of the degree of robustness of a solution involves analyzing its neighborhood. That value depends on the size of a δ-neighborhood of solution x and the percentage of its h neighboring points whose objective function values belong to a η-neighborhood of f(x). Those h neighboring points are randomly generated around solution x (see also [5]).

The degree of robustness of solution x is a value k, such that: (a) the percentage of solutions in the kδ-neighborhood of x, whose objective function values belong to the η-neighborhood of f(x) is greater than or equal to p; (b) the percentage of solutions in the (k+1)δ-neighborhood of x, whose objective function values belong to the η-neighborhood of f(x) is lower than p.

The degree of robustness k of a solution x is gradually computed as k increases (neighborhoods δ, 2δ, …, kδ), as well as the number of neighboring points of x (h, 2h …, kh), such that in the tδ-neighborhood of x th neighboring points ($t \in \{1, \ldots, k\}$) are analyzed.

The degree of robustness contributes for the evaluation of a solution (individual) of a population and enables to classify the solutions according to their level of robustness.

3. The evolutionary algorithm

An evolutionary algorithm encompassing the definition of degree of robustness has been implemented, which uses an elitist strategy with a secondary population (with feasible non-dominated solutions only) of constant maximum size, consisting of the following main steps:

- The fitness of the individuals composing the main population is computed;
- From the main population (consisting of *POP* individuals) *POP-E* individuals are selected by using a tournament technique (*E* is the size of the elite set);
- A new population is formed by the *POP-E* offspring generated by crossover and mutation, and *E* individuals (elite) that are the more robust in the secondary population;
- The fitness of individuals is evaluated by a dominance test and by taking into account the degree of robustness, which defines an approximation to the non-dominated frontier;
- The non-dominated solutions are computed and they are processed to update the secondary population using a sharing technique, if necessary.

The population used in the algorithm implementation consists of individuals represented by an array of binary values. The initial population is randomly generated with feasible non-dominated solutions only. Uniform crossover and binary mutation have been used, with probability *pc* and *pm*.

The fitness value of a solution depends on its degree of robustness and the dominance test. For each solution the fitness computation uses a "non-dominated sorting" technique as in "NSGA-II" [3, 4], and involves determining various solution fronts in the following way:

- The first front consists of all non-dominated solutions, a minimum fitness value equal to $POP \times (MaxDegree+1)$ being assigned to them;
- This fitness value of each solution is incremented by the degree of robustness;
- The solutions in the first front are temporarily ignored and the remaining feasible solutions (the dominated solutions) are processed by applying them a dominance test (the non-dominated solutions will belong to the second front);
- The minimum fitness value of the current front is obtained by subtracting *MaxDegree+1* to the minimum fitness value of the previous front, which is assigned to the solutions of the current front;

- For each solution in the current front, the fitness value is incremented by the degree of robustness;
- This process continues until all feasible solutions are assigned a fitness value;
- The same process is repeated for the infeasible solutions.

If two solutions have the same fitness value, then the better solution is the one with fewer solutions in its neighborhood, according to a pre-defined radius.

The sharing mechanism uses the degree of robustness as the priority criterion in the selection of the individuals that will belong to the new secondary population.

4. Illustrative results

This approach involving robustness analysis to characterize the non-dominated front has been applied to two test functions:

P1: Minimize $(f_1(x), f_2(x)) = (x_1, h(x_1) + g(x)\,S(x_1))$,
Subject to $0 \le x_1 \le 1$, $-1 \le x_i \le 1$, $i = 2, \ldots, 5$
Where $h(x_1) = 1 - x_1^2$
$g(x) = \sum_{i=2,\ldots,5} (10 + x_i^2 - 10\cos(4\,\pi\,x_i))$
$S(x_1) = 1 / (0.2 + x_1) + x_1^2$

P2: Minimize $(f_1(x), f_2(x)) = (x_1, h(x_2)\,(g(x) + S(x_1)))$,
Subject to $0 \le x_1, x_2 \le 1$, $-1 \le x_i \le 1$, $i = 3, \ldots, 5$
Where $h(x_2) = 2 - 0.8\exp(-((x_2 - 0.35) / 0.25)^2) - \exp(-((x_2 - 0.85) / 0.03)^2)$
$g(x) = \sum_{i=3,\ldots,5} (50\,x_i^2)$
$S(x_1) = 1 - x_1^{1/2}$

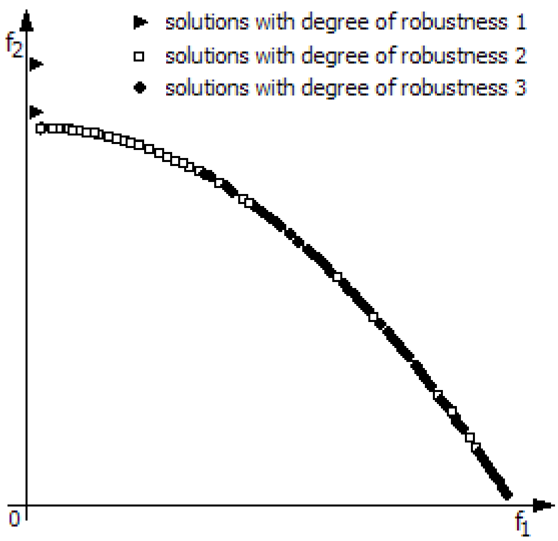

Figure 1.

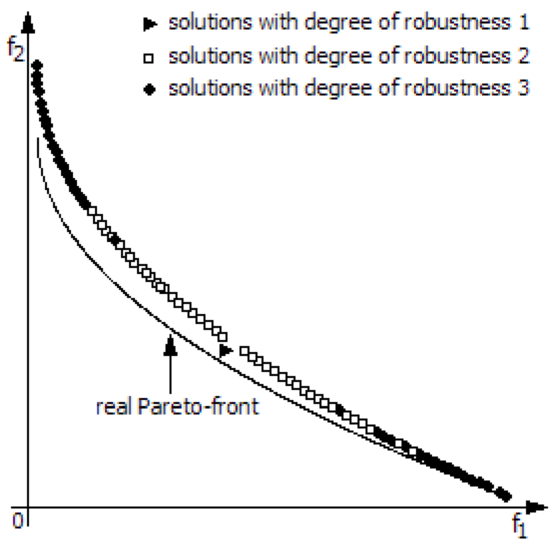

Figure 2.

Figure 1 shows the non-dominated front obtained for P1, with $\eta = 0.7$, $h = 100$, and $\delta = 0.002$.

The less robust solutions are located near the optimum of f_1. Figure 2 shows the non-dominated front obtained for P2, with $\eta = 0.7$, $h = 100$, and $\delta = 0.01$. In this case, the more robust solutions are located in the extremes, that is close to the optima of both f_1 and f_2.

5. Conclusions and Future Work

This paper presented a new approach to robustness analysis in multi-objective optimization problems. The concept of degree of robustness in incorporated into the evolutionary algorithm, particularly in the computation of the fitness value of the solutions. This approach also enables to classify the solutions of the Pareto-front according to the degree of robustness, not just classifying solutions as robust or not robust (such as in [5]). Research is underway to extend this approach to problems in which uncertainty lies on the model coefficients and it can be represented by means of interval numbers.

References

[1] Branke, J. (1998). “Creating Robust Solutions by means of an Evolutionary Algorithm”. In: Parallel Problem Solving from Nature, vol. 5, Lecture Notes in Computer Science. Springer, Berlin. 119-128.

[2] Branke, J. (2000). “Efficient Evolutionary Algorithms for Searching Robust Solutions”. In: ACDM. 275-286.

[3] Deb, K. (2001): Multi-Objective Optimization Using Evolutionary Algorithms. Wiley.

[4] Deb, K. and Pratap, A. and Agarwal, S., and Meyarivan, T. (2002): “A fast and elitist multi-objective genetic algorithm: NSGA-II”. In: IEEE Trans. Evolutionary Computation, 6 (2). 182-197.

[5] Deb, K. and Gupta, H. (2004): “Introducing Robustness in Multiple-Objective Optimization”. KanGAL Report Number 2004016. Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India.

[6] Hughes, E. J. (2001). “Evolutionary Multi-Objective Ranking with Uncertainty and Noise”. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization. 329-343.

[7] Jin, Y. and Sendhoff, B. (2003). “Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach”. In: Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization, LNCS 2632, Springer. 237-251.

[8] Teich, J. (2001). “Pareto-Front Exploration with Uncertain Objectives”. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization. 314-328.

[9] Tsutsui, S. and Ghosh, A. (1997). “Genetic Algorithm with a Robust Solution Searching Scheme”. In: IEEE Transactions on Evolutionary Computation, 1 (3). 201-219.