Robust Multiobjective Optimization using Regression Models and Linear Subproblems

Fillipe Goulart
Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil 31270-901
fgsm-c@ufmg.br

Sílvio T. Borges
Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil 31270-901
silviotassini@gmail.com

Fernanda C. Takahashi
Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil 31270-901
fctakahashi@gmail.com

Felipe Campelo*
Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil 31270-901
fcampelo@ufmg.br

ABSTRACT

We propose a technique for incorporating robustness as part of the search process of evolutionary multiobjective optimization algorithms. The proposed approach calculates the sensitivity of candidate solutions by solving a linear programming subproblem, defined by regression models fitted using points in the neighborhood of each candidate solution. This sensitivity information is then used as part of the selection process, to drive the search towards solutions that comply with robustness requirements defined a priori by the decision-maker. Preliminary results suggest that this approach is capable of correctly converging to the desired robust fronts.

CCS CONCEPTS

• Applied computing → Multi-criterion optimization and decision-making; • Theory of computation → Evolutionary algorithms;

KEYWORDS

Multiobjective optimization, robust optimization, hybrid algorithms

ACM Reference format:
Fillipe Goulart, Sílvio T. Borges, Fernanda C. Takahashi, and Felipe Campelo. 2017. Robust Multiobjective Optimization using Regression Models and Linear Subproblems. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3079191

*Corresponding author.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
GECCO '17, Berlin, Germany
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4920-8/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3071178.3079191

1 INTRODUCTION

Let a continuous multiobjective optimization problem (MOP) be defined as:

$$\begin{aligned} \text{minimize: } & \mathbf{f}(\mathbf{x}) \\ \text{subject to: } & \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n \end{aligned} \tag{1}$$

with $\mathbf{f}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$ representing a vector of objective functions to be minimized, and $\mathcal{X}$ the set of feasible candidate solutions:

$$\mathcal{X} \triangleq \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\} \tag{2}$$

with $\mathbf{g}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^p$ representing a vector of constraint functions.[1]

Multiobjective optimization techniques are generally employed to provide a *decision maker* (DM) with enough information for choosing the most adequate solution for a given problem [6]. In applied optimization scenarios, there is oftentimes the complicating factor that the problem is subject to uncertainties [2], which can be due to errors in measurement or estimation, or to manufacturing tolerances and imperfections. As a consequence, it is important to consider the possibility that small perturbations to an optimal solution may result in severe performance degradation, or even make a seemingly good solution unfeasible. Therefore, methodologies that are capable of finding *robust* solutions, which are less sensitive to perturbations, are of great value when tackling applied optimization scenarios.

While several different types of uncertainties can appear in optimization scenarios [4], this work is focused only on uncertainties in the variables $\mathbf{x}$, which usually arise due to manufacturing tolerances. In this context, a robust solution can be understood as one that presents small enough variations in objective function values when subject to uncertainties in the variables [4]. In general, there have been two main approaches for handling uncertainties in the variables [15]: replacing the objective functions by *effective functions*, and/or adding a *variability measure*.

[1]Throughout this paper, vector comparisons such as $\mathbf{a} < \mathbf{b}$ are defined as $a_i < b_i,\ \forall i$.

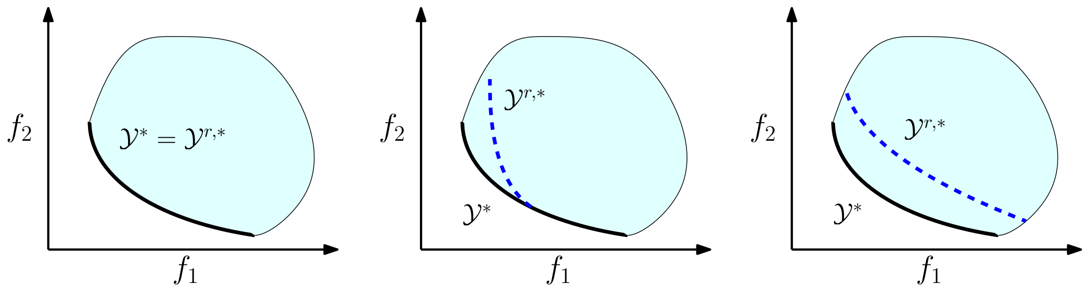


Figure 1: Robust efficient fronts can completely overlap; or differ in part (center) or in whole (right) from the non-robust front. Figure inspired by the discussion presented in [8].

In the effective function approach, each objective function $f_i(\cdot)$ is replaced by $f_i^{eff}(\cdot)$, which returns values for a given solution $\mathbf{x} \in \mathcal{X}$ that are influenced by the objective function values of points within a given neighborhood of $\mathbf{x}$. In the literature [8], this effective function usually appears as the mean value of $f_i(\cdot)$ within a neighborhood $\mathcal{N}(\mathbf{x}, \Delta\mathbf{x})$, or alternatively as its worst-case value within it (an approach also known as *robust regularization* [13]). This neighborhood set can be a ball of radius Δx or, more commonly, a box:

$$\mathcal{N}(\mathbf{x}, \Delta\mathbf{x}) = \{\mathbf{x}' \mid \mathbf{x} - \Delta\mathbf{x}/2 \leq \mathbf{x}' \leq \mathbf{x} + \Delta\mathbf{x}/2\} \tag{3}$$

Since computing the true mean or finding the true worst-value within $\mathcal{N}(\mathbf{x}, \Delta\mathbf{x})$ is usually impractical, this approach commonly uses approximated values, by sampling a number of points $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}, \Delta\mathbf{x})$, $j \in \{1, 2, \ldots, N\}$, and using either the sample mean or maximum of the observed values for each objective function.

The effective function approach is very simple in concept and, at least in principle, requires no modification in the optimization algorithm. However, the only parameter that controls the robustness, namely the uncertainty in the variables $\Delta\mathbf{x}$, is usually not adjustable and, as a result, the decision maker is unable to determine a robustness threshold for the solutions.

Another approach, which is sometimes used to address this issue, consists of computing a vector of variability indicators $\mathbf{s}(\cdot)$ (or a pooled variability measure $s(\cdot)$) to approximate the solution sensitivity, which can later be incorporated in the algorithm either as additional constraints or objective functions. Jin and Sendhoff [11] proposed to estimate the robustness with (4):

$$s_i(\mathbf{x}) = \frac{1}{n}\sum_{j=1}^{n} \frac{\sigma_{f_i}}{\sigma_{x_j}}, \quad \begin{array}{rcl} i & = & 1, \ldots, m \\ j & = & 1, \ldots, n \end{array} \tag{4}$$

where σ_{f_i} is the standard deviation of the i-th function, and σ_{x_j} the standard deviation of the j-th variable of $\mathbf{x}$. The authors proposed to estimate these values using points available in the population, in the case of population-based metaheuristics.

Deb and Gupta [8] proposed the use of a different function:

$$s(\mathbf{x}) = \frac{\|\mathbf{f}^{eff}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\|}{\|\mathbf{f}(\mathbf{x})\|} \tag{5}$$

Both (4) and (5) represent sensitivity measures, and a solution $\mathbf{x}$ is considered more robust for smaller values of $s_i(\mathbf{x})$ or $s(\mathbf{x})$.

If these measures are included as constraints in the MOP, the DM can define a threshold vector $\boldsymbol{\eta}$ (or a scalar η, if a pooled measure is used) such that solutions $\mathbf{x}$ with sensitivity greater than the threshold are considered unfeasible, thereby driving the search towards *robust enough* solutions. The requirement that solutions should be less sensitive than a threshold results in (robust) efficient fronts $\mathcal{Y}^{r,*}$ that may diverge from the original $\mathcal{Y}^*$, as illustrated in Fig. 1.

1.1 Literature Overview

While single-objective robust optimization has received much attention [2, 4, 12], the published literature dealing with robust multiobjective optimization problems (RMOPs) is more modest. In this section we describe the most representative works in this area.

Jin and Sendhoff [11] included a variance measure as an additional objective. Instead of sampling neighboring points to compute this measure, information available from the population at a given iteration is used to estimate the standard deviations in (4). An Evolution Strategies method is used to solve the resulting problem. The main drawbacks of this approach are its dependence on the population size; the strong underlying assumption that the population is able to correctly approximate local characteristics; and the doubling of the number of objectives, which can make the optimization and decision-making processes considerably more complex.

Gaspar-Cunha and Covas [9] addressed some of these problems by adopting an approach to optimize the original vector function $\mathbf{f}(\cdot)$ together with sensitivity measures (4). Sensitivities were estimated using points from the current population, augmented with the initial population for increased diversity in the space of decision variables. To reduce the resulting number of objective functions, the authors proposed the use of a global sensitivity index $s(\cdot)$, computed by taking the average (or the maximum) of the individual $s_i(\cdot)$ components. The resulting problem presents $m + 1$ objectives, instead of $2m$. Despite these improvements, the sensitivity estimation is still heavily dependent on the population size and its ability to adequately represent the neighborhood of each solution.

Deb and Gupta [8] presented two ideas for solving RMOPs. The first, referred to as *type-I robustness*, is essentially an effective function approach. The second, named *type-II robustness*, includes a variation measure (5) as an additional constraint, so that the problem becomes:

$$\begin{aligned}\text{minimize: } & \mathbf{f}(\mathbf{x}) \\ \text{subject to: } & \frac{\|\mathbf{f}^{eff}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\|}{\|\mathbf{f}(\mathbf{x})\|} \leq \eta \\ & \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n\end{aligned} \tag{6}$$

i.e., the overall percent variation must lie below a specified threshold η. The authors also propose to sample a number N of neighboring solutions for each $\mathbf{x}$ to approximate the effective function or the variation measure. The resulting problem then is solved using an NSGA-II.

Despite its promising results, some drawbacks are present in this method: the use of a single threshold for all objectives, which does not allow the possibility of the DM having different threshold values for each objective and complicates the definition and interpretation of the threshold value; and the high computational cost due to the large number of neighbors required to accurately approximate the effective function or the variation measure. This latter issue was addressed by Saha *et al.* [18], who employ a *type-II robustness* definition and store an archive containing up to 100μ previously evaluated points (with μ being the population size). When computing the effective function for a given point, N points are generated using Latin Hypercube Sampling (LHS), and the archived candidate solutions that are closest to the LHS points are used, without evaluating new points.

Finally, Barrico *et al.* [1] proposed the concept of *degrees of robustness*, which are combined with the front level of each solution in a non-dominated sorting approach. The authors argue that this information can assist the DM with trade-off analysis, considering also the robustness levels. However, the method requires a large number of additional function evaluations, which may exhaust the computational budget of the optimization method before a good convergence can be achieved.

In all the reviewed studies, the biggest challenge is arguably the reliable computation of a robustness measure without the need of a large number of neighboring points. Although this sampling of new solutions seems to be unavoidable [5], it is still possible to consider techniques for reducing the required sample size. In the following section we present an approach that employs linear programming subproblems, defined by regression models, for the evaluation of the sensitivity of candidate solutions. This approach is easily incorporated in existing multiobjective evolutionary algorithms (MOEAs), and its performance is assessed using functions for which the true robust fronts can be extracted analytically.

2 PROPOSED METHOD

We argue that using robust regularization is more interesting than using the mean for the effective function since, from an engineering perspective, one is usually more interested in guaranteeing worst-case scenarios than on the average performance degradation. Moreover, as the DM may have different robustness thresholds for each objective, we adopt the *type-II robustness* approach of Deb and Gupta [8], but considering individual sensitivities for each objective as:[2]

$$s_i(\mathbf{x}) \triangleq \frac{|f_i^{eff}(\mathbf{x}) - f_i(\mathbf{x})|}{|f_i(\mathbf{x})|} \tag{7}$$

The RMOP is then expressed as:

$$\begin{aligned}\text{minimize: } & \mathbf{f}(\mathbf{x}) \\ \text{subject to: } & \mathbf{s}(\mathbf{x}) \leq \boldsymbol{\eta},\ \boldsymbol{\eta} \in \mathbb{R}_+^m \\ & \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n\end{aligned} \tag{8}$$

As uncertainty information is usually available for each x_j, $j \in \{1, 2, \ldots, n\}$, we employ box neighborhoods (3) when calculating $\mathbf{s}(\mathbf{x})$. It is also important to mention that the uncertainties $\Delta\mathbf{x}$ are, in most cases, not freely adjustable. This arises from the fact that they are interpreted as representing constructive or operational uncertainties that arise from the specific application problem to which this approach is to be applied. These values should be furnished by the decision maker, which is assumed to possess specialist knowledge on the uncertainty profile of the problem.

2.1 Computing the effective function

The usual approach for computing $f_i^{eff}(\mathbf{x})$ is to sample N points within $\mathcal{N}(\mathbf{x}, \Delta\mathbf{x})$ and use the largest observed value of each objective [8]. Depending on the dimension of the problem and the size of $\Delta\mathbf{x}$, this can require very large samples, which can become prohibitive, e.g., in simulation-based engineering problems.

Our proposed strategy for reducing this number is to employ a linear approximation of each objective function. If we assume the size of the uncertainties is relatively small (which is common in most manufacturing processes), the quality of the approximation is expected to be reasonable even with small sample sizes. Assuming $\mathcal{P}$ is a set containing the points whose sensitivities we wish to compute, the process is performed as follows:

(1) For each $\mathbf{x} \in \mathcal{P}$, generate N solutions in the neighborhood $\mathcal{N}(\mathbf{x}, \Delta\mathbf{x})$. Let $\mathcal{P}_\mathbf{x}$ be the set with $\mathbf{x}$ and its neighbors;
(2) For each $f_i(\cdot), i \in \{1, 2, \ldots, m\}$,
   (a) Fit a first-order linear regression model using the points in $\mathcal{P}_\mathbf{x}$. Let $\mathbf{c}$ be the resulting vector of coefficients;
   (b) Solve the linear program, (e.g., using the Simplex method):

$$\begin{aligned}\text{maximize: } & \mathbf{c}^T\mathbf{y} \\ \text{subject to: } & \mathbf{y} \geq \mathbf{x} - \Delta\mathbf{x}/2 \\ & \mathbf{y} \leq \mathbf{x} + \Delta\mathbf{x}/2\end{aligned} \tag{9}$$

[2]If any $f_i(\cdot)$ has a significant chance of approaching zero, an alternative formulation using only the numerator of (7) can be adopted instead.

(c) Let $\mathbf{y}_l^*$ be the solution to (9). The value for the i-th effective function is calculated as:

$$f_i^{eff}(\mathbf{x}) = \max\left(f_i(\mathbf{y}_l^*),\ \max_{\mathbf{x}\in\mathcal{P}_\mathbf{x}} f_i(\mathbf{x})\right) \tag{10}$$

(3) Compute $\mathbf{s}(\mathbf{x})$ by using (10) in (7) .

Figure 2 illustrates this approach to the calculation of the effective function value.

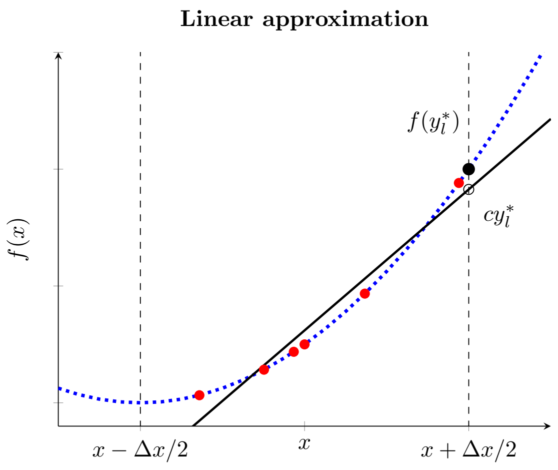


Figure 2: Computing $f^{eff}(\mathbf{x})$ by maximizing a linear regression model. The actual objective function is shown as a dotted blue curve, sampled points are indicated as red circles, the linear regression model is shown as a solid black line. The hollow black circle indicates the solution of the linear program, and the solid black dot marks the resulting effective function value.

2.2 Including robustness in MOEAs

In principle, the problem defined in (8) can be solved by any algorithm equipped with a constraint handling method [14]. We propose a simple structure which can be adapted to a variety of evolutionary methods:

(1) Let $\mathcal{R}(t)$ be the set containing both the current population, $\mathcal{P}(t)$, and the candidate solutions generated after the variation operators, $\tilde{\mathcal{P}}(t)$, at iteration t. Define $\mathcal{R}_r(t) \subseteq \mathcal{R}(t)$ as:

$$\mathcal{R}_r(t) = \{\mathbf{x} \in \mathcal{R}(t) \mid \mathbf{s}(\mathbf{x}) \leq \boldsymbol{\eta}\}$$

(2) If $|\mathcal{R}_r(t)| < \mu$, with μ denoting the population size, then insert solutions from $\mathcal{R}(t) \setminus \mathcal{R}_r(t)$ in increasing order of sensitivity (i.e., less sensitive first). If $|\mathcal{R}_r(t)| > \mu$, then remove solutions using additional criteria (e.g., nondominated sorting).

(3) Assemble the new population: $\mathcal{P}(t+1) \leftarrow \mathcal{R}_r(t)$.

With this structure we first guarantee robustness and then, among the robust points, we compute the best ones in terms of dominance. In the end, we have an approximation of the robust efficient front at a given threshold.

The proposed structure is general enough to be included in almost any MOEA. In this work, we use the Differential Evolution for Multi-objective Optimization (DEMO) [17] as the base algorithm, generating a "robust DEMO" (r-DEMO). Since it performs paired comparisons between each current solution and its specific challenger in its selection phase, we can specialize this step and include the robustness information as follows:

(1) For each pair $\left(\mathbf{x} \in \mathcal{P}(t);\ \tilde{\mathbf{x}} \in \tilde{\mathcal{P}}(t)\right)$:
- If both points have $\mathbf{s}(\cdot) > \boldsymbol{\eta}$, include the one with the lowest $\mathbf{s}(\cdot)$ in $\mathcal{R}_r(t)$;
- If only one point has $\mathbf{s}(\cdot) \leq \boldsymbol{\eta}$, include only this point in $\mathcal{R}_r(t)$;
- If both points have $\mathbf{s}(\cdot) \leq \boldsymbol{\eta}$, include the one that Pareto-dominates the other in $\mathcal{R}_r(t)$. If they are mutually non-dominated, then include both.

(2) If $|\mathcal{R}_r(t)| > \mu$, remove candidate solutions using non-dominated sorting, crowding distance, or indicators.

3 RESULTS

For the initial assessment of the proposed method, the test problem generator defined in [8] was employed:

$$\begin{aligned} \text{minimize: } & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T \\ \text{subject to: } & 0 \leq x_1 \leq 1 \\ & -1 \leq x_j \leq 1,\ j = 2, 3, \ldots, n \end{aligned} \tag{11}$$

with:

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 + 1 \\ f_2(\mathbf{x}) &= h(x_1) + g(\mathbf{x})S(x_1) + 1 \\ h(x_1) &= 1 - x_1^2 \\ g(\mathbf{x}) &= \sum_{i=2}^{n} 10 + x_i^2 - 10\cos(4\pi x_i) \\ S(x_1) &= \frac{\alpha}{0.2 + x_1} + \beta x_1^2 \end{aligned} \tag{12}$$

where α and β can be used to control the sensitivity of the problem, and a (+1) offset was added to each objective to guarantee that both functions are always strictly positive, so that the percent version of (7) could be tested.

3.1 Comparing the sensitivity computation

The first experiment was performed to compare the performance of the linear approximation approach (*linear*) against the usual strategy of using the worst observed function value (*wov*) out of N samples for the sensitivity estimation. For that, we randomly generated $N_r = 500$ points $\mathbf{x}_j$ within the feasible space of (11) with $\alpha = \beta = 1$, and computed $\mathbf{s}(\mathbf{x}_j), \forall\, j \in \{1, \ldots, N_r\}$ using both the *linear* and *wov* strategies. The values were then compared to the true sensitivity value[3] $\mathbf{s}_{\text{TRUE}}(\mathbf{x}_j)$. This design was replicated in the following conditions: test problem dimension $n = \{2, 5, 10, 20\}$, uncertainty size $\Delta x = \{0.005, 0.01, 0.05, 0.10\}$ (representing uncertainties from 0.5% to 10% of the range of each variable),

[3]The true sensitivity can be obtained by maximizing each function (12) within the interval $\mathbf{x} \pm \Delta\mathbf{x}/2$. For this particular case, this was done analytically.

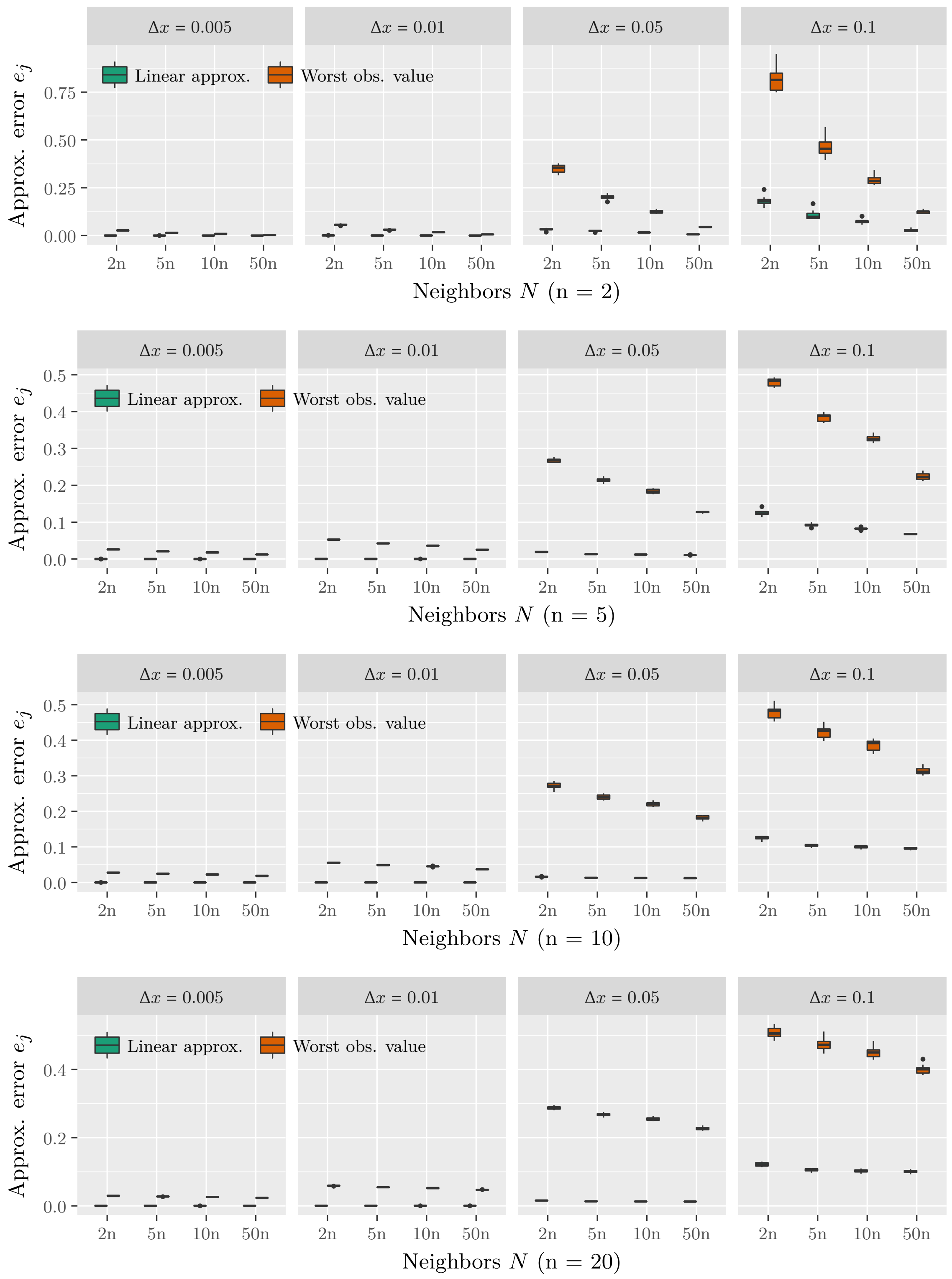


Figure 3: Comparing the *linear* and *wov* approaches in the computation of the sensitivity index on $f_2(\cdot)$. Notice that *linear* (left boxplots in each neighborhood size) is consistently more accurate than *wov* (right boxplots) in the evaluation of the sensitivity index for every dimension n, number of neighbors N and uncertainty range Δx. Only $f_2(\cdot)$ is illustrated here, since $f_1(\cdot)$ is a linear function and would suggest an unfair advantage to *linear*.

and number of neighbors $N = \{2n, 5n, 10n, 50n\}$. All combinations of these factors were replicated 10 times. For each $\mathbf{x}_j$, the absolute error $\mathbf{e}_j = \left|\mathbf{s}_{\text{TRUE}}(\mathbf{x}_j) - \mathbf{s}(\mathbf{x}_j)\right|$ was computed, and the results are shown in Figure 3.

As expected, it is evident that both the *linear* and *wov* strategies improve as more neighbors are used for the sensitivity calculations, although this improvement is more modest in higher dimensions. It is also very clear that the error is much smaller for the proposed method in this particular problem – the performance of the *linear* approach for $N = 2n$ neighbors is consistently as good as, or even better than, the *wov* performance for $N = 50n$. Even though *linear* requires the effort of model fitting and the solution of the resulting linear program, this is more than compensated by enabling far more accurate sensitivity estimation using a far smaller number of points, an advantage that is particularly important in expensive optimization problems.

Figure 3 also indicates that both techniques lose accuracy as the uncertainties grow, as was also expected. Again, *linear* presents errors that are consistently much smaller than *wov*. These results suggest that the proposed method can be employed with relatively small values of N and for uncertainties up to 10%, and still generate good estimations of solution robustness, at least for problems similar to (11).

Finally, a complete block design model [16] was fit, using the sensitivity evaluation method as the experimental factor, and problem dimension n, number of neighbors N and uncertainty range Δx as blocking factors. The analysis of this model confirms not only the statistically significant difference between the two methods ($p < 2 \times 10^{-16}$), but also indicates the magnitude of the observed difference, with *linear* presenting a mean approximation error (across all conditions tested) about 14% smaller than *wov* ($CI_{.95} = 0.143 \pm 0.029$). It is also interesting to observe that this difference is lower for smaller uncertainties, but gets larger as Δx is increased, as can be easily seen from Fig. 3.

3.2 Computing robust fronts

As mentioned in the Introduction, using the concept of variability to model robustness results in distinct robust Pareto-fronts for different threshold values $\boldsymbol{\eta}$ (see Fig. 1). For instance, considering the test problem (11) with $\alpha = \beta = 1$, Figure 4 illustrates a sampled version of the feasible objective space, together with its robust solutions and efficient fronts for different values of η (set equal in both objectives for simplicity).

In this experiment, we employed the test problem (11) with dimension $n = 10$, uncertainties $\Delta\mathbf{x} = [0.01\ 0.02\ 0.02\ \ldots\ 0.02]^T$ (indicating 1% of uncertainty in terms of the range of each variable), and $N = 2n$ neighbors to estimate the sensitivity with the proposed approach. The top panel of Figure 5 shows the true sensitivity of both objectives for points belonging to the efficient set of (11) with $\alpha = \beta = 1$. Depending on the threshold value η, different portions of the Pareto-front are no longer robust (see Figure 4). This provides a baseline to evaluate the performance of the r-DEMO for this problem, as illustrated in the bottom panel of Figure 5.

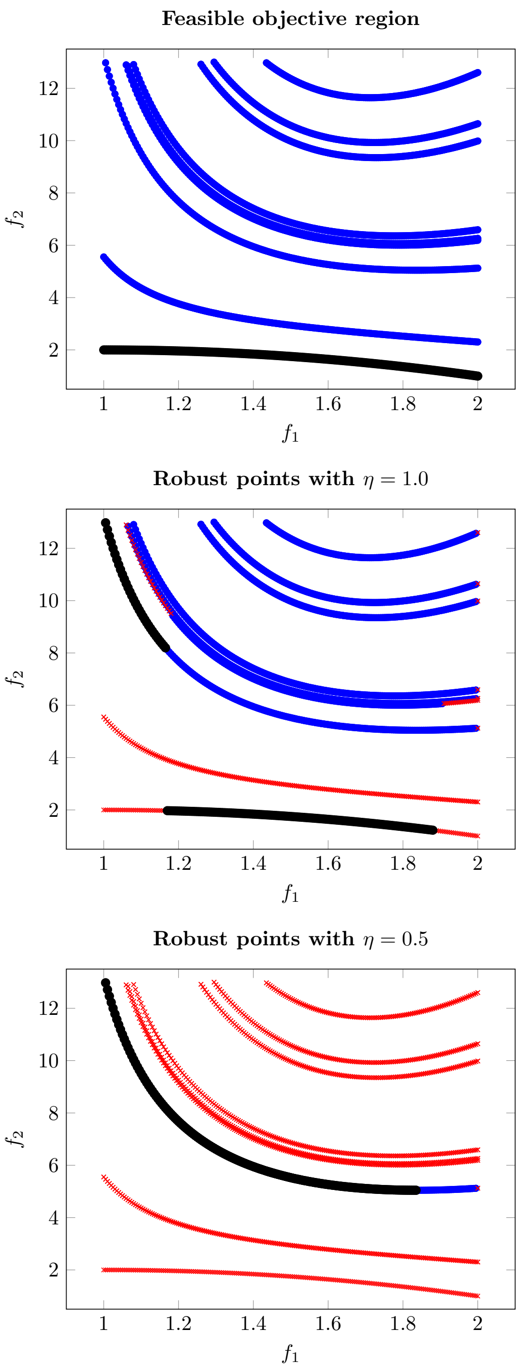
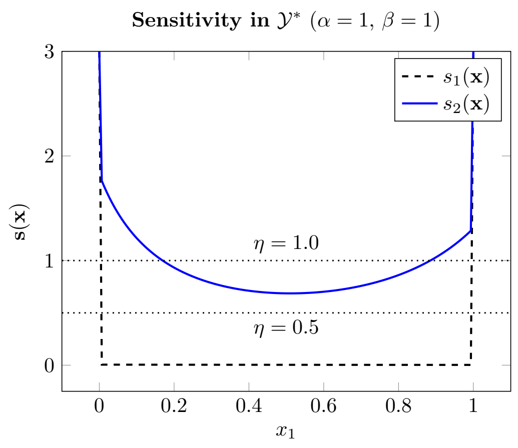


Figure 4: Robust solutions for distinct values of η. Blue circles and red crosses indicate robust and non-robust solutions, respectively. Black disks represent the robust efficient front. Top: discretized objective set and Pareto front without sensitivity constraints. Middle: robust solutions for $\eta = 1$. Bottom: robust solutions for $\eta = 0.5$.



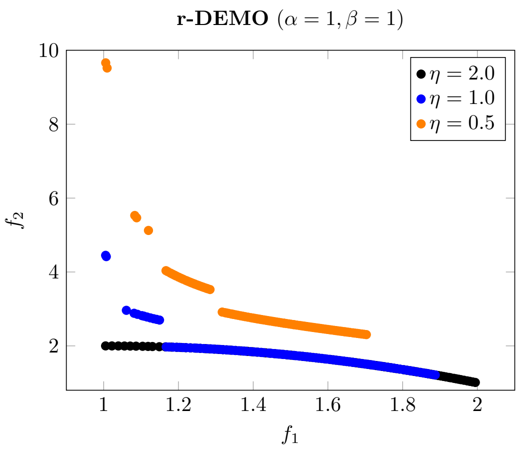
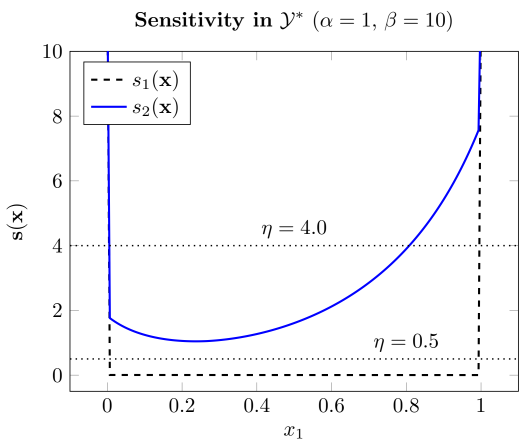


Figure 5: Test problem (11) with $\alpha = \beta = 1$. Top: sensitivities of the efficient solutions in both objectives and threshold lines. Bottom: typical results of running the r-DEMO.



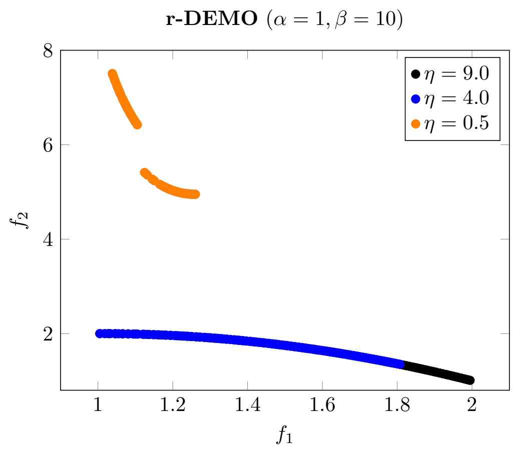


Figure 6: Similar results to Fig. 5, with $\alpha = 1,\ \beta = 10$.

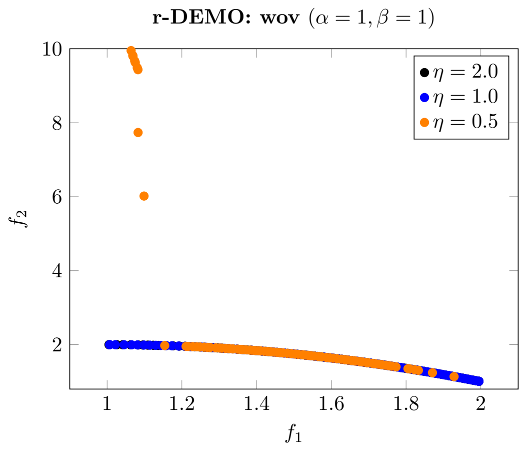


Figure 7: Typical result obtained by r-DEMO using the *wov* strategy to compute the sensitivity, for the same problem shown in Figure 5.

When $\eta = 2.0$, the whole front is robust, so the algorithm approximates its complete extension as the usual DEMO does. Lowering η to 1.0, a small portion with low values of x_1 becomes too sensitive (which translates to a non-robust portion on the "left" of the efficient front), and this is detected by the algorithm by favoring a more robust front instead. Finally, when $\eta = 0.5$, all original Pareto-solutions become unfeasible, and the r-DEMO correctly returns a robust front that is completely different from the original one.

Similar results are obtained for different values of α and β, as illustrated in Figure 6 for $\alpha = 1$ and $\beta = 10$. For $\eta = 9.0$, the algorithm returns the original front as anticipated. For $\eta = 4.0$, solutions with larger values of x_1 (located in the right portion of the front) become too sensitive, so they are not returned by the r-DEMO. Finally, setting $\eta = 0.5$ results in a robust front that is completely distinct from the non-robust one, as in the previous case.

For comparison, Figure 7 shows the results obtained using r-DEMO equipped with the *wov* approach to compute the sensitivity, in which case the algorithm fails to detect the non-robust regions. The approximations for $\eta = 2.0$ and 1.0 are almost identical, and, for $\eta = 0.5$, the algorithm still approximates a portion of the efficient front, even though its whole extension is unfeasible. Similar results are obtained for neighborhood sizes up to $N = 50n$ points, suggesting that the strategy of simply relying on the worst observed value in a neighborhood as a surrogate for solution sensitivity may be inadequate even for large sample sizes.

4 DISCUSSION AND CONCLUSIONS

The preliminary results presented in this paper indicate that the proposed approach is able to correctly converge to solutions with sensitivities within provided specifications. This feature simplifies the final decision process, as all approximated solutions will satisfy the stated requirements of the DM.

A strong advantage of this *a priori* robustness treatment when compared to the usual *post hoc* sensitivity analysis [3] is the ability to avoid non-robust regions and focus the search only on the most interesting solutions, i.e., those that are feasible from a robustness perspective. In this sense, the proposed approach is similar to the inclusion of preferences in an evolutionary algorithm [7, 10], from both effectiveness and computational burden perspectives. For instance, if half the efficient front is not robust according to the DM's specified threshold, the *post hoc* approach would waste about half of the approximated solutions (and corresponding computational costs), while the technique proposed in this work would provide an even more representative sample of the robust portion. This line of argument can be extended to the extreme situation in which the whole Pareto front is too sensitive (as illustrated in Figs. 5 and 6), in which case *a posteriori* sensitivity analyses would result in no useful solution.

Despite the promising results, the experiments and analysis presented in this paper represent only a preliminary investigation on the abilities and characteristics of the proposed approach. A more thorough investigation, including comparisons with existing approaches using a wider set of test problems, is a natural follow-up to this paper. For that, however, an interesting question is posed: how to adequately compare robust optimization algorithms that define robustness in distinct ways, without introducing biases in the comparison? This question is relevant not only for the future development and evaluation of the proposed approach, but also for the wider research field of robust optimization in general.

Another essential step in the development of this approach, which was not treated in the present work, is constraint handling. In its current version, the algorithm is only capable of effectively handling box violations, as solutions closer than $\Delta\mathbf{x}$ to the boundaries can be easily detected as non-robust. An extension to general, nonlinear constraints is fundamental to allow the broad applicability of this method, and also represents a natural continuation of this research.

Besides the two aspects pointed above, there are other minor gaps in the current that must be investigated further: the validation of the proposed method in applied expensive optimization problems, for which robustness is an essential part of the decision-making process; and the use of different methods for generating points in the neighborhood of each solution (e.g., uniform sampling, Sobol sequences, Latin Hypercube Sampling etc.), coupled with effective archival and retrieval techniques to use all points already visited by the optimization algorithm, aimed at reducing the computational costs of the sensitivity computation.

ACKNOWLEDGMENT

The authors would like to thank Prof. Elizabeth Wanner for her insightful ideas and contributions. This work has been partially funded by grants and scholarships from Brazilian agencies FAPEMIG, CNPq, and CAPES. All authors are members of the Operations Research and Complex Systems Laboratory – ORCS Lab.

REFERENCES

[1] Carlos Barrico, C.H. Antunes, and Dulce Fernão Pires. 2009. Robustness Analysis in Evolutionary Multi-Objective Optimization. *Proc. 9th European Conference on Evolutionary Computation in Combinatorial Optimization* (2009), 1–36.

[2] A. Ben-Tal, L.E. Ghaoui, and A. Nemirovski. 2009. *Robust Optimization.* Princeton University Press.

[3] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. 2007. Theory and applications of Robust Optimization. (2007).

[4] H. Beyer and B. Sendhoff. 2007. Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering* 196 (July 2007), 3190–3218.

[5] Hans Georg Beyer and Bernhard Sendhoff. 2007. Evolutionary algorithms in the presence of noise: To sample or not to sample. *Proc. 2007 IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007* May (2007), 17–24.

[6] J Branke, K Deb, K Miettinen, and R Slowinski. 2008. *Multiobjective optimization: Interactive and evolutionary approaches.*

[7] CAC Coello. 2000. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. *Evolutionary Computation, 2000. Proc. 2000* (2000), 30–37.

[8] Kalyanmoy Deb and Himanshu Gupta. 2006. Introducing robustness in multi-objective optimization. *Evolutionary Computation* 14, 4 (Dec. 2006), 463–494.

[9] A. Gaspar-Cunha and J. A. Covas. 2008. Robustness in multiobjective optimization using evolutionary algorithms. *Computational Optimization and Applications* 39, 1 (2008), 75–96.

[10] Fillipe Goulart and Felipe Campelo. 2016. Preference-guided evolutionary algorithms for many-objective optimization. *Information Sciences* 329 (2016), 236–255.

[11] Yaochu Jin and Bernhard Sendhoff. 2003. *Trade-Off between Performance and Robustness: An Evolutionary Multiobjective Approach.* 237–251 pages.

[12] Yaochu Jin Yaochu Jin and J. Branke. 2005. Evolutionary Optimization in Uncertain Environments - A Survey. *IEEE Transactions on Evolutionary Computation* 9, 3 (2005), 303–317.

[13] A. S. Lewis. 2002. *Robust Regularization.* Technical Report.

[14] Efrén Mezura-Montes and Carlos A. Coello Coello. 2008. *Constrained Optimization via Multiobjective Evolutionary Algorithms.* Springer Berlin Heidelberg, Berlin, Heidelberg, 53–75.

[15] Seyedali Mirjalili and Andrew Lewis. 2016. Obstacles and difficulties for robust benchmark problems: A novel penalty-based robust optimisation method. *Information Sciences* 328 (2016), 485 – 509.

[16] Douglas C. Montgomery. 2012. *Design and Analysis of Experiments* (8 ed.). Wiley.

[17] T Robič and B Filipič. 2005. DEMO: Differential evolution for multiobjective optimization. *Evolutionary Multi-Criterion Optimization* (2005), 520–533.

[18] Amit Saha, Tapabrata Ray, and Warren Smith. 2011. Towards practical evolutionary robust multi-objective optimization. *2011 IEEE Congress of Evolutionary Computation, CEC 2011* (2011), 2123–2130.