Sensitivity to constraints in blackbox optimization *

Charles Audet † J.E. Dennis Jr. ‡ Sébastien Le Digabel §

August 26, 2010

Abstract: The paper proposes a framework for sensitivity analyses of blackbox constrained optimization problems for which Lagrange multipliers are not available. Two strategies are developed to analyze the sensitivity of the optimal objective function value to general constraints. These are a simple method which may be performed immediately after a single optimization, and a detailed method performing biobjective optimization on the minimization of the objective versus the constraint of interest. The detailed method provides points on the Pareto front of the objective versus a chosen constraint. The proposed methods are tested on an academic test case and on an engineering problem using the mesh adaptive direct search algorithm.

Key words: Sensitivity analysis, trade-off studies, Blackbox optimization, Constrained optimization, Biobjective optimization, Mesh Adaptive Direct Search algorithms (MADS).

1 Introduction

Optimization aims at identifying an argument that minimizes a given objective function subject to satisfying a set of constraints. But when this goal is reached, the next question is to asses the problem formulation by considering the change in the optimal value of the objective function when constraints are slightly perturbed. In the smooth case, and under appropriate constraint qualifications, this sensitivity study is usually done by analyzing the Lagrange multipliers, which represent the directional derivative of the objective function in the directions of the gradient of the constraints. In a more general context, it is frequent that these multipliers do not exist or they are not unique. Still, the sensitivity question remains important. We propose here two practical ways of providing information to answer this question, and introduce a general framework designed to work with other derivative-free methods [13].

*Work of the first author was supported by NSERC grant 239436-05, AFOSR FA9550-07-1-0302, and ExxonMobil Upstream Research Company.

†GERAD and Département de mathématiques et de génie industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec H3C 3A7 Canada, www.gerad.ca/Charles.Audet, Charles.Audet@gerad.ca.

‡Computational and Applied Mathematics Department, Rice University - MS 134, 6100 South Main Street, Houston, Texas 77005-1892 USA, www.caam.rice.edu/∼dennis, dennis@rice.edu.

§GERAD and Département de mathématiques et de génie industriel, Ecole Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec H3C 3A7 Canada, www.gerad.ca/Sebastien.Le.Digabel, Sebastien.Le.Digabel@gerad.ca.

Our first approach can be applied immediately after a single optimization of the problem, and does not require any supplementary function evaluations. Our detailed approach is not so inexpensive and neither approach is as precisely related to sensitivity derivatives as the Lagrange multipliers, but we believe that our approaches provide useful information about how much could be gained in the objective function by a trade-off of slightly relaxing the constraints.

Consider the general constrained optimization problem

$$\min_{x \in \Omega} \quad f(x) \tag{1}$$

where $\Omega = \{x \in X : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$ denotes the feasible region, and $f, c_j : X \to \mathbb{R} \cup \{\infty\}$ for all $j \in J = \{1, 2, \ldots, m\}$, and X is a subset of $\mathbb{R}^n$. The constraints $c_j(x) \leq 0$ with $j \in J$ provide measures of their violations and thus are referred to as *quantifiable constraints*. The set X may also be defined with functions or relations, but these are not explicitly provided or give no measure of violation, such as binary constraints for instance. The objective function f, most of the functions $c_j(x)$, and most of the functions defining X are considered as blackbox functions according to the terminology used in [8]. Such functions are typically the result of computer simulations that sometimes fail to evaluate even for points satisfying all explicit constraints. Not all functions are necessarily provided as blackboxes and some may simply be bounds on the variables for example.

Following the terminology used in [8] and in the book [13], we partition the constraints in three categories: First, *unrelaxable constraints* either cannot be violated by any trial point in order for the simulation to execute, or else, the user has specified that they should always hold – linear inequalities might be an example of the latter case. For example, a quantifiable constraint representing a length needs to be nonnegative otherwise the code will fail. Second, *relaxable constraints* may be violated and the simulation will execute. A constraint representing a monetary budget is an example of a relaxable constraint. For these constraints, a measure of how much the constraint is violated must be provided. Finally *hidden constraints* [12] refer to constraints that are not known *a priori* and is a convenient term to exclude the points in the feasible region at which the blackbox fails to return a value. In [11] and [3] the simulations failed to execute on 60% and 43% of the calls, respectively.

With this terminology, it is natural to label the constraints defining the set X as unrelaxable. Hidden constraints are implicitly part of the definition of X. However the quantifiable constraints $c_j \leq 0$ with $j \in J$ may be relaxable or unrelaxable. In the example above, the nonnegative length is unrelaxable, but the monetary budget is relaxable. Note also that explicit bounds on the variables may be considered as quantifiable constraints and may be written as $c_j = x_i - u_i \leq 0$ or $c_j = l_i - x_i \leq 0$ for some $j \in J$ and $i \in 1, 2, \ldots, n$. Bounds may be treated as relaxable or unrelaxable constraints, as can any other quantifiable constraints. Let J_R and J_U denote a partition of J containing the indices of the relaxable and unrelaxable constraints, respectively.

There are different strategies to deal with individual blackbox constraints and the strategies need not be identical for every constraint. The most drastic method to treat a constraint is the *extreme barrier* approach (EB) which consists in rejecting any trial point that violates the constraint [6]. Constraints that are not quantifiable are required in our codes to be treated by the EB. This means that the entire set X including the hidden constraints are handled by the EB. However a user may choose to treat some relaxable constraints with the EB approach. A useful consequence is that when the functions defining the problem are evaluated sequentially, then the EB approach can interrupt this process and save the costly process of launching (or completing) the simulation to evaluate the remaining functions.

Alternate relaxable constraint-handling approaches are the *filter* [5] and the *progressive barrier* [7] methods, originally designed for the Generalized Pattern Search (GPS) [11] and the Mesh Adaptive Direct Search algorithm (MADS) [6] for blackbox optimization, respectively. These methods may be used with different derivative-free algorithms to allow exploration of the infeasible region by exploiting the measures of the constraints violations. The constraints defining X as well as quantifiable unrelaxable constraints (i.e. those indexed by J_U) are treated by the EB within these strategies. Thus only relaxable constraints $c_j \leq 0$ with $j \in J_R$ are explicitly brought to feasibility by the filter or the progressive barrier.

In summary, the extreme barrier is used for the constraints defining X and the quantifiable unrelaxable constraints $c_j \leq 0$ with $j \in J_U$, and the filter or the progressive barrier (not both) are used to handle the relaxable constraints $c_j \leq 0$ with $j \in J_R$.

The present paper proposes two methods, named the *simple* and the *detailed* methods, to study the trade-offs of the objective versus the quantifiable constraints including bounds on the variables. The outputs of both our methods are related to so-called "trade studies" and they are represented by plots of the objective function f versus each quantifiable constraints $c_j(x) \leq 0$ over the domain

$$x \in \Omega_j = \{x \in X : c_i(x) \leq 0, i \in J \setminus \{j\}\} .$$

The set Ω_j is a relaxation of the true domain Ω, obtained by eliminating the quantifiable constraint $c_j(x) \leq 0$. The simple method produces a coarse approximation of the sensitivity at no cost simply by inspecting the *cache* (or the *history*) generated by the derivative-free algorithm. The cache contains all trial points and function values evaluated during one or several executions of the algorithm. The detailed method requires more function evaluations but provides a better approximation of the sensitivity. It consists in solving biobjective optimization problems of f versus c_j with $x \in \Omega_j$ for selected $j \in J$, while requiring feasibility with respect to the other constraints. Thus the detailed method requires the use of a derivative-free method for biobjective optimization problems such as the ones proposed in [9] or [14]. See [17] for interesting contextual examples in aerospace design.

The paper is divided as follows: Section 2 presents the simple and detailed methods to analyze the sensitivity to constraints. Section 3 illustrates them with the MADS algorithm on an academic smooth test problem to confirm that the proposed sensitivity analysis indeed produces approximations of the Lagrange multipliers. Numerical results are also given for a previously studied styrene process production problem [4] containing relaxable, unrelaxable, and hidden constraints.

2 Trade-offs between the objective and constraints

Both the filter and progressive barrier approaches use a constraint violation function [16]

$$h(x) \;:=\; \begin{cases} \displaystyle\sum_{j \in J_R} (\max(c_j(x), 0))^2 & \text{if } x \in X\,, \\ \infty & \text{otherwise.} \end{cases}$$

to handle the relaxable constraints. This function equals 0 if and only if all relaxable constraints are satisfied. Otherwise h aggregates the violation of all relaxable constraints and is strictly positive. The filter and the progressive barrier accept or reject trial points according to compromises between the objective function and the constraints violation values. In addition, a trial point is not considered when its constraint violation value exceeds a given threshold h_{max}, or any of the unrelaxable constraints is violated. In situations where the constraints are evaluated sequentially, the process may be interrupted as soon as the incomplete sum of squared violations exceeds h_{max}. This procedure is similar to the one where trial points violate the unrelaxable constraints and avoids unnecessary computational work.

Papers [15, 16, 5, 7, 8] using the filter or the progressive barrier approaches provide some sort of sensitivity analysis by plotting the objective function value f versus the constraint violation $h(x)$. These plots can give some valuable information, as they can be use to perform local explorations around slightly infeasible solutions having a low objective function value. However these plots rely on an aggregation of the constraints and make it impossible to study the sensitivity with respect to an individual constraint. The approaches proposed in the next two subsections treat separately the trade-off between each quantifiable constraint $c_j, j \in J_U \cup J_R$ versus the objective f whether or not the constraint is treated with the extreme or progressive barriers, or the filter.

2.1 A post-optimization sensitivity analysis: the simple method

In practice, applying derivative-free software to an optimization problem creates a cache containing the coordinates of a finite number of trial points and their corresponding objective and constraint values. An existing cache may be provided as input to a new execution of the software, resulting in a larger cache.

Let $V \subset \mathbb{R}^n$ denote the set of all trial points generated during one or several executions of the algorithm. The cache contains every $x \in V$ as well as values for $f(x)$, $c_j(x), j \in J$, and a binary flag indicating whether or not x is in X. Another flag indicates if f or some values of c_j are missing.

The following steps are defined to construct a rough approximation of the trade-offs between the quantifiable constraints and the objective function. The form of these approximations consists of a series of m two-dimensional plots $\mathcal{P}_j$ for $j \in J$. Let $V' \subseteq V$ be the set of trial points satisfying the unrelaxable constraints $x \in X$ and for which f and all values of c_j with $j \in J$ are defined. The set V' is constructed by a simple inspection of the cache. The simple method enumerates every trial point $x \in V'$ and for every constraint $j \in J$, it adds the point $(c_j(x), f(x))$ to the plot $\mathcal{P}_j$.

The plot $\mathcal{P}_j$ will possibly contain positive and negative values of $c_j(x)$ but only nonpositive values for unrelaxable constraints. This provides information of the sensitivity of f when perturbing the constraint.

The analysis is also valid for integer or boolean constraints. A boolean constraint for example will produce a plot indicating the objective function values for both binary values.

The quality of the sensitivity analysis obtained by the inspection of the cache depends heavily on the effort deployed by the optimization process. There are cases, as we will see in the numerical results below, in which the simple analysis does not give significant insight. But this is mitigated since the simple analysis is obtained for free after a single optimization, i.e. without any additional function calls. The next section proposes a way to enhance the analysis.

2.2 Sensitivity analysis using biobjective optimization: the detailed method

In this section we propose a way to generate a more complete sensitivity plot $\mathcal{T}_j$ of the objective function f versus the quantifiable constraint $c_j(x) \le 0$. We suggest to start by solving problem (1) and then to follow with a simple analysis, as defined in Section 2.1. Then, perform detailed analyses in sequence for each active or nearly active constraints of interest. The cache is not purged between the different detailed analyses, and it grows at each analysis. The set V' is defined as in the previous section to be the set of previously generated trial points satisfying the unrelaxable constraints for which all function values are defined in the cache.

The detailed analysis for the quantifiable constraint $c_j \le 0$ is done as follows. Let $\underline{c}_j \le 0$ and $\bar{c}_j \ge 0$ be two user-provided bounds to restrain the analysis to interesting ranges of constraint violations. The constraint $c_j \le 0$ is replaced with the two related constraints $\underline{c}_j \le c_j(x) \le \bar{c}_j$. The status of the two new constraints is unrelaxable even if the original constraint they replace was relaxable. If one only wishes to analyze a relaxation of the constraint, then one would set $\underline{c}_j = 0$. The value $\underline{c}_j = -\infty$ is also possible if one wishes to study extreme tightening of the constraint. Similarly if one only wishes to analyze a tightening of the constraint, then one would set $\bar{c}_j = 0$. The value $\bar{c}_j = \infty$ is also possible if one wishes to study the effect of removing the constraint.

The trade-off plot $\mathcal{T}_j$ is obtained by applying the simple method proposed in the previous version by analyzing the enriched cache obtained after the application of the derivative-free solver to the following biobjective problem

$$\begin{array}{rl} \min\limits_{x \in \Omega_j} & (c_j(x),\ f(x)) \\ \text{s.t.} & \underline{c}_j \le c_j(x) \le \bar{c}_j \end{array} \tag{2}$$

using a biobjective derivative-free algorithm.

Unlike the free simple analysis, each detailed analysis requires additional function evaluations. It is the price to pay for a more complete trade-off study. Besides, a designer may consider at least some of these trade-off results to be an essential part of the design process. However this cost may be entirely controlled by the user of the method by selecting which quantifiable constraints to analyze, and by setting appropriate stopping criteria or precision in the biobjective algorithm. In the numerical results of this paper, an upper bound on the number of evaluations is used as the stopping criterion.

2.3 Sensitivity and Lagrange multipliers

The present work is designed for blackbox optimization. Nonetheless, to understand the behavior of the proposed framework, we interpret the signification of the results on smoother problems.

Let x^* be a local optimal solution of problem (1) and suppose that for some quantifiable constraint $j \in J$ a unique Lagrange multiplier exists, is strictly positive, and is denoted by λ_j. The KKT conditions ensure that the constraint is active: $c_j(x^*) = 0$. It is well-known that λ_j represents the sensitivity of the optimal objective function value with respect to the right-hand side value of the corresponding constraint [19]. Under appropriate constraint qualifications, this signifies that if the constraint $c_j(x) \leq 0$ is replaced by $c_j(x) \leq \epsilon$, then for small values of ϵ the optimal objective function value varies by approximately $-\epsilon\lambda_j$. Another way of stating this result is that if $x^*(\epsilon)$ is the optimal solution to the perturbed problem with the same active constraints as x^*, then

$$\left.\frac{df(x^*(\epsilon))}{d\epsilon}\right|_{\epsilon=0} = -\lambda_j. \tag{3}$$

Define $\Omega'_j := \{x \in \Omega_J : \underline{c}_j \leq c_j(x) \leq \bar{c}_j\}$ to be the domain of the biobjective problem (2) considered by the detailed analysis, and let $\mathcal{T}_j^*$ be the set of all optimal trade-off solutions of the same problem:

$$\mathcal{T}_j^* = \{(c_j(x),\ f(x)) \ : \ x \in \Omega'_J, y \not\prec x \ \forall y \in \Omega'_J\}$$

where $y \prec x$ stands for "y dominates x" and means that $c_j(y) \leq c_j(x)$ and $f(y) \leq f(x)$ with at least one of the two inequalities being strict. Therefore a solution $x \in \Omega'_j$ belongs to $\mathcal{T}_j^*$ if and only if it is not dominated by any solution of Ω'_J. The undominated points $\mathcal{T}_j$ generated by the detailed analysis is an approximation of the trade-off set $\mathcal{T}_j^*$.

Theorem 2.1 *Let x^* be a local optimal solution of problem (1) with a unique Lagrange multiplier λ, and let $j \in J$ be index of an active constraint with $\lambda_j > 0$. The slope of the tangent to the set $\mathcal{T}_j^*$ at $c_j(x) = 0$ is equal to $-\lambda_j$.*

Proof. Since j is the index of an active constraint, the trade-off set $\mathcal{T}_j^*$ contains the optimal solutions of problem (1) in which the constraint $c_j(x) \leq 0$ is replaced by $c_j(x) \leq \epsilon$ for all values of ϵ ranging from $\underline{c}_j$ to $\bar{c}_j$. The plot of the set $\mathcal{T}_j^*$ contains the trade-offs between the objective function value $f(x)$ versus the quantifiable constraint $c_j(x)$. When it is nonzero, the slope of the tangent to that set at $c_j(x) = 0$ corresponds to the variation rate of the objective function value with respect to the right-hand-side of the constraint $c_j(x) \leq \epsilon$. Equation (3) ensures that the slope is equal to the negative of the Lagrange multiplier λ_j. □

Observe that the set $\mathcal{T}_j^*$ contains more information than what can be deduced from a Lagrange multiplier. The set provides the exact optimal values of the optimization problem when the upper bound on the constraint varies from $\underline{c}_j$ to $\bar{c}_j$, as the multiplier only gives the slope at $c_j(x) = 0$. The numerical results below illustrate this observation on two problems.

3 Numerical results

This section shows the application of the framework described in the previous sections. A first subsection gives the technical details concerning the implementation with the NOMAD software. Two test-problems are then considered: First an academic smooth and convex problem for which the Lagrange multipliers are known, and second a real blackbox application from chemical engineering.

3.1 Implementation details

All numerical experiments are performed using version 3.4 of the NOMAD software [1, 18] designed for constrained blackbox optimization. It implements the Mesh Adaptive Direct Search algorithm (MADS) [6] and is distributed under the LGPL licence.

Two new tools were added to this version of the NOMAD package to conduct sensitivity analyses. The first one, called `cache_inspect`, corresponds to the simple analysis presented in Section 2.1. It consists in looping through the cache generated by a NOMAD optimization run and either displays V' or undominated points relative to the objective and to the studied constraint or bound. The second tool is called `detailed_analysis` and corresponds to the detailed analysis described in Section 2.2 that transforms a given constraint or variable bound into a second objective.

The BIMADS algorithm described in [9] is used for biobjective optimization. It is included in the NOMAD package. The BIMADS method consists in launching several single-objective reformulations of the biobjective problem in order to approximate the Pareto front. The two first single-objective optimizations correspond to the two individual objectives while the other runs combine both objective. Because of well-known drawbacks to weighting the objectives, BIMADS uses nonlinear reformulations which are solved by the single-objective MADS algorithm. Although the detailed analysis may be launched directly, it is typically performed after a simple analysis, and each run adds points to the existing cache.

It is possible to initiate NOMAD with several starting points but we observed that considering all the cache points as starting solutions leads to bad behavior and unbalanced results. Instead the cache is inspected prior to the biobjective optimization and only one feasible starting point with the best value of the BIMADS single-objective formulation is used as starting point.

This cache inspection allows us to deactivate the Latin Hypercube (LH) search [20] performed by default at the first BIMADS iteration thus saving some computational resources. The original purpose of this search is to provide a good initial guess for the first single-objective optimization. Another advantage of disabling the LH search is that it allows to perform deterministic experiments as long as the ORTHOMADS [2] types of directions are used, as they are by default.

The snap_to_bounds strategy is also enabled.This consists in projecting back to the bounds points that are generated outside the boundaries.

When the constraint corresponds to a bound on a variable, then $c_j(x) = x_i - u_i \leq 0$ or $c_j(x) = l_i - x_i \leq 0$ is replaced with $\underline{c}_j \leq c_j(x) \leq \bar{c}_j$. New bounds are defined in order to refine the domain and a simple rescaling of the variable is applied in order to conserve the original magnitude.

A maximal number of evaluations bb_eval may be given as well as an initial maximal number of evaluations init_bb_eval which limit the number of evaluations performed by the two first single-objective optimizations. This is motivated by the fact that a more consistent effort on these two first optimizations is necessary in order to obtain a well distributed approximation of the Pareto front. For the academic problem, the values bb_eval$= 1000$ and init_bb_eval$= 100$ are considered.

The post-optimization tools for sensitivity analysis (cache inspection and automatic biobjective formulation) as well as the two test-problems described in the next two sections are included in the NOMAD package and are described in the user-guide.

3.2 An academic test problem

Consider the three-variable optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^3_+} \quad & (x_1 - 5)^2 + (x_2 - 6)^2 + (x_3 - 2)^2 \\ \text{s.t.} \quad & x_1 \leq 2, \\ & x_2 \leq 2, \\ & x_3 \leq 2, \\ & x_1^2 + x_2^2 + x_3^2 - 9 \leq 0. \end{aligned}$$

The unique optimal solution is $x^* = (2, 2, 1)$ with optimal value $f^* = 26$ and the Lagrange multipliers are $\lambda = (2, 4, 0, 1)$. The NOMAD software was applied to this problem with the default parameters and it terminated after 592 function evaluations as the mesh size parameter dropped below the numerical precision of the machine. The bounds on the variables were treated by the extreme barrier approach and the spherical constraint was treated by the progressive barrier.

The simple analysis is illustrated in Figure 1 which zooms in on the objective function values for every point in Ω_j, $j = 1, 2, 3, 4$, and the wider plot on the bottom shows the trade-off solutions for the three active constraints at the optimal solution. Using the notation introduced in Section 2, the first four plots display the sets $\mathcal{P}_j$ and the last one displays $\mathcal{T}_j^*$.

The fact that the upper bounds on the variables are treated by the extreme barrier is apparent on the figure as no points have a constraint value exceeding zero. The constraint $x_2 \leq 3$ is inactive at the optimal solution and becomes active near $x_3 - 2 \leq -1$, i.e. when $x_3 \leq 1$. This is why it does not appear on the plot for the active constraints.

The light colored continuous curve on each four plots associated to one constraint represents the analytical minimal objective function value with respect to the constraint value, and coincides

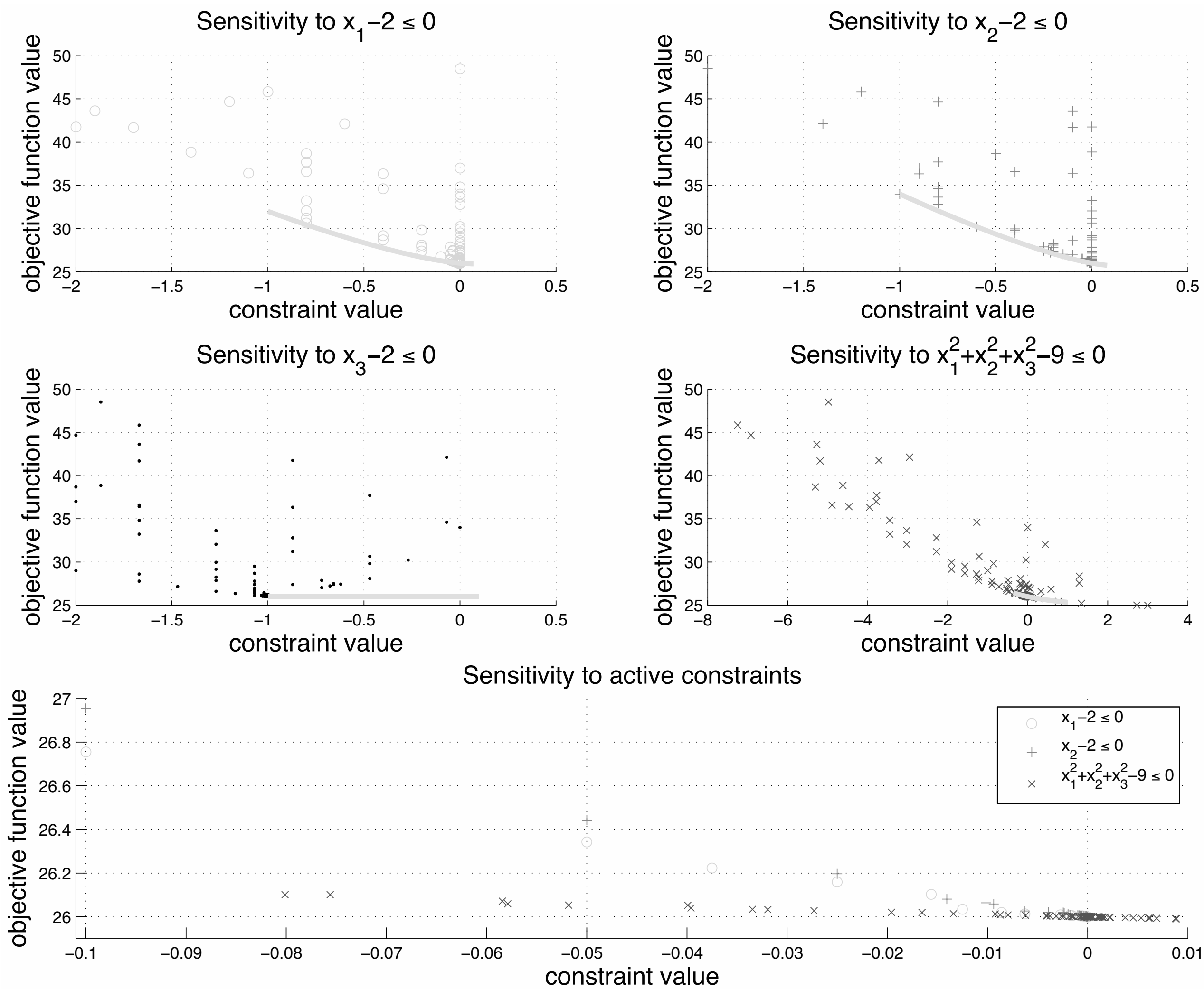

Figure 1: Simple sensitivity analysis for the academic problem.

by construction with the set $\mathcal{T}_j^*$. The curve is only plotted for the constraint values for which the optimal solution have the same three active constraints. The slope at $c_j = 0$ of this curve is equal to the corresponding Lagrange multiplier. We observe that the points generated by the simple approach are close to this theoretical curves.

The bottom figure suggests that the upper bound on x_1 and on x_2 have a more significant effect on the objective function value than the spherical constraint. Indeed, the corresponding multipliers are 2 and 4 for the bounds and 1 for the nonlinear constraint. The relative effect of the two bound constraints is not clearly apparent on the figure. To make it more apparent, three detailed analyses were performed, each with a budget of $1,000$ function evaluations. The first one studied f versus the upper bound on x_1 and is illustrated in Figure 2 for $1.9 \le x_1 \le 2.1$. The undominated points of the set $\mathcal{T}_j$ generated by this particular analysis are very close to the light colored theoretical curve.

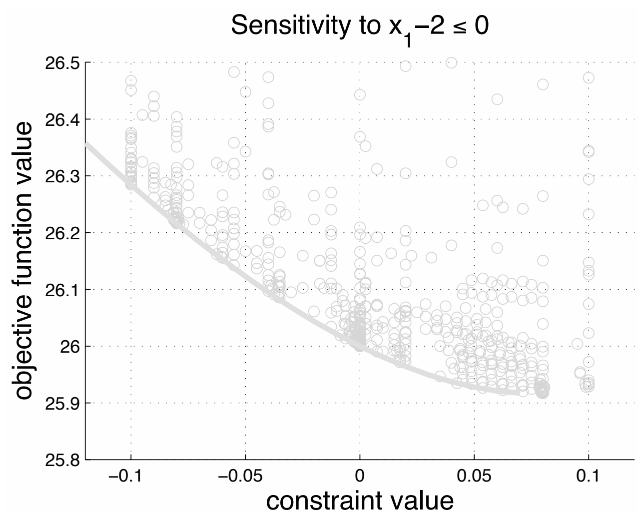

Figure 2: Detailed sensitivity analysis for the academic problem: f versus $x_1 - 2 \leq 0$.

Detailed sensitivity analyses were then performed on the active constraints $x_2 - 2 \leq 0$ and $x_1^2 + x_2^2 + x_3^2 - 9 \leq 0$ and the resulting sets $\mathcal{T}_j$ are displayed in Figure 3. The trade-offs between the objective and the constraint are much more visible than in the plots of Figure 1.

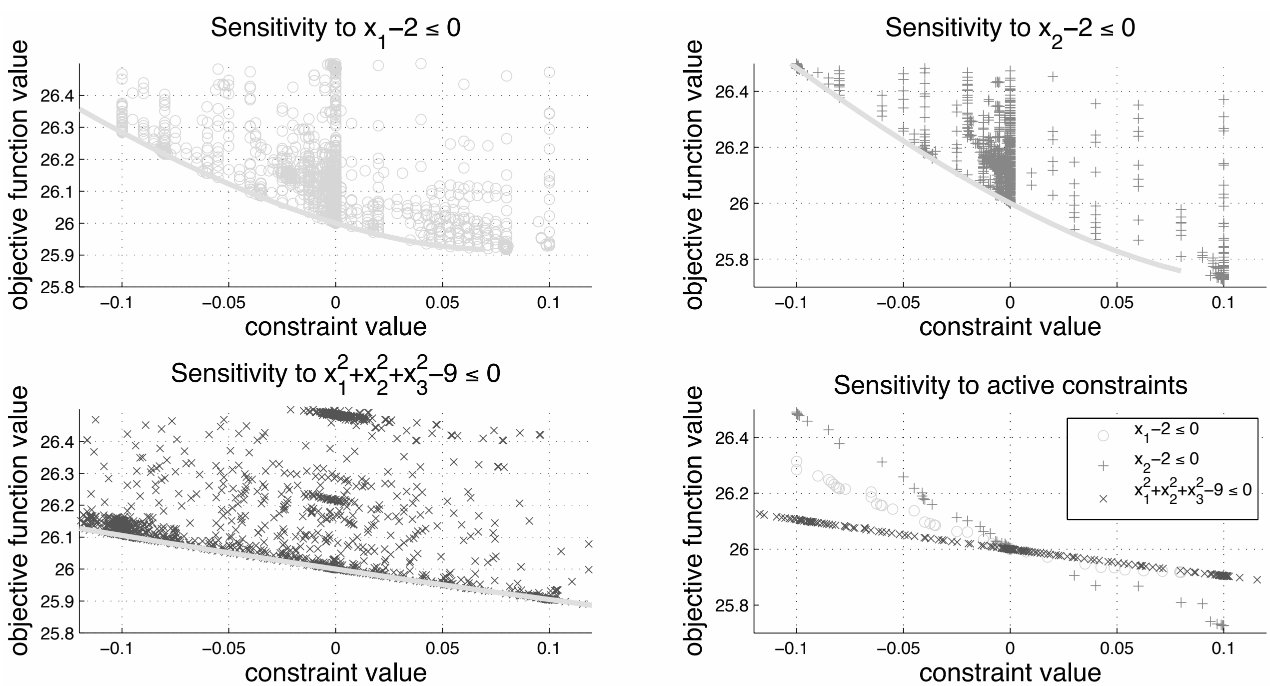

Figure 3: Detailed sensitivity analyses for the academic problem: the active constraints.

Notice that the first plot of Figure 3 differs significantly from that of Figure 2. Additional points satisfying $x_1 \leq 2$ were generated during the detailed analyses of the two other constraints. The undominated points in each plots are very close to the theoretical ones obtained by studying the analytical expression of the problem. The fourth subplot clearly reveals the relative importance of the constraints. Undoubtedly the slopes at $c_j = 0$ of the trade-off curve approximations are close to -2, -4, and -1, which are the negative of the Lagrange multipliers as predicted by Theorem 2.1.

3.3 A styrene production process test problem

The styrene production problem is described in [4]. It has 8 variables and 11 relaxable constraints, 4 of which are 0-1 constraints. Hidden constraints are also encountered when the internal numerical methods fail to converge. This happens for roughly 14% of the evaluations, as reported in [8]. The styrene problem is freely available for download on the NOMAD website [1]. This version of the problem includes scaling as all variables are between 0 and 100.

Figure 4 illustrates the sensitivity of the objective function value versus six active (or nearly active) constraints obtained with the simple analysis. Again all default NOMAD parameters were used and it terminated after 1840 function evaluations, having reached the machine precision. The extreme barrier was used for the bounds on the variables and for the four binary constraints, and the progressive barrier on the remaining seven blackbox constraints.

The constraint $c_2 = 0$ is boolean and the figure shows that there is a loss resulting from its application. The constraints that are not represented in the figure are either boolean or clearly inactive. The constraints $c_5 \leq 0$ and $c_6 \leq 0$ are inactive by a small amount of the order of 10^{-3} and they could be made more restrictive without affecting the optimal value. The three other constraints in the figure are active at the solution. The plot to the bottom shows their trade-off values. The simple analysis does not clearly reveal the relative importance of these constraints.

We performed three detailed analyses, each with a budget of $10,000$ function evaluations. The results are illustrated in Figure 5. This overall budget of $30,000$ evaluations was chosen to be identical to that used in [10] where a tri-objective version of the same optimization problem is analyzed.

The usefulness of such a sensitivity analysis could go as follows. Figure 5 reveals that the optimal value is not very sensitive to the upper bound on x_3 but increasing the upper bound on x_1 or decreasing the lower bound on x_5 would lead to an important improvement of the optimal objective function value.

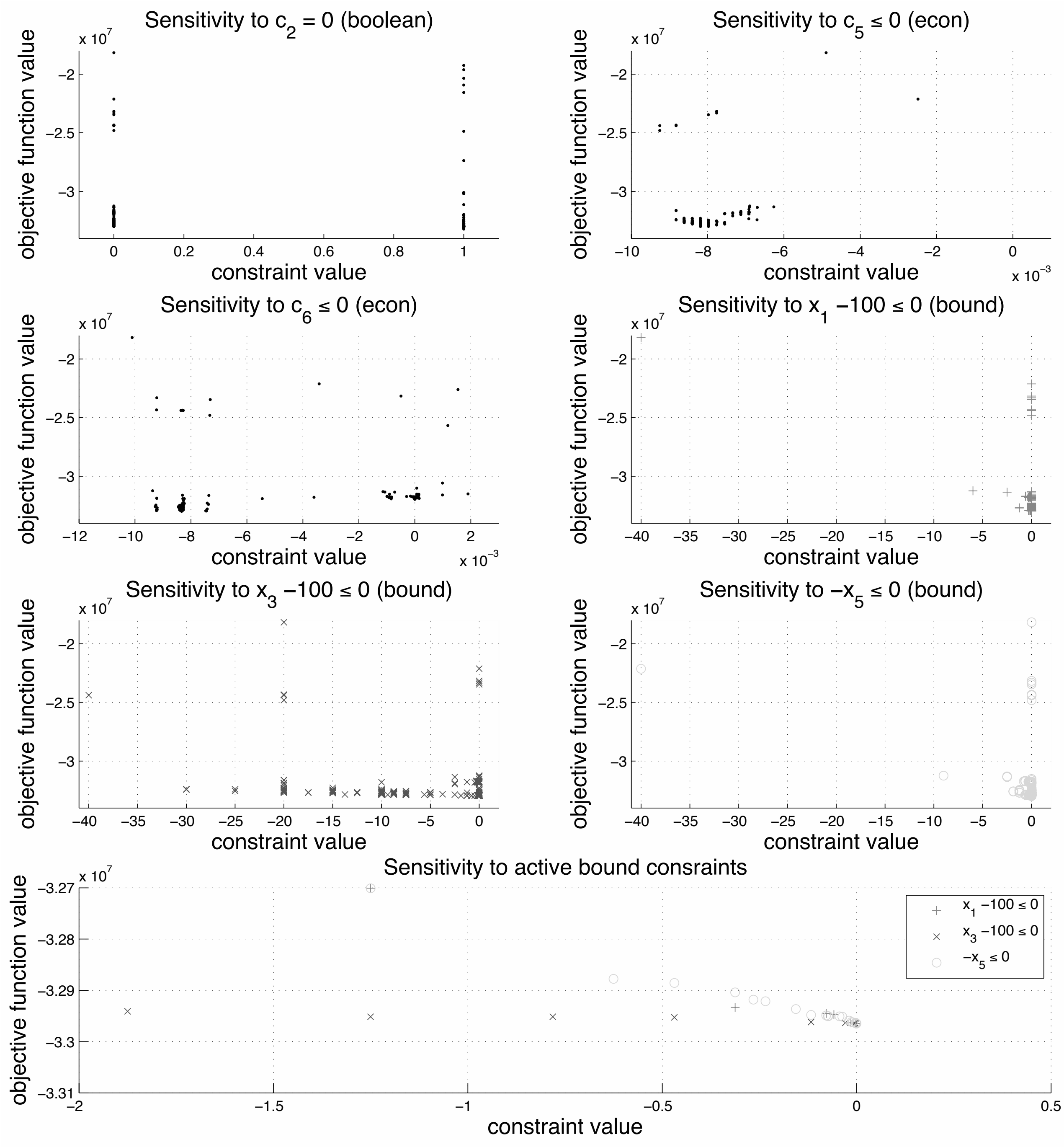

Figure 4: Simple sensitivity analysis for the styrene problem.

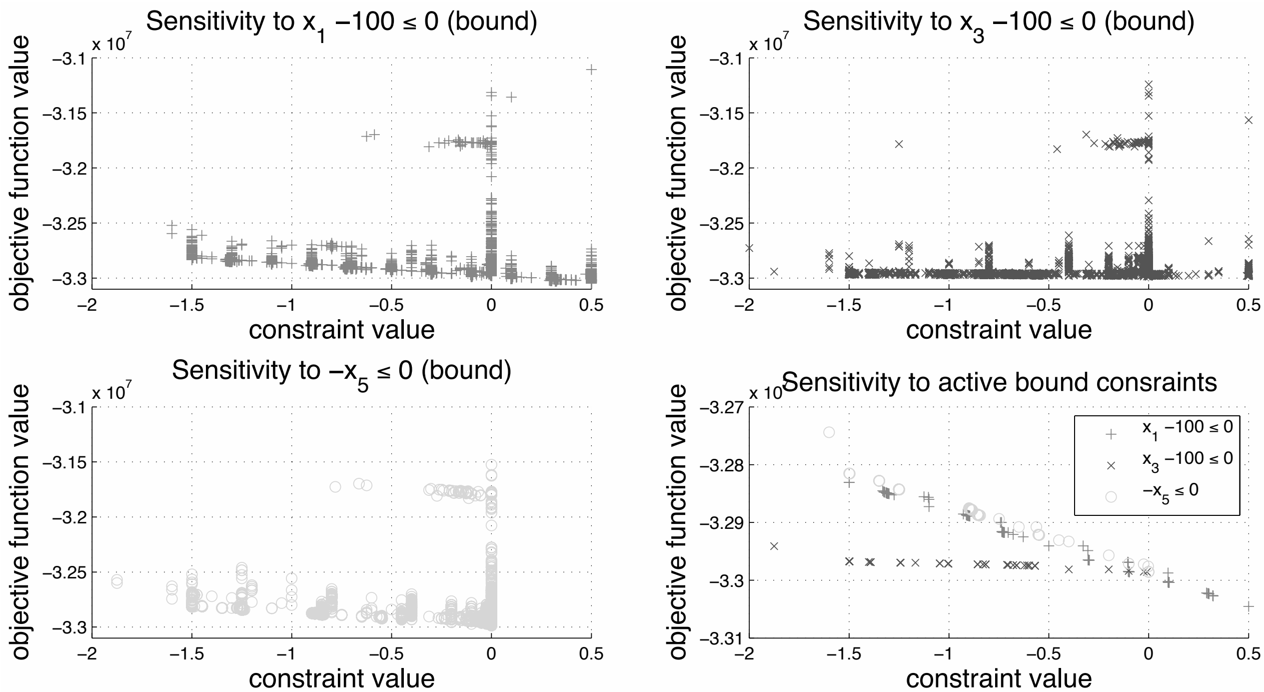

Figure 5: Detailed sensitivity analyses for the styrene problem: the active constraints.

Discussion

The paper presents a general framework for sensitivity analyses of quantifiable constraints for blackbox optimization problems. Results produced by the application of the framework can be visualized by plots of the objective function value versus small changes in the right-hand-sides of the constraints. No constraint qualifications are necessary to perform this analysis. However, when appropriate constraint qualifications are satisfied, the slope of these plots are equal to the negative of the Lagrange multipliers.

Numerical tests were performed with the NOMAD software, an implementation of the MADS and BIMADS algorithms. The current version of NOMAD now contains both the simple and detailed algorithmic tools to analyze the sensitivity to constraints. The simple strategy can be launched immediately after any run on the original optimization problem. It inspects the cache and does not require any supplementary function evaluations.

The analysis can be made more precise by performing the detailed analysis, which solves the biobjective optimization problem consisting of the minimization of the objective function versus the constraint.

The framework is illustrated on an academic problem, only to verify that the produced results coincide with the Lagrange multiplier theory. We also illustrate it on an engineering problem and showed how the results can be interpreted in practice. We chose to use our NOMAD implementation but the method would also be appropriate for any other solver that records a cache of the evaluated trial points. The biobjective direct search solver used in the detailed analysis could also be replaced by the recent method of [14].

References

[1] M. A. Abramson, C. Audet, G. Couture, J. E. Dennis, Jr., and S. Le Digabel. The NOMAD project. Software available at http://www.gerad.ca/nomad.

[2] M. A. Abramson, C. Audet, J. E. Dennis, Jr., and S. Le Digabel. OrthoMADS: A deterministic MADS instance with orthogonal directions. *SIAM Journal on Optimization*, 20(2):948–966, 2009.

[3] C. Audet, V. Béchard, and J. Chaouki. Spent potliner treatment process optimization using a MADS algorithm. *Optimization and Engineering*, 9(2):143–160, 2007.

[4] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. *Journal of Global Optimization*, 41(2):299–318, June 2008.

[5] C. Audet and J. E. Dennis, Jr. A pattern Search Filter Method for Nonlinear Programming without Derivatives. *SIAM Journal on Optimization*, 14(4):980–1010, 2004.

[6] C. Audet and J. E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[7] C. Audet and J. E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. *SIAM Journal on Optimization*, 20(4):445–472, 2009.

[8] C. Audet, J. E. Dennis, Jr., and S. Le Digabel. Globalization strategies for mesh adaptive direct search. *Computational Optimization and Applications*, 46(2):193–215, June 2010.

[9] C. Audet, G. Savard, and W. Zghal. Multiobjective optimization through a series of single-objective formulations. *SIAM Journal on Optimization*, 19(1):188–210, 2008.

[10] C. Audet, G. Savard, and W. Zghal. A mesh adaptive direct search algorithm for multiobjective optimization. *European Journal of Operational Research*, 204(3):545–556, 2010.

[11] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, February 1999.

[12] T. D. Choi and C. T. Kelley. Superlinear convergence and implicit filtering. *SIAM Journal on Optimization*, 10(4):1149–1162, 2000.

[13] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS/SIAM Book Series on Optimization. SIAM, Philadelphia, 2009.

[14] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. Technical Report Preprint 10-18, Dept. of Mathematics, Univ. Coimbra, May 2010. Paper available at http://www.mat.uc.pt/˜lnv/papers/dms.pdf.

[15] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter. On the global convergence of trust-region SQP-filter algorithms for general nonlinear programming. *SIAM Journal on Optimization*, 13(3):635–659, 2002.

[16] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, Series A, 91:239–269, 2002.

[17] A. J. Keane and P. Nair. *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. John Wiley & Sons, New York, 2006.

[18] S. Le Digabel. NOMAD: Nonlinear optimization with the MADS algorithm. Technical Report G-2009-39, Les cahiers du GERAD, 2009. To appear in ACM Transactions on Mathematical Software.

[19] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, Berlin, second edition, 2006.

[20] B. Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, 1993.