Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 16, No. 4, pp. 595–610, 2020

CONSTRAINT SCALING IN THE MESH ADAPTIVE DIRECT SEARCH ALGORITHM*

Charles Audet, Gilles Caporossi and Stéphane Jacquet

Abstract: In an optimization problem, multiplying an inequality constraint by a positive scalar has no effect on the domain. However, such a transformation might have an effect in practice. A common strategy in constrained optimization is to aggregate the sum of all constraint violation in a single real-valued function. Multiplying a constraint by a scalar impacts that function. The present work proposes a dynamic methodology to select weights for each constraint in the Mesh Adaptive Direct Search (MADS) algorithm with the progressive barrier.

Key words: *blackbox optimization, derivative-free optimization, surrogate functions, progressive barrier*

Mathematics Subject Classification: *90C56*

---

1 Introduction

The formulation of an optimization problem impacts the solution process. A constraint requiring that a road length be less than 1km is equivalent to being less than 1000m. The present work studies the impact of scaling on the Mesh Adaptive Direct Search (MADS) algorithm [6].

The MADS algorithm is designed for blackbox optimization. In such a field, characteristics of the problem such as derivatives and Lipschitz constants are unavailable. MADS handles inequality constraints through the progressive barrier [5]. However the progressive barrier depends on the scaling of the initial problem. This means that an algorithm may generate different solutions for problems that differ only by the constraint scalings.

Let $n, m \in \mathbb{N}$, $X \subset \mathbb{R}^n$. Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ and for all $i \in [\![1, m]\!]$, $c_i : \mathbb{R}^n \mapsto \mathbb{R}$. Consider the blackbox optimization problem:

$$\begin{cases} \min\limits_{x \in X} & f(x) \\ \text{subject to} & c_i(x) \leq 0, \ \forall i \in [\![1, m]\!]. \end{cases}$$

Denote $\Omega = \{x \in X : \forall i \in [\![1, m]\!], \ c_i(x) \leq 0\}$ the feasible domain. Using the terminology from [17], the constraints $c : X \to \mathbb{R}^m$ are quantifiable and relaxable, and the set X contains the unrelaxable ones. Typically $X = \mathbb{R}^n$ or X is an hyperrectangle in $\mathbb{R}^n$.

It is easy to imagine a constraint that takes high values, like a production cost in dollars, and others that take low values, such as execution time of a simple task in minutes. Even if

*Work of the first author was supported by NSERC Discovery grant #2015-05311.

the user is aware of these magnitudes and rectifies the constraints, the scaling differences may create an imbalance between the constraints. This can give more importance to constraints compared to others. The problem

$$\begin{cases} \min\limits_{x \in X} & f(x) \\ \text{subject to} & \frac{c_i(x)}{a_i} \leq 0, \; \forall i \in [\![1, m]\!] \end{cases}$$

where $0 < a \in \mathbb{R}^m$ is equivalent to the previous problem.

The MADS algorithm was expanded [8] to take into account the scaling of the input variables $x \in \mathbb{R}^n$. However, no scaling was performed for the constraints. This is the main focus of the present work.

MADS is a direct search algorithm based on a discretization of the space of variables called " mesh ". MADS performs two different types of steps at a given iteration. The first is called the search. It can be any user-defined strategy: a quadratic model [13], latin hypercube sampling [11], or a Nelder-Mead search [9] for example. The second one is the poll step, a local exploration on the mesh. The convergence analysis of MADS relies on that step. Unlike pattern search algorithms, there is not only one parameter that describes the mesh, but two: the mesh size parameter (δ^k) that defines the resolution of the mesh and the frame size parameter (Δ^k) that defines the resolution of the frame where the points can be evaluated in the poll step.

At iteration $k \in \mathbb{R}^n$ of a poll step, the set of points that can be evaluated at this iteration is:

$$P_k = \{x^k + \delta^k d : d \in D_k\}$$

where D_k is a positive spanning set in $\mathbb{R}^n$.

During a search or a poll step, a finite list of elements $\mathcal{L}^k \subset \mathbb{R}^n$ is given for evaluation of the blackbox. But in order to accelerate the convergence, the list $\mathcal{L}^k$ is not fully evaluated. As soon as a new incumbent solution is found, then the other elements of $\mathcal{L}^k$ are not evaluated. This is called the " opportunistic strategy ". It is preferable to identify this solution as early as possible in order to save many function evaluations. To achieve this, the elements of $\mathcal{L}^k$ are sorted from most to least promising using an ordering strategy. The importance of ordering the elements of $\mathcal{L}^k$ in the opportunistic strategy is quantified in [22].

The present work is structured as follows. Section 2 describes the progressive barrier to handle the constraints. Section 3, which is the main focus of this work, defines three weightings and how they will impact the progressive barrier. Section 4 shows the numerical results. The different weightings apply on both analytical problems and blackboxes. Section 5 concludes and discusses the theoretical aspect, the numerical results and describes future work.

2 Progressive Barrier in MADS

In [6], MADS handles the constraints using the extreme barrier by simply rejecting infeasible points by optimizing the function:

$$f_\Omega(x) = \begin{cases} f(x) & \text{if } x \in \Omega \\ +\infty & \text{if } x \notin \Omega. \end{cases}$$

In [5], MADS-PB offers a different way to handle the constraints using the constraint violation function, which is an adaptation from [14]. For this function called h, if $x \notin X$ then one of the unrelaxable constraints is not satisfied and thus the other values cannot be trusted. So the constraint violation function takes the value $+\infty$.

Definition 2.1. The constraint violation function h is defined as:

$$h(x) = \begin{cases} \sum_{i=1}^{m} \max(0, c_i(x))^2 & \text{if } x \in X \\ +\infty & \text{otherwise.} \end{cases}$$

The function h aggregates all the constraint violations. It also satisfies the following property: for every $x \in \mathbb{R}^n$: $h(x) = 0$ if and only if x belongs to $\Omega = \{x \in X : c_i(x) \leq 0, \forall i \in [\![1, m]\!]\}$.

The progressive barrier does not reject all infeasible points. Let $V^k \subseteq X$ be the set of points previously evaluated by the beginning of iteration k that satisfy all non-relaxable constraints. Considered the bi-objective optimization problem where the functions are the objective f and the constraint violation h. In the progressive barrier, the only feasible points that are kept are the ones with the lowest value of f. Let

$$F_k = \underset{x \in V^k}{\operatorname{Argmin}}\{f(x) \ : \ h(x) = 0\}$$

be the set of the best feasible points. The best value of f among the feasible points is

$$f_k^F = \begin{cases} +\infty & \text{if } F_k = \emptyset \\ f(x) & \text{for any } x \in F_k \text{ otherwise.} \end{cases}$$

A partial order relation is created amongst the infeasible points, and the non-dominated points are kept according to the following order relation.

Definition 2.2. Let $x, y \in V^k$ be two evaluated points. It is said that x dominates y, denoted $x \prec y$, if $h(x) < h(y)$ and $f(x) \leq f(y)$, or if $h(x) \leq h(y)$ and $f(x) < f(y)$.

Amongst all infeasible points of V^k, only the non-dominated ones are kept. Let us define $U_k = \{x \in V^k - \Omega : \nexists y \in V^k - \Omega, y \prec x\}$ the set of non-dominated infeasible points.

A positive scalar called h^k_{max} is also defined at each iteration k. Rather than rejecting all the infeasible points, as with the extreme barrier, the progressive barrier rejects those whose constraint violation function value exceeds the threshold h^k_{max}. At iteration k, every point x from the cache that verifies $h(x) > h^k_{max}$ is rejected. The key to this method is that as k increases, the threshold $h^k_{\max}$ progressively decreases. The progressive barrier keeps all the infeasible elements from the set

$$I_k = \underset{x \in U_k}{\operatorname{Argmin}}\{f(x) \ : \ 0 < h(x) < h^k_{max}\}.$$

The rules to update h^k_{max}, as described in [5], guarantee that the sequence $\{h^k_{max}\}_{k \in \mathbb{N}}$ is non-increasing. Furthermore, the sequence is bounded below by 0 so it converges. The progressive barrier has also been adapted in a trust region context [4].

The notion of a refined subsequence [5] also needs to be defined to analyze the convergence of the method. Let $\mathbb{U} \subseteq \mathbb{N}$ the subset of the indices of unsuccessful iterations. If the poll was performed around an element $x_k^F \in F_k$, with $k \in \mathbb{U}$, then x_k^F is called "feasible minimal frame center" and if the poll was done around $x_k^I \in I_k$, with $k \in \mathbb{U}$, then x_k^I is called "infeasible minimal frame center".

Definition 2.3. A subsequence of the MADS-PB minimal frame centers $\{x_k\}_{x \in K}$, with $K \subseteq \mathbb{U}$ is a refining subsequence if $\{\Delta^k\}_{k \in K}$ converges to 0. The limit of a convergent refining subsequence $\hat{x}$ is called a refined point. If $\lim_{k \in L} \frac{d_k}{||d_k||}$ converges (to say $v \in \mathbb{R}^n$), with $L \subseteq K$ and poll direction $d_k \in D_k(x_k)$, and if $x_k + \delta^k d_k \in X$ for infinitely $k \in L$, then the limit v is said to be a refining direction of $\hat{x}$.

The analysis of the progressive barrier also uses the definition of the hypertangent cone. We use the definition from [6], but an equivalent one is found in [16].

Definition 2.4. Let $A \subseteq \mathbb{R}^n$ and $\hat{x} \in A$. Then $v \in \mathbb{R}^n$ in an hypertangent vector to the set A at the point $\hat{x}$ if there exists $\varepsilon \in \mathbb{R}^*_+$ such that

$$y + tw \in A \text{ for all } y \in A \cap B_\varepsilon(v), w \in B_\varepsilon(v) \text{ and } 0 < t < \varepsilon\}.$$

$T^H_A(\hat{x})$ is the set of all the hypertangent vectors to A at $\hat{x}$ and is called the hypertangent cone of A at $\hat{x}$.

Jahn [16] generalizes derivative the Clarke derivative [12] to take the domain X into account:

Definition 2.5. The generalized gradient of Clarke of $f : X \mapsto \mathbb{R}$ at $\hat{x} \in X$ in the direction $d \in \mathbb{R}^n$ is the following limit, if it exists:

$$f^\circ(x; d) \ = \lim_{y \to \hat{x}, y \in X} \ \sup_{t \downarrow 0, y + tv \in X} \frac{f(y + td) - f(y)}{t}.$$

Two assumptions are made.
Assumption A1: There exists some x^0 provided by the user in V^0 such that $x^0 \in X$, $f(x^0)$, $h(x^0)$ are both finite.
Assumption A2: All trial points generated by the algorithm lie in a compact set.

It is now possible to describe two of the main convergence results of the progressive barrier. Both assumptions come directly from [5].

Theorem 2.6. *Let assumptions A1 and A2 hold and assume that the algorithm generates a refining subsequence $\{x^F_k\}_{k \in K}$, with $x^F_k \in F_k$ converging to a refined point $\hat{x}^F$ in X near which f is lipschitz. If $v \in T^H_X(x^F)$ is a refining direction for $\hat{x}^F$, then $f^\circ(\hat{x}^F; v) \geq 0$.*

The second result is a similar theorem but on h with a refining subsequence of infeasible elements.

Theorem 2.7. *Let assumptions A1 and A2 hold, and assume that the algorithm generates a refining subsequence $\{x^I_k\}_{k \in K}$, with $x^I_k \in I_k$ converging to a refined point $\hat{x}^I$ in X near which h is lipschitz. If $v \in T^H_X(x^I)$ is a refining direction for $\hat{x}^I$, then $h^\circ(\hat{x}^I; v) \geq 0$.*

Since the union of all normalized refining MADS directions is dense in the unit sphere [1], Theorem 2.6 gives conditions ensuring that the method produces a limit point that satisfies nonsmooth necessary optimality conditions for the minimization of f over Ω, and Theorem 2.7 gives nonsmooth necessary optimality conditions for the minimization of h over X.

3 Scaling of the Output: Impact on the Constraint Violation Function

Section 2 described the progressive barrier technique. The definition of the constraint violation function is impacted by the constraint scaling. In the introduction, two formulations of an equivalent optimization problem were given, leading to two formulations of the constraint violation function:

$$h(x) = \sum_{i=1}^{m} \max\left(0, c_i(x)\right)^2 \quad \text{ or } \quad h(x) = \sum_{i=1}^{m} \max\left(0, \frac{c_i(x)}{a_i}\right)^2.$$

In this section, we analyze three different weights to scale the constraints and study its impact on the progressive barrier.

In order to do that, we generalize the constraint violation function definition by adding a second argument containing the weights.

Definition 3.1. Let $0 < a \in \mathbb{R}^n$ be a scaling parameter. The constraint violation function $h : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is defined by

$$h(x;a) = \begin{cases} \displaystyle\sum_{i=1}^{m} \max\left(0, \frac{c_i(x)}{a_i}\right)^2 & \text{if } x \in X \\ +\infty & \text{otherwise.} \end{cases}$$

Since a has positive components, then for $x \in \mathbb{R}^n$, x is feasible if and only if $h(x;a) = 0$. Also if the vector from $\mathbb{R}^n$ with only ones as components is denoted $\mathbf{1}$, then $h(\cdot;\mathbf{1})$ corresponds to the constraint violation function from Definition 2.1.

3.1 Three different weights

Three weightings are used, based on a sequence $\{a^k\}_{k\in\mathbb{N}}$ of vectors with positive elements to create the sequence of constraint violation functions $\{h(\cdot;a^k)\}_{k\in\mathbb{N}}$. All sequences $\{a^k\}_{k\in\mathbb{N}}$ are initialized by $a^0 = \mathbf{1}$.

First violation. The first weighting aims to correct the scaling as soon as possible. In order to do that, for a given constraint, the weight will be set to value of the first encountered violation of that constraint. More formally, let $i \in [\![1;m]\!]$ be the index of a constraint. The value a_i is set equal to $c_i(x)$ where x is the first point found in V^k such that $c_i(x) > 0$. Thus, the weighting is done as soon as it matters, so when a candidate point x that violates the constraint: $c_i(x) > 0$. Moreover, when it has been modified once, all subsequent a_i^k will keep the same value throughout the rest of the optimization process.

Recalculating the values of $h(x;a^k)$, for all $x \in V^k$ is a problem that is asked at most once per constraint. However, two drawbacks are anticipated: (i) The value of the weighting is entirely dependant of the value found by the first violation, which can be very different by the usual values returned by that constraint. This can create an unbalance with respect to the other constraints. (ii) if a_i^k is very small, computational difficulties may arise due to divisions by small numbers.

A second weighting uses the median in order to avoid those drawbacks.

Median violation. The second option is to wait until the constraint is violated a total of n times, where n is the dimension of the problem. This strategy provides a more representative sample. At iteration $k \in \mathbb{N}$, let $j \in \mathbb{N}$ denote the number of time the constraint c_i was violated. Let $\{x^1, \ldots, x^j\} \subseteq \mathbb{R}^n$ be the corresponding points that violate the constraint c_i, ordered by increasing values of $c_i(x^j)$, $j \in [\![1;n]\!]$. Then, let

$$a_i^k = \begin{cases} 1 & \text{if } j < n \\ c_i(x^{\lceil \frac{n}{2} \rceil}) & \text{if } j \geq n \end{cases}$$

be the violation associated to the median. Just like for the weighting with the first violation, the changes of weighting occurs only once per constraint.

Even if the risks of having a weighting that is not adapted are reduced, because the median is used, it is always possible that the n first violations are not representative of the values usually taken by that constraint.

Maximum violation. The third weighting takes the value of the largest violation of the constraint. More precisely, the sequence $\{a_i^k\}_{k\in\mathbb{N}}$ of general term

$$a_i^k = \begin{cases} 1 & \text{if } \{c_i(x) : x \in V^k, c_i(x) > 0\} = \emptyset \\ \max\limits_{x\in V^k} c_i(x) & \text{otherwise} \end{cases}$$

for all $i \in [\![1; m]\!]$. As a consequence, $0 \leq \frac{c_i(x)}{a_i^k} \leq 1$ for every $x \in V^k$. This makes the constraints' magnitude comparable. This dynamic scaling is well adapted to binary constraints since the coefficient is then equal to 1.

That way to calculate the a_i^k may lead to three drawbacks. (i) if a very high value is found compared to the other typical valued found for that constraint, a_i^k risks to lower significantly $\frac{c_i(x)}{a_i^k}$ and lower too strongly the importance given to that constraint. (ii) an arbitrarily high value (such as 10^{20}) is returned for some blackbox problems, when there is an error (provoked for example by a hidden constraint). In that case, it has a direct impact on the value of a_i^k and on the calculation of h^k. (iii) we have to update a_i^k each time a new higher positive value is found for that constraint. Thus, it is required to check every time if the weights need to be updated, when the other methods had only at most one change of the weight by constraint.

$\boxed{3.2}$ Impact on the progressive barrier

When there is no scaling of the output, the function h remains the same throughout the algorithm. The constraint violation functions are calculated with the same formula, so there is no need to recalculate for elements that have been evaluated previously. This is no longer the case with scaling of the output, since the weightings change the way to calculate h (through the sequence $\{h^k\}_{k\in\mathbb{N}}$). This leads to many changes on the progressive barrier when one of the coefficients changes.

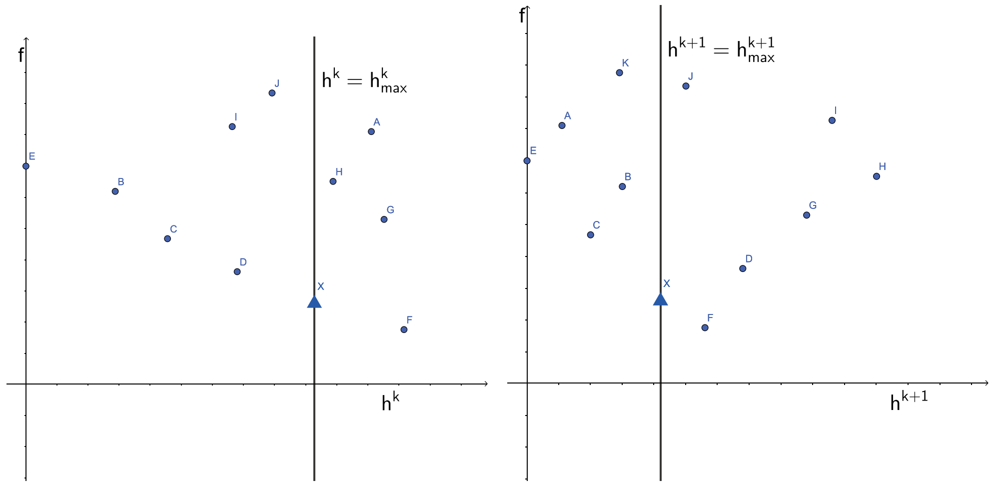
Firstly, h_{max}^k should be updated accordingly to $\{h(\cdot; a^k)\}_{k\in\mathbb{N}}$. For example, it is possible that, at iteration $k \in \mathbb{N}$, if h_{max}^k does not have a different update rule when $h(\cdot; a^k)$ changed, that all the points be on the other side of the barrier, which means that for all $x \in V^k$ that are infeasible, $h(x; a^k) > h_{max}^k$. But, in that case, all the points from the cache could be rejected from the progressive barrier. Lacking of points, MADS terminates. So, an update of h_{max}^k becomes necessary.

Figure 1 illustrates how h_{max}^k is determined when a coefficient changes.

At iteration $k \in \mathbb{N}$ of MADS, if there is at least one infeasible point then h_{max}^k is chosen such that it has the line of equation $h = h_{max}^k$ go through one of the points in the corresponding diagram h vs f from the traditional progressive barrier. This point is denoted by $x_{max}^k \in \mathbb{R}^n$, and represented by the triangle X in the left diagram $h(\cdot; a^k)$ vs f of Figure 1. So after the change of one of the coefficient and the update on h^{k+1} compared to $h(\cdot; a^k)$, the point X changes position on the $h(\cdot; a^{k+1})$ vs f, as seen on the right image of the figure. When $h(\cdot; a^k) \neq h(\cdot; a^{k+1})$, it is decided to choose h_{max}^{k+1} such that the line of equation $h(\cdot; a^{k+1}) = h_{max}^{k+1}$ still goes through the point X in the diagram $h(\cdot; a^{k+1})$ vs f. So, it gives the guarantee that at least one point will not be cut by the progressive barrier, which is the point x_{max}^k. This will be called " the update of h_{max}^k ".

$\boxed{3.3}$ Convergence analysis

The scaling of the output impacts the progressive barrier and its convergence analysis. Several weightings have been developed and they do not all have the same impact. Those



a) Beginning of iteration k. b) Beginning of iteration $k+1$.

Figure 1: Update of $h(\cdot; a^k)$ and h^k_{max}.

weightings are analysed separately.

First and median violation. The two first weightings are grouped because they have identical convergence analyses. Firstly, in both cases, there is at most one change in the weights per constraint, that weight being equal to 1 initially and taking a value noted $a_i \in \mathbb{R}^*_+$. That means that, for all $i \in [\![1; m]\!]$, there is an iteration number $k_i \in \mathbb{N}$ such that for all $k \in [\![0; k_i[\![$, $a_i^k = 1$ and for all $k \in [\![k_i; +\infty[\![$, $a_i^k = a_i > 0$. So $\{a_i^k\}_{k\in\mathbb{N}}$ converges to a positive value. Then, by construction $h(\cdot; a^k) = h(\cdot; a^{k_{\max}})$ for every $k \geq k_{\max}$ where $k_{max} = \max\{k_i : i \in [\![1; m]\!]\}$. So, from the iteration k_{max}, the sequence $\{h(\cdot; a^k)\}_{k\in\mathbb{N}}$ always takes the same value, and thus the convergence analysis of the progressive barrier mentioned in 2 and described in [5] is still valid from that iteration number.

Maximum violation. The reasoning for the convergence analysis of the first violation and the median weighting is not possible for the weighting using the maximum. This is because there are cases where the algorithm will converge to the optimal solution with an infinite number of updates if an infinite budget of evaluation is given. For example, consider $f : [0; 2] \mapsto \mathbb{R}$ defined by its general term $f(x) = -x$ and $c : [0; 2] \mapsto \mathbb{R}$ defined by

$$c(x) = \begin{cases} 0 & \text{if } x \in [0; 1] \\ 3 - x & \text{if } x \in]1; 2]. \end{cases}$$

The optimization problem

$$\begin{cases} \min\limits_{x\in\mathbb{R}} & f(x) \\ \text{subject to} & c(x) \leq 0 \\ & 0 \leq x \leq 2 \end{cases}$$

has a single optimal solution $x^* = 1$.

If MADS is used with the starting point $x_0 = 0.9$, points will be generated from either side of x^*. As the algorithm progresses, the mesh size will diminish and points higher and closer to 1 will be generated, which will produce higher values of the constraint c. Thus, there will be an infinite number of points which will lead to a new scaling of h. However, some properties can be proven.

Theorem 3.2. *The sequence of functions* $\{h(\cdot; a^k)\}_{k\in\mathbb{N}}$ *converges.*

Proof. For all $i \in [\![1; m]\!]$, it should be noted that the sequence $\{a_i^k\}_{k\in\mathbb{N}}$ is non-decreasing from a certain rank (either from the first violation of the i-th constraint, if this constraint is violated at least once, or from the rank 0 if the constraint is never violated), by construction of the weighting.

Since $\{a_i^k\}_{k\in\mathbb{N}}$ is a positive sequence non-decreasing from a certain rank, two cases can occur. Either that sequence is majored and it converges to a real number $a_i > 0$, either it is not majored and it diverges to $+\infty$. In both cases, $\left\{\frac{1}{a_i^k}\right\}_{k\in\mathbb{N}}$ converges to a non-negative value. The convergence of that last sequence implies the convergence of $\{h(\cdot; a^k)\}_{k\in\mathbb{N}}$, which proves the theorem. □

In the case where the sequence $\{a_i^k\}_{k\in\mathbb{N}}$ for a $i \in [\![1; m]\!]$ diverges to $+\infty$, the conventions $a_i = +\infty$ and $\frac{1}{a_i} = 0$ are adopted.

The next theorem is about the non-increasing aspect of the sequence $\{h_{max}^k\}_{k\in\mathbb{N}}$. It was one of the properties the convergence analysis of the progressive barrier [5].

Theorem 3.3. *The sequence* $\{h_{max}^k\}_{k\in\mathbb{N}}$ *is non-increasing from a certain rank.*

Proof. The proof of the previous theorem showed that, for all $i \in [\![1; m]\!]$ the sequence $\{a_i^k\}_{k\in\mathbb{N}}$ is non-decreasing from a certain rank. Let note k_i that rank, and define $k_{max} = \max\{k_i : i \in [\![1; m]\!]\}$. From the rank k_{max}, either the value h_{max}^k changes because of a change in the weighting as explained in section 3.2, or it changes because of the way the progressive works originally as explained in 2 and in [5]. In the second case, it is known that h_{max}^k is updated with a lower value. In the first case, let $k \in \mathbb{N}$, $k \geq k_{max}$ and let $\bar{x}$ the element that was used for the update of h_{max}^k. Then $h_{max}^k = h^k(\bar{x})$ and $h_{max}^{k+1} = h(\bar{x}; a^{k+1})$. It is just needed to show that $h(\bar{x}; a^{k+1}) \leq h(\bar{x}, a^k)$. But since, $k \geq k_{max}$, then for all $i \in [\![1; m]\!]$, $0 < a_i^k \leq a_i^{k+1}$, so $0 < \frac{1}{a_i^{k+1}} \leq \frac{1}{a_i^k}$. This is true for all $i \in [\![1; m]\!]$ so $h(\bar{x}; a^{k+1}) \leq h(\bar{x}; a^k)$. Thus, $h_{max}^{k+1} \leq h_{max}^k$. This shows that from the rank k_{max}, whatever the way the barrier has been updated, $h_{max}^{k+1} \leq h_{max}^k$, which proves the theorem 3.3. □

A similar proof can be used to show that, for any given $x \in X$, the sequence $\{h(x; a^k)\}_{k\in\mathbb{N}}$ is non-increasing from the certain rank k_{max}. Furthermore, by construction, the sequence $\{h_{max}^k\}_{k\in\mathbb{N}}$ is minored by 0. And since it is non-increasing from a certain rank, then it converges.

There is a property close from the non-increasing property of $\{h(x; a^k)\}_{k\in\mathbb{N}}$ for all $x \in X$ from the rank k_{max}. In fact, as soon as an element $x \in \mathbb{R}$ is evaluated by the blackbox, the value of $\{h(x; a^k)\}$ cannot increase. This is summarized with Theorem 3.4.

Theorem 3.4. *Let* $k_0 \in \mathbb{N}$. *For any* $x \in V^{k_0} \cap X$, *the sequence* $\{h(x; a^k)\}_{k\geq k_0}$ *is non-increasing.*

Proof. Let $k_0 \in \mathbb{N}$ and $A = \{i \in [\![1; m]\!] : \forall x \in V^{k_0}\ c_i(x) \leq 0\}$. A is the set of the constraints that have not been violated at the beginning of the iteration k_0. Let $x \in V^{k_0} \cap X$. Then, for all $k \in [\![1; m]\!]$:

$$\begin{aligned} h(x;a^k) &= \sum_{i=1}^{m} \max\left(0, \frac{c_i(x)}{a_i^k}\right)^2 \\ &= \sum_{i\notin A} \max\left(0, \frac{c_i(x)}{a_i^k}\right)^2 \text{ (by definition of } A) \end{aligned}$$

Furthermore, for all $i \notin A$, there exists $x^i \in V^{k_0}$, such that $c_i(x^i) > 0$. But the sequence $\{a_i^k\}_{k\in\mathbb{N}}$ is non-decreasing from the first violation of the constraint c_i. So, for all $i \notin A$, $\{a_i^k\}_{k\geq k_0}$ is non-decreasing and has positive values. Thus, $i \notin A$, $\{\frac{1}{a_i^k}\}_{k\geq k_0}$ is non-decreasing and has positive values. It leads to the fact that $\{h(x;a^k)\}_{k\geq k_0}$ is non-decreasing, which proves the theorem. □

Theorem 3.4 shows that the only elements $x \in X$ for which the sequence $\{h(x;a^k)\}_{k\geq k_0}$ is not non-increasing are among the points that have not been evaluated yet. It shows also that from the diagram $h(\cdot;a^{k_0})$ vs f, a point will never move to the right on the diagram $h(\cdot;a^{k_0+1})$ vs f.

In the progressive barrier (see Section 2), there are convergence analysis results both on f and h. The results on f remain unchanged. However, since h has been substituted by $\{h(\cdot;a^k)\}_{k\in\mathbb{N}}$. Ideally, the best would be if the same results remain for

$$h(\cdot;a) = \lim_{k\to+\infty} h(\cdot;a^k).$$

In [5], the results on h relies on the hypothesis that h is lipschitz at the convergent point. The same hypothesis could be made on $h(\cdot;a)$. Another assumption needs to be made.
Assumption A3: For all $i \in [\![1;m]\!]$, c_i is bounded above on X.

Theorem 3.5. *Under assumptions A1, A2 and A3, $\{h(\cdot;a^k)\}_{k\in\mathbb{N}}$ converges uniformly to $h(\cdot;a)$ on X.*

Proof. Let $x \in X$ and $k \in \mathbb{N}$. Since for all $i \in [\![1;m]\!]$, c_i is upper-bounded on X, then

$$C = \max_{i\in[\![1;m]\!]} \{\sup \max(0, c_i(x)), x \in X\}$$

is well defined.

$$\begin{aligned} \left|h(x;a^k) - h(x;a)\right| &= \left|\sum_{i=1}^{m} (\max(0, \frac{c_i(x)}{a_i^k})^2 - \max(0, \frac{c_i(x)}{a_i})^2\right| \\ &= \left|\sum_{i=1}^{m} (\frac{1}{(a_i^k)^2} - \frac{1}{(a_i)^2}) \max(0, c_i(x))^2\right| \\ &\leq C^2 \sum_{i=1}^{m} \left|\frac{1}{(a_i^k)^2} - \frac{1}{(a_i)^2}\right| \end{aligned}$$

This last inequality is true for all $x \in X$, so

$$\sup_{x\in X} \left|h(x;a^k) - h(x;a)\right| \leq C^2 \sum_{i=1}^{m} \left|\frac{1}{(a_i^k)^2} - \frac{1}{(a_i)^2}\right| \to 0$$

Which proves the theorem 3.5. □

The next section will analyse how weightings can be used for the surrogate function of the constraint violation function.

3.4 Impact of a surrogate on the constraint violation function $\tilde{h}$

As described in the introduction, MADS uses the opportunistic strategy. In order to help the opportunistic strategy, it is possible to use a surrogate for each constraint.

In this work, for all $i \in [\![1; m]\!]$, the surrogate of the constraint c_i is noted $\tilde{c_i}$ and is a quadratic model [13] (default choice in Nomad, the optimization software based on MADS). With those surrogates, another surrogate of h (which is equal to $h(\cdot; \mathbf{1})$ with current notations) can be built:

$$\tilde{h}(x; \mathbf{1}) = \begin{cases} \displaystyle\sum_{i=1}^{m} \max\left(0, \tilde{c_i}(x)\right)^2 & \text{if } x \in X \\ +\infty & \text{otherwise.} \end{cases}$$

Currently, if it is supposed that a surrogate $\tilde{f}$ for f and $\tilde{h}(\cdot; \mathbf{1})$ for h is at disposal, the ordering strategy is the following: let x and y, then x is given to the blackbox before y if and only if $(\tilde{f}(x) \leq \tilde{f}(y)$ and $\tilde{h}(x; \mathbf{1}) < \tilde{h}(y; \mathbf{1}))$ or $(\tilde{f}(x) < \tilde{f}(y)$ and $\tilde{h}(x; \mathbf{1}) \leq \tilde{h}(y; \mathbf{1}))$.

Since $\tilde{h}(\cdot; \mathbf{1})$ is important in the way MADS works, a weighting on $\tilde{h}(\cdot; \mathbf{1})$ seems fair. $\tilde{h}$ is chosen as

$$\tilde{h}(x; a^k) = \begin{cases} \displaystyle\sum_{i=1}^{m} \max\left(0, \frac{\tilde{c_i}(x)}{a_i^k}\right)^2 & \text{if } x \in X \\ +\infty & \text{otherwise.} \end{cases}$$

The weightings on h and $\tilde{h}$ can be considered independently. It is possible to test the weighting on h, with $\{h(\cdot; a^k)\}_{k \in \mathbb{N}}$, without doing it on $\{\tilde{h}(\cdot; a^k)\}_{k \in \mathbb{N}}$ and vice versa. It is also possible to use both weightings together. So when surrogates are available, it will be possible to compare the default version (no weightings), and a version with weightings on $\{\tilde{h}(.; a^k)\}_{k \in \mathbb{N}}$.

4 Numerical Results

The numerical results are divided in two sections. Section 4.1 studies cases for which no surrogate functions are used. The absence of surrogates allows to study the numerical impact of the weightings on the progressive barrier. Adding surrogate functions might compensate some flaws, so this section focuses on the progressive barrier. In addition, experimental results compare the three proposed weightings. Section 4.2 uses the best weighting for problems where surrogates are available.

All numerical results are done on Nomad 3.8.0 with the directions generated by OrthoMADS [1] and the budget of evaluation is set to 1500. Data profiles [19] are generated to compare different versions. For each problem and for each algorithm the following test of convergence is performed:

$$f(x_0) - f(x) \leq (1 - \tau)(f(x_0) - f_L), \tag{4.1}$$

where $x_0 \in \mathbb{R}^n$ is the feasible starting point (all algorithms start with the same starting point), f_L the best value found by all the algorithms compared given a budget of evaluation and $\tau \in \mathbb{R}_+$ the wished precision. If, for some problem, an algorithm produces a point $x \in \mathbb{R}^n$ that verifies equation (4.1), this algorithm is said to solve the problem at precision τ. The ordinate of data profiles show the ratio of problem verifying the test convergence at a given precision.

4.1 Without surrogates

In this section, no surrogates for the objective function and the constraints are used. Numerical results are divided in two groups. The first one contains analytical problems and the second contains blackbox problems, described in Tables 1 and 2, respectively.

#	Name	Source	n	m	Bounded
1	PIGACHE	[20]	4	11	yes
2	PVMC	[10]	4	3	no
3	RCBM	[15]	3	2	yes
4	SRMMC	[10]	7	11	yes
5	SMMC	[21]	3	4	yes

Table 1: Five analytical optimization problems from the literature.

#	Name	Source	n	m	Bounded
1	Styrene	[3]	8	11	yes
2	MDO	[23]	7	4	yes
3	Lockwood	[18]	6	4	yes

Table 2: Three blackbox optimization problem from the literature.

In order to test badly scaled problems, a modified version of each problem, called "unbalanced", is created by multiplying the i-th constraint by the coefficient 10^{j_i}, $j_i \in \mathbb{Z}$. The coefficient j_i are listed in Table 3.

For each problem, feasible points are generated using a latin hypercube, either on the entire domain, or around a known feasible point.

Name	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8	j_9	j_{10}	j_{11}
PIGACHE	2	4	-7	-5	-3	1	3	-2	-1	5	7
PVMC	0	3	-3								
RCBM	3	-3									
SRMMC	-1	-2	-3	-4	-5	0	1	2	3	4	5
SMMC	-4	-2	2	4							
Styrene	0	0	0	0	-3	-2	-1	0	1	2	3
MDO	3	0	0								
Lockwood	3	0	0	0							

Table 3: Coefficient j_i for the unbalanced variants of the problems

Analytical problems. The first tests are on the 5 unmodified analytical problems from 100 feasible starting points. This makes a total of 500 instances. The same starting points are used on the "unbalanced" problems.

Figure 2 contains the data profiles from the 500 instances on the unmodified analytical problems. The weighting that uses the maximum violation dominates the other methods on the profiles. In particular, at precision $\tau = 10^{-5}$, it solves 58% of the problems, and all other strategies solve less than 35% of them.

Figure 3 shows the data profiles from the 500 instances on the "unbalanced" analytical problems. These problems appear to be more difficult, as all the curves are slighly lower than the corresponding ones in Figure 2. Once again, the graphs show a domination of

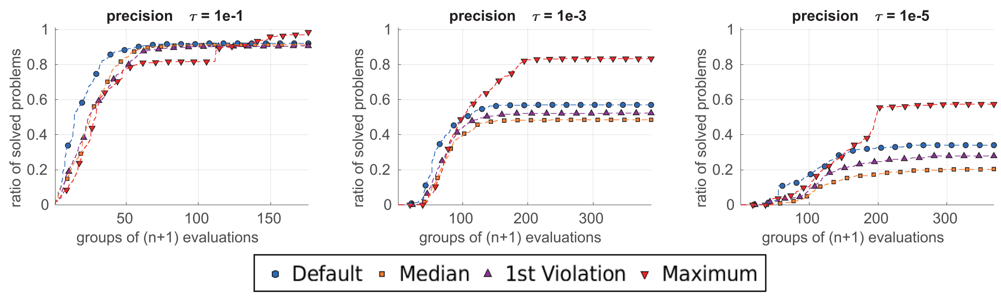


Figure 2: Data profiles for unmodified analytical problems.

the weighting that uses the maximum violation. The two other weightings seem to perform slightly worst than the default strategy that does not alter the weights of the constraints.

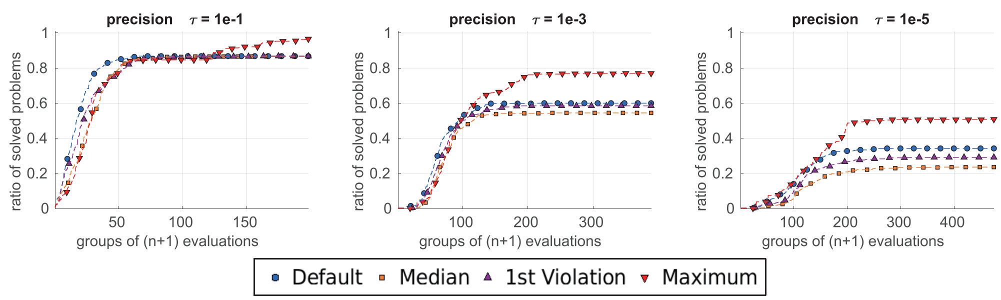


Figure 3: Data profiles for "unbalanced" analytical problems.

The analytical problems show a clear domination of the weighting that uses the maximum violation.

Blackbox problems The blackbox problems are "Styrene", "MDO" and "Lockwood", and their descriptions are found in [3, 7, 9]. For each blackboxes 30 feasible starting points are used.

Figure 4 shows the data profiles on those 90 instances from the 3 unmodified blackboxes at precision $\tau = 10^{-1}$, $\tau = 10^{-3}$ and $\tau = 10^{-5}$. The scaling methods, or even having a scaling or not, does not seem to be an important impact. All curves are very close to each other.

Tests are then made on instances from "unbalanced" blackboxes from the same starting points. Figure 5 shows that the weighting that uses the first violation is outperformed by the others. This is the case at precision $\tau = 10^{-2}$ et $\tau = 10^{-3}$. Unlike the plots on the unmodified blackboxes, where it had worst results than the default version, the weighting with the median violation performs as well as the default version. Concerning the weighting with the maximum violation, it performs as well as the default variation at precision $\tau = 10^{-1}$ and $\tau = 10^{-3}$ but performs slightly better at $\tau = 10^{-2}$. However, this is not very significant as all curves are very close to each other.

The blackbox problems do not show a clear domination of the scaling strategies.

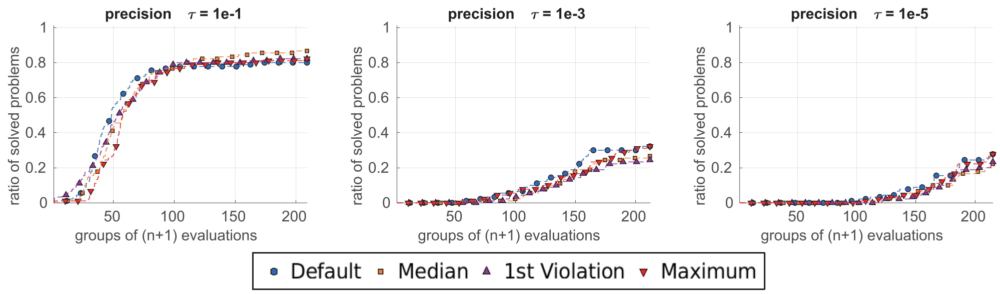


Figure 4: Data profiles for unmodified blackbox problems.

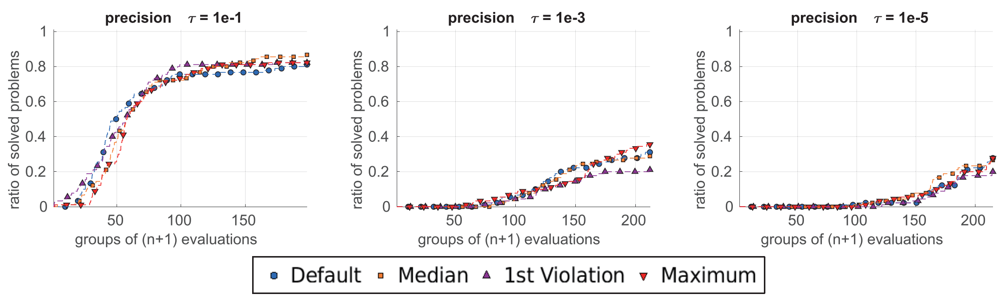


Figure 5: Data profiles with for "unbalanced" blackbox problems.

4.2 With surrogates

This last subsection on numerical experiments compares the scaling strategies on the blackbox problems using surrogate functions (the quadratic models described in [13]) for the objective function and the constraints. The first one, represented by circles on the profiles, is the default version that does not rescale the output. The second one, represented by squares, only adds weights on the surrogate constraint violation function $\tilde{h}$. The third one, represented by triangles, only adds weights on both the constraint violation h and on its surrogate $\tilde{h}$. The weightings are done using the maximum violation of each constraint.

Figure 6 looks at unmodified blackboxes. At the weighting $\tau = 10^{-1}$, the weightings show no improvements and the tested blackboxes. At precision $\tau = 10^{-2}$ and $\tau = 10^{-3}$, the weighting done only on $\tilde{h}$ shows slightly better results than a weighting both on h and $\tilde{h}$ and the default version. However, the differences are very small and the results in the sub-section 4 showed very few satisfactory results on unmodified blackboxes.

The "unbalanced" blackboxes are also tested. They were unbalanced the same way as in sub-section 4. Figure 7 shows the results for the "unbalanced" blackboxes. It shows that the version where the weighting is made both on h and $\tilde{h}$ dominates the two other versions. It is interesting to note that at precision $\tau = 10^{-2}$, the weighting on $\tilde{h}$ only dominates the default version but is dominated by the other one. This shows the cumulative effects of the weighting on constraints. Even if it better to have a weighting on h or $\tilde{h}$, the best is to have the weighting on both functions h and $\tilde{h}$.

Compared to Figure 6, Figure 7 shows the advantages of weighting of constraints on problems where constraints are not well scaled.

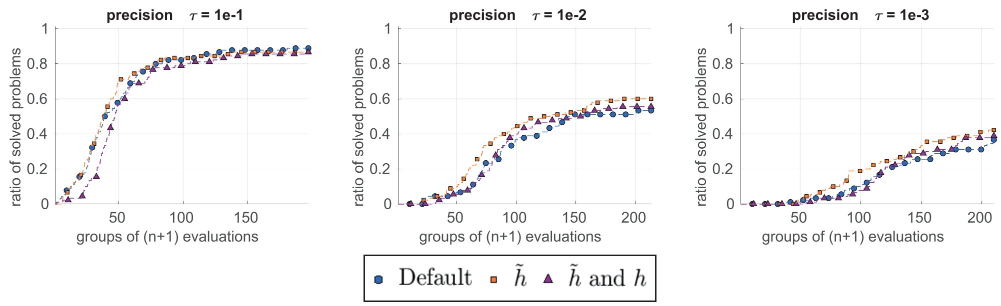


Figure 6: Data profiles for unmodified blackboxes

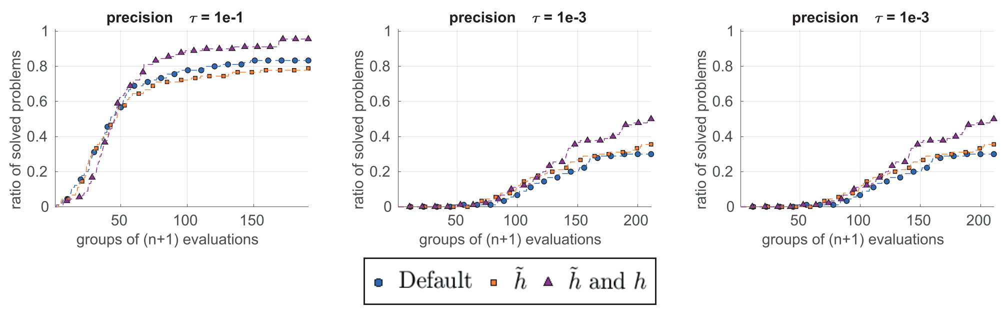


Figure 7: Data profiles for "unbalanced " blackboxes.

5 Discussion

This work offers several weightings techniques of the constraints in order to compensate scaling issues in the formulation of a blackbox optimization problem. These weightings rely on the values taken by the constraints: the first violation, the median violation on the n first violations and the highest violation.

From a theoretical point of view, the convergence analysis followed that of the progressive barrier. It was shown that the two first weightings had no impact on the convergence analysis from the rank where all the weightings were calculated. For the last weighting, the properties of $\{h^k_{max}\}_{k\in\mathbb{N}}$ were preserved.

Numerical experiments on the analytical problems suggest that the strategy with the maximum violation is preferable to the others on both the unmodified and the unbalanced problems. The results on the blackbox problems were inconclusive. None of the method clearly dominates the others. A final set of experiments were conducted on these blackbox problems, with the utilization of a surrogate. Here, the results on the unbalanced blackboxes revealed a dominant strategy. Applying the weighting with the maximal violation on both the constraint violation function h and its surrogate $\tilde{h}$ is more efficient than the other strategies.

This work focused on weighting the constraints so that they have all approximately the same importance. However, other choices can be made. Learning the importance of the constraints through the optimization process is a possibility. A selection of the most influential variables in a context of blackbox optimization problem was achieved in [2]. A similar analysis for the constraints could be considered in future work.

References

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

[2] S. Alarie, N. Amaioua, C. Audet, S. Le Digabel and L.-A. Leclaire, Selection of variables in parallel space decomposition for the mesh adaptive direct search algorithm, *Les cahiers du GERAD, G-2018-38*, 2018.

[3] C. Audet, V. Béchard and S. Le Digabel, Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search, *J. Global Optim.* 41 (2008) 299–318.

[4] C. Audet, A.R. Conn, S. Le Digabel and M. Peyrega, A progressive barrier derivative-free trust-region algorithm for constrained optimization, *Comput. Optim. Appl.* 71 (2018) 307–329.

[5] C. Audet and J.E. Dennis, A progressive barrier for derivative-free nonlinear programming, *SIAM J. Optim.* 20 (2009) 445–472.

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

[7] C. Audet, A. Ianni, S. Le Digabel and C. Tribes, Reducing the number of function evaluations in mesh adaptive direct search algorithms, *SIAM J. Optim.* 24 (2014) 621–642.

[8] C. Audet, S. Le Digabel and C. Tribes, Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization, *Optim. Eng.* 17 (2016) 333–358.

[9] C. Audet and C. Tribes, Mesh-based Nelder-Mead algorithm for inequality constrained optimization, *Comput. Optim. Appl.* 71 (2018) 331–352.

[10] L.C. Cagnina, S. Esquivel and C.A. Coello Coello, Solving engineering optimization problems with the simple constrained particle swarm optimizer, *Informatica(Ljubl.)* 32 (2008) 319–326.

[11] J. Chen and P.Z.G. Qian, Latin hypercube designs with controlled correlations and multi-dimensional stratification, *Biometrika* 101 (2014) 319–332.

[12] F. Clarke, *Optimization and Nonsmooth Analysis*, Society for Industrial and Applied Mathematics, 1990.

[13] A.R. Conn and S. Le Digabel, Use of quadratic models with mesh-adaptive direct search for constrained black box optimization, *Optimization Methods and Software* 28 (2013) 139–158.

[14] R. Fletcher and S. Leyffer, Nonlinear programming without a penalty function, *Math. Program.* 91 (2002) 239–269.

[15] A. H. Gandomi, X.-S. Yang and A. H. Alavi, Mixed variable structural optimization using firefly algorithm, *Comput. & Structures* 89 (2011) 2325–2336.

[16] J. Jahn, *Introduction to the Theory of Nonlinear Optimization*, Springer, 1994.

[17] S. Le Digabel and S.M. Wild, A Taxonomy of Constraints in Simulation-Based Optimization, Les cahiers du GERAD, G-2015-57, 2015.

[18] L.S. Matott, K. Leung and J. Sim, Application of MATLAB and Python optimizers to two case studies involving groundwater flow and contaminant transport modeling, *Computers & Geosciences* 37 (2011) 1894–1899.

[19] J.J. Moré and S.M. Wild, Benchmarking derivative-free optimization algorithms, *SIAM J. Optim.* 20 (2009) 172–191.

[20] F. Pigache, F. Messine and B. Nogarede, Optimal design of piezoelectric transformers: A rational approach based on an analytical model and a deterministic global optimization, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 54 (2007) 1293–1302.

[21] R.G. Regis, Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points, *Eng. Optim.* 46 (2014) 218–243.

[22] L.A. Sarrazin-Mc Cann, Opportunisme et ordonnancement en optimisation sans dérivées, *Master's thesis, École Polytechnique de Montréal*, 2018.

[23] C. Tribes, J.-F. Dubé and J.-Y. Trépanier, Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design, *Eng. Optim.* 37 (2005) 775–796.

Manuscript received 17 September 2019
revised 24 March 2020
accepted for publication 1 April 2020

Charles Audet
GERAD and Polytechnique Montréal
E-mail address: charles.audet@gerad.ca

Gilles Caporossi
GERAD and HEC Montréal
E-mail address: gilles.caporossi@gerad.ca

Stéphane Jacquet
UQAC
E-mail address: stephane.jacquet@gerad.ca