

A MADS algorithm with a progressive barrier for derivative-free nonlinear programming *

Charles Audet [†] J.E. Dennis Jr. [‡]

June 26, 2007

Abstract

We propose a new algorithm for general constrained derivative-free optimization. As in most methods, constraint violations are aggregated into a single constraint violation function. As in filter methods, a threshold, or barrier, is imposed on the constraint violation function, and any trial point whose constraint violation function value exceeds this threshold is discarded from consideration. In the new algorithm, unlike the filter method, the amount of constraint violation subject to the barrier is progressively decreased as the algorithm evolves.

Using the Clarke nonsmooth calculus, we prove Clarke stationarity of the sequences of feasible and infeasible trial points. The new method is effective on two academic test problems with up to 50 variables, which were problematic for our GPS filter method. We also test on a chemical engineering problem. The proposed method generally outperforms our LTMADS in the case where no feasible initial points are known, and it does as well when feasible points are known.

Keywords : Mesh adaptive direct search algorithm, filter algorithm, barrier approach, constrained optimization, nonlinear programming.

*Work of the first author was supported by FCAR grant NC72792 and NSERC grant 239436-05. The second author was supported by LANL 94895-001-04 34, and both authors were supported by AFOSR F49620-01-1-0013, the Boeing Company, 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 (www.caam.rice.edu/~dennis, dennis@rice.edu)

1 Introduction

In [5], we modified the filter approach pioneered by Fletcher and Leyffer [15] to treat general nonlinear constraints without derivatives by a generalized pattern search (GPS-filter) algorithm. A filter aggregates all constraints into a single constraint violation function and treats the optimization problem as an unconstrained biobjective problem with priority given to feasibility versus a low objective function value.

The convergence results we were able to provide for the GPS-filter method are limited by the GPS restriction to a fixed finite set of directions in which the space of variables is explored locally. We gave some pathological cases in which the sequence of GPS-filter iterates fails to produce limit points that satisfy desirable optimality conditions. Still, the GPS-filter has solved some difficult industrial problems despite less than desired theoretical support [2, 5, 20, 22, 21].

The mesh adaptive direct search class (MADS) class of algorithms [6] generalizes GPS [26, 4] by removing the restriction of the local exploration of the space of variables to a fixed finite set of directions. This enables the algorithm to handle constraints, including nonlinear ones, by the extreme barrier approach in which infeasible points are simply rejected from consideration. We will call this algorithm MADS-EB, where EB stands for *extreme barrier*.

Specific MADS-EB limit points satisfy optimality conditions that depend on the local smoothness of the objective function under a constraint qualification by Rockafellar [25], i.e., that the hypertangent cone to the feasible region is nonempty at the limit point. This is a weaker constraint qualification than is usually assumed for derivative-based methods such as SQP. For a strictly differentiable [18] objective function, the convergence analysis shows that MADS-EB generates limit points that are KKT points. If the objective function is only Lipschitz near the limit point, then it is a Clarke stationary point.

In this paper, we propose to combine ideas from the GPS-filter and MADS-EB approaches for general nonlinear optimization

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

where $\Omega = \{x \in X : c_j(x) \leq 0, j \in J\} \subset \mathbb{R}^n$ and $f, c_j : X \rightarrow \mathbb{R} \cup \{\infty\}$ for all $j \in J = \{1, 2, \dots, m\}$, and where X is a subset of \mathbb{R}^n . Some useful terminology differentiates between constraints that must always be satisfied, such as those that define X , and constraints that need only be satisfied at the solution, such as the $c_j(x) \leq 0$. The former are *closed* constraints and the latter are *open* constraints.

Our proposed approach is called MADS with a progressive barrier, MADS-PB. No differentiability assumptions on the objective and constraints are required

to apply this new algorithm. However, the strength of the optimality results at a limit point \hat{x} is closely tied to the local smoothness of the functions and to properties of the tangent cones to Ω and X at \hat{x} . MADS-PB is shown theoretically and numerically to work for all the cautionary examples we gave for the GPS-filter method. As for MADS-EB, we prove convergence of MADS-PB to Clarke stationary points.

We call the algorithm proposed here a *progressive barrier* algorithm, as opposed to an *extreme barrier* algorithm. The distinction is as follows. An extreme barrier algorithm rejects all infeasible trial points. A progressive barrier algorithm places a threshold on the constraint violation it allows, and progressively tightens this threshold as the algorithm progresses. We do not use a filter, but we do use the notion of dominance fundamental to filters to determine adaptively how to reduce the constraint violation threshold at each iteration. The user has the discretion within MADS-PB to specify certain constraints to be closed and always treated by the extreme barrier approach.

Given the strength of the MADS-EB convergence results and the positive reports we hear from users, it is reasonable to ask what motivates us to undertake this research rather than to abandon the filter in favor of the barrier for constraints. There are several reasons.

- First, the MADS-EB approach requires a feasible starting point. Yet for some practical problems like the aircraft planform results given in [2], there is no initial feasible point. In fact, the first feasible point found by the GPS-filter was acceptable as a solution. In the MADS-PB method, a user can decide to treat a constraint as open until it becomes feasible and then move it into the closed constraints, which are treated by the extreme barrier approach and whose feasible region defines X .
- Second, some users have observed that the GPS-filter method provides useful information about the sensitivity of the solution with respect to the constraints. The extreme barrier approach does not provide that information, but MADS-PB does.
- Third, in an industrial optimization context, the functions defining the problem often are provided as a black-box computer codes. The codes read input variables and output some values, or else they may fail to return a value for various reasons. In this case, the function value is set to infinity. There are sometimes constraints that return a boolean value that indicates feasibility or not without quantifying the infeasibility. There might also be some constraints that must be satisfied in order for the simulation to work because the objective function f or some constraints c_j might not be defined outside X .

These Boolean constraints are incorporated in the set X , and X is handled by the proposed algorithm through the barrier approach. Other constraints quantify the amount by which they are violated, and so they may be treated by a filter or progressive barrier if the user desires.

- It is not unreasonable to expect in nonlinear optimization that allowing constraint violations in the course of solving a problem often enables one to solve the problem with fewer function and constraint evaluations. This might happen if the domain is disjoint, or if feasibility at every step requires the iteration to take small steps along an erratic boundary to get into a better part of the feasible region. Of course this might be mitigated in some problems for which checking the constraints may be less expensive than computing the objective function, and neither MADS version asks for function values at points outside X [19].

Thus, we are providing the user with options by extending MADS-EB to the MADS-PB algorithm, including the option of deciding which constraints should be satisfied before evaluating the objective function.

The paper is organized as follows. The next section describes the new MADS-PB algorithm. Section 3 breaks down the convergence analysis into three cases. First it analyzes subsequences of feasible iterates; the results are similar to those of MADS-EB [6]. Second, subsequences of infeasible iterates are considered; results on a measure of the constraint violations are presented. Finally, we analyze the case where a subsequence of infeasible iterates converges to a feasible point. Numerical results are presented in Section 4 on three test problems.

2 A MADS algorithm with a progressive barrier

MADS-PB is an iterative algorithm, and the iteration counter is denoted by the index $k \in \mathbb{N}$. We will use $V_k \subset X \subseteq \mathbb{R}^n$ to represent the set of points where the f and all c_j function values have been evaluated by the start of iteration k . This means that they satisfy the closed constraints. Each set V_k contains a finite number of points. The set of initial points is V_0 . It may contain feasible and infeasible points with respect to the open constraints.

2.1 The barrier on constraint violation

Adapting the filter terminology [15] to a mixture of open and closed constraints, we define the constraint violation function

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

With this definition, $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is a nonnegative function, and $x \in \Omega$ if and only if $h(x) = 0$. Moreover, if $0 < h(x) < \infty$ then $x \in X \setminus \Omega$. The constraint violation function could have been defined in other ways - the ℓ_1 norm is commonly used. We favor the squared violations since it was shown in [5] that this performs better in the present context, as it passes on any smoothness of the constraints to h .

We introduce the non-negative barrier parameter h_k^{\max} , which is set at each iteration. Any trial point whose constraint violation function value exceeds h_k^{\max} is rejected from consideration. The barrier parameter h_k^{\max} is non-increasing with the iteration number k ; the rules for updating it at the end of an iteration are presented in Section 2.4. In [6], an extreme barrier is used, i.e., $h_k^{\max} = 0$ for all k , and every infeasible trial point is rejected from consideration.

The initial barrier parameter $h_0^{\max} \geq 0$ can be fixed by the user. Alternatively, the default value implemented in our code is $h_0^{\max} = \infty$. In the numerical results, we will see that setting h_0^{\max} to a smaller value can be useful when the initial points are all feasible.

The algorithm proposed here does not require that the initial points are feasible with respect to the open constraints c_j , $j \in J$. The algorithm can be applied to a problem that satisfies only the first of the following assumptions. Its analysis requires the remaining two assumptions. We will say more about the second and third assumptions when we repeat them as they come into play. In particular, the hypertangent and generalized derivative are defined in Section 3. We list them formally here to refer to them all together later.

- A1 :** There exists some point x_0 in the user-provided set V_0 such that $x_0 \in X$, $f(x_0) < \infty$ and $h(x_0) < h_0^{\max}$.
- A2 :** All trial points considered by the algorithm lie in a bounded set.
- A3 :** For every hypertangent direction $v \in T_\Omega^H(\hat{x}) \neq \emptyset$, there exists an $\varepsilon > 0$ for which $h^\circ(x; v) < 0$ for all $x \in \{x \in X \cap B_\varepsilon(\hat{x}) : h(x) > 0\}$.

The algorithm and its analysis partition the trials point into two sets: the feasible and the infeasible points. The infeasible ones that do not satisfy the closed

constraints $x \in X$ are rejected through the barrier approach. The infeasible ones in $\Omega \setminus X$ are treated differently. We next introduce definitions of best feasible and infeasible incumbents at iteration k .

2.2 Feasible and infeasible incumbents

At the start of iteration k , two sets of incumbent solutions are constructed from the set V_k . The first one is the set of feasible incumbents. It consists of the feasible points found by the start of iteration k that have the best objective function value.

Definition 2.1 *At iteration k , the set of feasible incumbent solutions is defined to be*

$$F_k = \{\arg \min_{x \in V_k} \{f(x) : h(x) = 0\}\}.$$

The set of infeasible incumbent solutions is constructed with the notion of dominance used by filter algorithms [15, 5]. We first introduce the set of infeasible undominated points.

Definition 2.2 *At iteration k , the set of infeasible undominated points is defined to be*

$$U_k = \{x \in V_k \setminus \Omega : \nexists y \in V_k \setminus \Omega \text{ such that } y \prec x\},$$

where $y \prec x$ signifies that $h(y) < h(x)$ and $f(y) \leq f(x)$, or that $h(y) \leq h(x)$ and $f(y) < f(x)$.

The set of infeasible incumbent solutions is constructed using the set of undominated infeasible points U_k , and the barrier parameter h_k^{\max} .

Definition 2.3 *At iteration k , the set of infeasible incumbent solutions is defined to be*

$$I_k = \{\arg \min_{x \in U_k} \{f(x) : 0 < h(x) \leq h_k^{\max}\}\}.$$

These two sets, F_k and I_k , allow definition of the incumbent values at iteration k . The incumbent feasible f -value is defined to be

$$f_k^F = \begin{cases} \infty & \text{if } F_k = \emptyset \\ f(x), \text{ for any } x \in F_k & \text{otherwise,} \end{cases} \quad (2)$$

and the incumbent infeasible h and f -value are

$$(h_k^I, f_k^I) = \begin{cases} (\infty, \infty) & \text{if } I_k = \emptyset \\ (h(x), f(x)), \text{ for any } x \in I_k & \text{otherwise,} \end{cases} \quad (3)$$

Figure 1 illustrates the construction of the incumbent values at iteration k . The circles represent the images of all 13 trial points generated by the algorithm by the start of iteration k , i.e., the set V_k . The barrier on the constraints rejects all three trial points that map outside of the shaded area. The set of undominated infeasible trial points U_k is indicated by arrows. It contains four elements, three of them with a constraint violation function value less than h_k^{\max} . The one with the best objective function value is (h_k^I, f_k^I) . Notice that this new incumbent was not necessarily generated during iteration $k - 1$. It belongs to V_k , but could have been generated at any iteration prior to iteration k . The role of the parameter ρ appearing in the right part of the figure is detailed in Section 2.5.

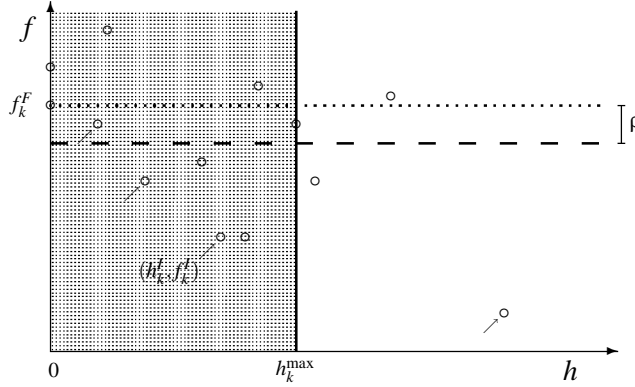


Figure 1: New incumbent values at iteration k .

An immediate result is that if $F_k \neq \emptyset$ at some iteration k , then $F_\ell \neq \emptyset$ at every iteration $\ell \geq k$. We propose in Section 2.4 an update rule for h_k^{\max} that ensures the analogue result for I_k .

Once the incumbents sets F_k and I_k and incumbent values $(0, f_k^F)$ and (h_k^I, f_k^I) are updated by the above definitions at the start of an iteration, the goal during the iteration is to change one of the incumbents. This occurs naturally when a trial point that dominates one of the incumbents is generated. When no such points are generated during an iteration, other measures must be taken. If the set V_{k+1} contains an infeasible point with a smaller constraint violation function value than h_k^I , then the barrier parameter is reduced to the largest value less than h_k^I . Otherwise, the algorithm will refine the exploration in the space of variables. The next section

details how the algorithm explores the space of variables. Section 2.4 describes the parameter update rules.

2.3 Description of an iteration : the SEARCH and POLL steps

In GPS, MADS-EB and the present algorithm, every trial point must be chosen on an underlying mesh defined on the space of variables whose fineness or coarseness is dictated by a nonnegative scalar called the *mesh size parameter* Δ_k^m .

Definition 2.4 *At iteration k , the current mesh is defined to be the following union:*

$$M_k = \bigcup_{x \in V_k} \{x + \Delta_k^m D z : z \in \mathbb{N}^{n_D}\} ,$$

where $D = GZ \in \mathbb{R}^{n \times n_D}$ is a positive spanning matrix¹, for some non-singular matrix $G \in \mathbb{R}^{n \times n}$ and integer matrix $Z \in \mathbb{Z}^{n \times n_D}$.

Note that D, G and Z are fixed matrices, independent of the iteration number k . Frequent choices for these matrices are $G = I_n$, the $n \times n$ identity matrix and $D = Z = [I_n \ -I_n]$. For convenience, the matrix D will often be treated as a set: $d \in D$ signifies that d is a column of D . The mesh structure allows a convergence analysis without requiring sufficient decrease conditions to accept new incumbent solutions.

There is great algorithmic flexibility in searching for new incumbent solutions. Each iteration is divided into two steps: the SEARCH and POLL steps. In the SEARCH step, any number of trial points may be generated, as long as these points belong to the mesh M_k , and that the strategy to identify them terminates in finite time.

The SEARCH strategy may, for example, be based on a heuristic exploration of the domain, or it may employ surrogate functions based on response surfaces, interpolatory models or simplified physics models. Surrogates are most often tailored for specific applications, see, e.g., [5, 7, 9, 19, 20, 22]. Let us simply denote by S_k the finite set of mesh points used in the SEARCH step at iteration k .

Unlike the freedom of the SEARCH step, the POLL step is more rigidly defined. It consists of a local exploration around incumbent solutions in F_k and I_k . The POLL step depends on another nonnegative scalar called the poll size parameter Δ_k^p . There is some flexibility in the choice of the poll size parameter. It must, however, be tied to the mesh size parameter in a way that satisfies:

$$\lim_{k \in K} \Delta_k^m = 0 \quad \text{if and only if} \quad \lim_{k \in K} \Delta_k^p = 0, \quad \text{for every infinite subset of indices } K. \quad (4)$$

¹ nonnegative linear combinations of the columns of D span \mathbb{R}^n , see [13].

For example, one may set $\Delta_k^p = \sqrt{\Delta_k^m}$ (as in [6]) so that the poll size parameter goes to zero slower than the mesh size parameter. The original idea of a frame is from [11, 23, 24], and it is more general than the version given below.

Definition 2.5 *At iteration k , $D_k(x)$ is said to be a set of frame directions around a frame center $x \in V_k$ if $D_k(x)$ is a finite set of directions in \mathbb{N}^n such that for each $d \in D_k(x)$,*

- *$d \neq 0$ can be written as a nonnegative integer combination of the directions in D , and $d = Du$ for some vector $u \in \mathbb{N}^D$ that may depend on the iteration number k and on x ;*
- *the distance from the frame center x to $x + \Delta_k^m d$ is bounded above by a multiple of the poll size parameter: $\Delta_k^m \|d\| \leq \Delta_k^p \max\{\|d'\| : d' \in D\}$.*

This last definition is the fundamental difference between GPS and MADS. In GPS, the directions in the set D_k are restricted to be chosen from the fixed set D . In MADS, the number of candidates directions in $D_k(\cdot)$ grows without bound as Δ_k^m goes to zero.

In MADS-EB, the POLL set was always constructed around a feasible point since any infeasible point was rejected by the barrier. In the present MADS-PB approach, we need to adapt the definition of the POLL set.

Definition 2.6 *At iteration k , the POLL set P_k is defined to be*

$$P_k = \begin{cases} P_k(x_k^F) & \text{for some } x_k^F \in F_k, \text{ if } I_k = \emptyset \\ P_k(x_k^I) & \text{for some } x_k^I \in I_k, \text{ if } F_k = \emptyset \\ P_k(x_k^F) \cup P_k(x_k^I) & \text{for some } x_k^F \in F_k \text{ and } x_k^I \in I_k, \text{ otherwise,} \end{cases}$$

where $P_k(x) = \{x + \Delta_k^m d : d \in D_k(x)\} \cap X \subset M_k$ is called a frame around x .

Figure 2 illustrates an example in which both a feasible x_k^F and an infeasible incumbent x_k^I solution exist. In the figure, the feasible region Ω is delimited by the nonlinear curves, and $X = \mathbb{R}^2$. The mesh is constructed using Δ_k^m and is represented by the intersection of all thin lines. The poll set is constructed by taking some mesh points inside the two regions delimited by the thick lines, based on $\Delta_k^p > \Delta_k^m$. In this example, the frame around the feasible incumbent is constructed using the positive spanning set of directions $D_k(x_k^F) = \{(-3, 4)^T, (4, 0)^T, (-1, -4)^T\}$ and the frame around the infeasible one is built using a single direction $D_k(x_k^I) = \{(3, -4)^T\}$. Therefore, the poll set P_k is the union of the frames $P_k(x_k^F) = \{p_1, p_2, p_3\}$ with $P_k(x_k^I) = \{p_4\}$. Implementable strategies of constructing the poll set P_k are presented in Section 2.5.

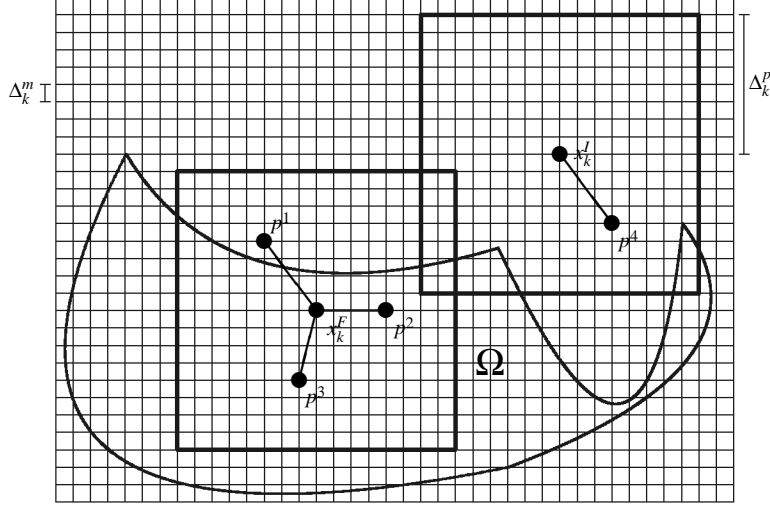


Figure 2: A MADS POLL set $P_k = P_k(x_k^F) \cup P_k(x_k^I) = \{p^1, p^2, p^3\} \cup \{p^4\}$.

2.4 Parameter update at the end of an iteration

After the POLL and SEARCH steps are completed, the algorithm has evaluated f and h at one or more trial points. At the end of iteration k , V_{k+1} contains all the trial points since the algorithm was initiated. The function values of the points in V_{k+1} govern the way that the mesh size Δ_{k+1}^m and the barrier h_{k+1}^{\max} parameters are updated.

The way this is done depends on the result of iteration k . There are three possibilities: the iteration may be *dominating*, *improving*, or *unsuccessful*. A dominating iteration generates a trial point that dominates an incumbent. An improving iteration is not dominating, but it improves the feasibility of the infeasible incumbents, and so it replaces the infeasible incumbent set. Unsuccessful iterations are neither dominating nor improving. These three types of iterations are detailed below and illustrated by having a point of V_{k+1} in the appropriate shaded area of Figure 3.

- Iteration k is said to be *dominating* whenever a trial point $y \in V_{k+1}$ that dominates an incumbent is generated, i.e., when

$$h(y) = 0 \text{ and } f(y) < f_k^F, \quad \text{or} \quad y \prec x \text{ for all } x \in I_k.$$

- Iteration k is said to be *improving* if it is not dominating, but if there is an

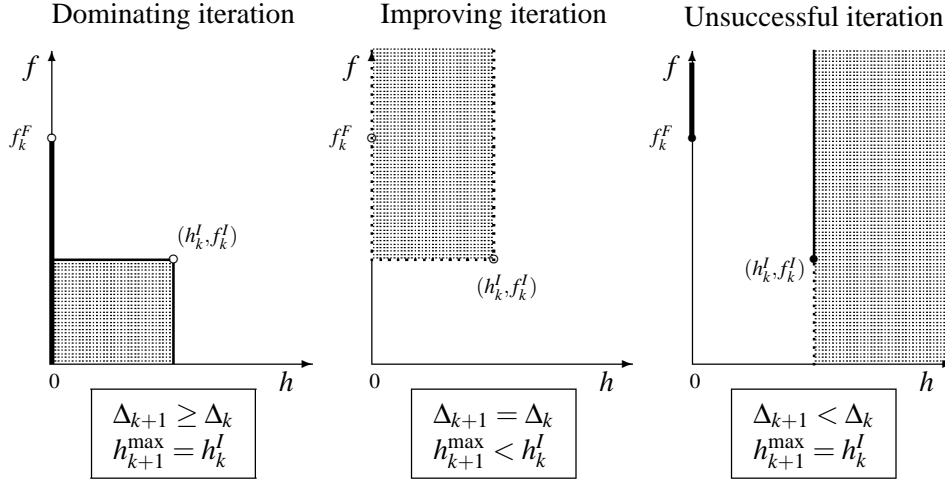


Figure 3: Mesh size and barrier parameter update rules.

infeasible solution $y \in V_{k+1}$ with a strictly smaller value of h , i.e., when

$$0 < h(y) < h_k^I \text{ and } f(y) > f_k^I.$$

- Iterations that are neither dominating nor improving are called *unsuccessful iterations*. This happens when every trial point $y \in V_{k+1}$ is such that:

$$h(y) = 0 \text{ and } f(y) \geq f_k^F, \text{ or } h(y) = h_k^I \text{ and } f(y) \geq f_k^I, \text{ or } h(y) > h_k^I.$$

To clarify these ideas, we refer to Figure 2 to illustrate various constructions. Assume that at iteration k the incumbent sets are $F_k = \{x_k^F\}$ and $I_k = \{x_k^I\}$ and that $V_{k+1} = \{x_k^F, x_k^I, p^1, p^2, p^3, p^4\}$, and that no other points have been generated so far. Let us consider the infeasible points. In the case where $0 < h(p^1) < h(p^4) < h(x_k^I)$ and $f(p^1) > f(p^4) > f(x_k^I)$, then the next infeasible incumbent set would be $I_{k+1} = \{p^4\}$, and iteration k would be improving because $h(p^4) < h(x_k^I)$ but $p^4 \notin x_k^I$. The iteration would have been dominating if instead $f(p^4) \leq f(x_k^I)$, or of course, if either $f(p^2)$ or $f(p^3)$ were strictly less than $f(x_k^F)$.

The classification of the iterations dictates the way that the mesh and poll size parameters are updated from one iteration to another. More precisely, given a fixed rational number $\tau > 1$, and two integers $w^- \leq -1$ and $w^+ \geq 0$, the mesh size parameter is updated as follow:

$$\begin{aligned} \Delta_{k+1}^m &= \tau^{w_k} \Delta_k^m \\ \text{for some } w_k &\in \begin{cases} \{0, 1, \dots, w^+\}, & \text{if iteration } k \text{ is dominating} \\ \{0\}, & \text{if iteration } k \text{ is improving} \\ \{w^-, w^- + 1, \dots, -1\}, & \text{if iteration } k \text{ is unsuccessful.} \end{cases} \end{aligned} \quad (5)$$

Typical choices [6] are $\tau = 4$, and $w^+ = -w^- = 1$.

The update rules for the barrier parameter are:

$$h_{k+1}^{\max} = \begin{cases} \max_{y \in V_{k+1}} \{h(y) : h(y) < h_k^I\} & \text{if iteration } k \text{ is improving} \\ h_k^I & \text{otherwise.} \end{cases} \quad (6)$$

There are three sub-cases when an iteration is dominating. First, it is possible that a trial point that improves the feasible incumbent is generated, and that no trial point dominates the infeasible one. In that case, the consequence of the update rules is that $(h_{k+1}^I, f_{k+1}^I) = (h_k^I, f_k^I)$ but $f_{k+1}^F < f_k^F$. Second, it is possible that a dominant trial point t with $h(t) = h_k^I > 0$ is generated. In that case, the consequence of the update rules is that $h_{k+1}^I = h_k^I$ but $f_{k+1}^I < f_k^I$. The last possibility is that $h_{k+1}^I < h_k^I$ and $f_{k+1}^I \leq f_k^I$.

Consequences of the barrier parameter update rule (6) are that h_k^{\max} is non-increasing with respect to k , and if $I_k \neq \emptyset$ then $I_\ell \neq \emptyset$ at every iteration $\ell \geq k$.

Figure 4 summarizes our new MADS-PB algorithm. Notice that if the initial set V_0 contains a feasible point, and if $h_0^{\max} = 0$, then the algorithm reduces to MADS-EB [6].

This high-level description of the algorithm contains in the initialization phase an optional frame center trigger ρ , which is discussed in the next section.

Remark: In practice, some users may wish to allow soft constraints in problem (1). Then one can apply the method described in this paper to the optimization problem $\min_x \{f(x) : h(x) \leq h^{\min}\}$ where $h^{\min} > 0$ is a user-selected threshold on the function h under which any trial point is considered feasible. This may be done by redefining dominating and improving iterations by replacing $h(x) = 0$ by $h(x) \leq h^{\min}$ and $h(x) > 0$ by $h(x) > h^{\min}$.

2.5 A frame center selection rule

We refer the reader to Section 4.1 of [6] for an explicit way to construct the positive basis D_k used to construct a frame. This construction depends only on the mesh and poll size parameters Δ_k^m and Δ_k^p , and it satisfies the requirements of Definition 2.6. The set D_k is constructed by first generating a direction b_k , and then completing it to a positive basis B_k . This is done in such a way that $\cup_{k=1}^{\infty} \left\{ \frac{b_k}{\|b_k\|} \right\}$ is dense in the unit sphere with probability 1 (see Theorem 4.3 in [6] and [3]). MADS with this choice of D_k is called LTMADS.

At each iteration, there are either one or two frame centers. When $F_k = \emptyset$ or $I_k = \emptyset$, then there is only one frame center, call it x^1 , and it is arbitrarily chosen in whichever of F_k or I_k is nonempty (by A1). That frame center is then called the *primary* frame center. The POLL set will simply be $P_k = P_k(x^1) = \{x^1 + \Delta_k^m d : d \in$

A MADS-PB ALGORITHM FOR CONSTRAINED OPTIMIZATION

- **INITIALIZATION** (given a set of initial points V_0 that satisfies assumption A1):
 - Define the mesh matrices G and Z as in Definition 2.4, and mesh parameters τ , w^- and w^+ as in equation (5), and $0 < \Delta_0^m \leq \Delta_0^p$.
 - (Optional) Define the frame trigger $\rho > 0$ as in Definition 2.7.
 - Set the iteration counter $k \leftarrow 0$.
- **INCUMBENT DEFINITION:** Define the incumbent sets F_k and I_k as in Definitions 2.1 and 2.3, and incumbent values f_k^F and (h_k^I, f_k^I) as in Equations (2) and (3).
- **SEARCH:** Evaluate h and f on a finite set S_k of trial points in X on the current mesh M_k as in Definition 2.4. This step is optional, i.e., $S_k = \emptyset$ is allowed. If an improving or dominating point is found in S_k , then the SEARCH may terminate, skip the next - POLL - step, and go directly to the PARAMETER UPDATE step. Otherwise the algorithm must go to the POLL step.
- **POLL:** Evaluate h and f on the poll set P_k of trial points in X on the current mesh M_k as in Definition 2.6 (optional: use Definition 2.7). This step may terminate opportunistically.
- **PARAMETER UPDATE:**
 - Define V_{k+1} to be the union of V_k with the sets of points in X visited in the SEARCH and POLL steps.
 - Classify the iteration as being dominating, improving, or unsuccessful, and update Δ_{k+1}^m according to equation (5), and Δ_{k+1}^p according to (4).
 - Update the barrier parameter h_{k+1}^{\max} as in equation (6).
 - Increase $k \leftarrow k + 1$ and go back to the INCUMBENT DEFINITION step.

Figure 4: A MADS-PB algorithm for constrained optimization

$D_k(x^1)\}$ where $D_k(x^1)$ is the positive basis constructed in [6] (it is denoted by D_k in that reference).

In the event that both F_k or I_k are nonempty, a *secondary* frame center x^2 will be chosen as well as a *primary* poll center. The next definition provides a practical rule to choose which are the primary and secondary frame centers. It is based on another (optional) user supplied parameter $\rho > 0$ called the *frame center trigger*.

Definition 2.7 (Frame center selection rule) *Let $\rho > 0$ be a given constant and suppose that $F_k \neq \emptyset$ and $I_k \neq \emptyset$. If $f_k^F - \rho > f_k^I$, then the primary poll center x_k^1 is chosen in I_k and the secondary poll center x_k^2 is chosen in F_k . Otherwise the primary poll center x_k^1 is chosen in F_k and the secondary poll center x_k^2 is chosen in I_k .*

The dashed horizontal line in Figure 1 just below the feasible incumbent represents $f = f_k^F - \rho$. On that particular example, the frame center selection rule would choose the primary poll center in the infeasible incumbent set I_k . There are both theoretical and computational reasons for this approach. The following corollary to a later result shows the theoretical value of the frame center selection rule. In this corollary, we use the notion of a refining sequence familiar to readers of our earlier papers. We will define it later, but for now suffice it to say that refining sequences are the ones for which our strongest convergence results apply, and refined points are our solutions.

Corollary 2.8 *Suppose that assumptions A1, A2 and A3 hold, then there can not be a refining subsequence of infeasible primary poll centers satisfying the poll trigger condition that converges to a feasible refined point.*

Proof. The proof is immediate from Theorem 3.12, since the feasible points guaranteed by that theorem would negate the poll trigger condition. ■

If the infeasible incumbent is below $f_k^F - \rho$, then we might hope that by emphasizing it as the primary poll center we can reach a better part of the feasible region than the one containing the feasible incumbent.²

3 Convergence analysis

The MADS-PB algorithm can be applied to any nonlinear problem of the form (1) provided that assumption A1 is satisfied. There are two possible behaviors for the iterates produced by the algorithm. One possibility is that the iterates go unbounded, in which case no necessary optimality conditions may be guaranteed. We repeat the standard assumption from Section 2.1 that the iterates remain bounded.

A2 : All trial points considered by the algorithm lie in a bounded set.

This assumption may be reformulated in our notation as follows: There exists some bounded set in \mathbb{R}^n containing V_k for every k . By its definition, V_k does not contain any points that violate any of the closed constraints. Thus, it is easy to ensure A2 is satisfied by having membership in a bounded set as a closed constraint. Indeed, engineering problem statements usually have bounds on all the optimization variables.

²We are indebted to Dr. Paul Frank of the Boeing Company for suggesting the utility of this strategy in the context of the GPS-Filter algorithm given in [5].

Combining assumption A2 with the mesh structure was shown in [6] to be enough to ensure that $\liminf_k \Delta_k^m = 0$. Our main interest will be in the subsequence of frame centers for which the corresponding mesh size parameters converge to zero.

Define $\mathbb{U} \subseteq \mathbb{N}$ to be the subset of iteration indices corresponding to unsuccessful iterations. The POLL step generates trial points around either or both feasible and infeasible incumbents. If $k \in \mathbb{U}$, and polling was done around $x_k^F \in F_k$, then x_k^F is called a *feasible minimal frame center*. If $k \in \mathbb{U}$, and polling was done around $x_k^I \in I_k$, then x_k^I is called an *infeasible minimal frame center*. We need to study both type of frame centers separately, but notice that if $k \in \mathbb{U}$ then the iteration necessarily has a minimal frame centers of both types – unless either incumbent set is empty.

Definition 3.1 *A subsequence of the MADS-PB minimal frame centers, $\{x_k\}_{k \in K}$ for some subset of indices $K \subseteq \mathbb{U}$, is said to be a refining subsequence if $\{\Delta_k^p\}_{k \in K}$ converges to zero.*

The limit \hat{x} of a convergent refining subsequence is called a refined point. If $\lim_{k \in L} \frac{d_k}{\|d_k\|}$ exists for some subset $L \subseteq K$ with poll direction $d_k \in D_k(x_k)$, and if $x_k + \Delta_k^m d_k \in X$ for infinitely many $k \in L$, then this limit is said to be a refining direction for \hat{x} .

The analysis relies on the following definitions. The Clarke generalized derivative of f at $\hat{x} \in \Omega$ in the direction $v \in \mathbb{R}^n$ is defined as

$$f^\circ(\hat{x}; v) = \limsup_{\substack{y \rightarrow \hat{x}, y \in X \\ t \downarrow 0, y + tv \in X}} \frac{f(y + tv) - f(y)}{t}.$$

This definition from Jahn [17] generalizes the original one by Clarke [10] to the case where f is not defined outside X . Similarly, we say that a function is locally Lipschitz if it is Lipschitz with a finite constant in some nonempty neighborhood intersected with X .

Our convergence results involve different types of tangent cones. As in the MADS analysis [6], the most important one for the present context is the hypertangent cone [25] to Ω at \hat{x} :

$$T_\Omega^H(\hat{x}) = \{v \in \mathbb{R}^n : \exists \varepsilon > 0 \text{ such that } y + tw \in \Omega \\ \text{for all } y \in \Omega \cap B_\varepsilon(\hat{x}), w \in B_\varepsilon(v) \text{ and } 0 < t < \varepsilon\}.$$

The analysis is divided into three cases. First, Section 3.1 considers the case where the algorithm generates a convergent feasible refining subsequence. We give

conditions under which the Clarke derivative of f is non-negative in the hypertangent cone to Ω at the feasible refined point. Second, we analyze the function h in Section 3.2. We give conditions under which the Clarke derivative of h is non-negative in the hypertangent cone to the closed constraints X at a refined point. Finally in Sections 3.3 and 3.4, we look at the case where the algorithm generates an infeasible refining subsequence converging to a point on the boundary of Ω . We propose conditions in the form of an external constraint qualification under which the Clarke derivative of f is non-negative in the hypertangent cone to Ω at the refined point. Thus, since the Clarke tangent cone is the closure of the hypertangent cone, when the latter is nonempty, all three cases result in a proof of Clarke stationarity. We finally briefly summarize our results in section 3.5.

3.1 A feasible refining subsequence: results for f

The analysis presented in this subsection is similar to that of [6] where all constraints are treated through the barrier. The following lemma from elementary analysis will be useful.

Lemma 3.2 *If $\{a_k\}$ is a bounded real sequence and $\{b_k\}$ is a convergent real sequence, then $\limsup_k(a_k + b_k) = \limsup_k a_k + \lim_k b_k$.*

Theorem 3.3 *Suppose that assumptions A1 and A2 hold, and that the algorithm generates a refining subsequence $\{x_k^F\}_{k \in K}$ with $x_k^F \in F_k$ converging to a refined point \hat{x}^F in Ω , near which f is Lipschitz. If $v \in T_\Omega^H(\hat{x}^F)$ is a refining direction for \hat{x}^F , then $f^\circ(\hat{x}^F; v) \geq 0$.*

Proof. Let $\{x_k^F\}_{k \in K}$ with $x_k^F \in F_k$ be a feasible refining subsequence converging to $\hat{x}^F \in \Omega$ and $v = \lim_{k \in L} \frac{d_k}{\|d_k\|} \in T_\Omega^H(\hat{x}^F)$ be a refining direction for \hat{x}^F with $d_k \in D_k(x_k^F)$ for every $k \in L$. For each $k \in L$, define

$$t_k = \Delta_k^m \|d_k\| \rightarrow 0 \quad \text{and} \quad y_k = x_k^F + t_k \left(\frac{d_k}{\|d_k\|} - v \right) \rightarrow \hat{x}^F \quad (7)$$

(the fact that $t_k \rightarrow 0$ follows from the last bullet of Definition 2.5). Since f is Lipschitz with constant $\lambda > 0$ near \hat{x}^F , it follows that

$$\left| \frac{f(x_k^F) - f(y_k)}{t_k} \right| \leq \lambda \left\| \frac{d_k}{\|d_k\|} - v \right\|,$$

which then converges to 0. This will be our sequence $\{b_k\}$ of Lemma 3.2 in going from the first to the second line below. Adding and subtracting $f(x_k^F)$ to the

numerator of the definition of the Clarke derivative, one gets

$$\begin{aligned}
f^\circ(\hat{x}^F; v) &\geq \limsup_{k \in L} \frac{f(y_k + t_k v) - f(x_k^F) + f(x_k^F) - f(y_k)}{t_k} \\
&= \limsup_{k \in L} \frac{f(y_k + t_k v) - f(x_k^F)}{t_k} + \lim_{k \in L} \frac{f(x_k^F) - f(y_k)}{t_k} \\
\text{using eq. (7)} &= \limsup_{k \in L} \frac{f(x_k^F + \Delta_k^m d_k) - f(x_k^F)}{t_k} + \lim_{k \in L} \frac{f(x_k^F) - f(y_k)}{t_k} \geq 0
\end{aligned}$$

For sufficiently large $k \in L$, $x_k^F + \Delta_k^m d_k \in \Omega$ since v is an hypertangent direction. Therefore, the last inequality follows from the fact that $f(x_k^F + \Delta_k^m d_k)$ was evaluated and compared by the algorithm to $f(x_k^F)$, but x_k^F is a feasible minimal frame center. ■

The case where the set of refining directions is dense in a nonempty hypertangent cone to Ω ensures Clarke stationarity:

Corollary 3.4 *Suppose that assumptions A1 and A2 hold, and the algorithm generates a refining subsequence $\{x_k^F\}_{k \in K}$ with $x_k^F \in F_k$ converging to a feasible refined point \hat{x}^F in Ω , near which f is Lipschitz. If the set of refining directions for \hat{x}^F is dense in $T_\Omega^H(\hat{x}^F) \neq \emptyset$, then \hat{x}^F is a Clarke stationary point for (1).*

Proof. The assumptions ensure that $f^\circ(\hat{x}^F; v) \geq 0$ for a set of directions v which is dense in the closure of $T_\Omega^H(\hat{x}^F)$. Furthermore, the closure of the hypertangent cone coincides with the Clarke tangent cone wherever the hypertangent cone is nonempty [25]. ■

3.2 A convergent infeasible refining subsequence: results for h

Before considering the f values for an infeasible refining sequence, we examine the constraint violation function h at limits of refining subsequences. There are two possibilities for the value at a refined point \hat{x}^I . One possibility is that $h(\hat{x}^I) = 0$. This means that there is a nonempty feasible region and that the algorithm produced a global minimizer of h over the domain X defined by the closed constraints. Otherwise, \hat{x}^I satisfies some necessary conditions to be a local minimizer of h .

The issue of local versus global minimizer is not the main point here. After all, in analyzing SQP iterations, one generally makes strong assumptions like linear independence of constraint gradients, which ensures that any local minimizer of h is a global minimizer. Since we do not assume continuous differentiability, we will not make that specific assumption to ensure there are no local minimizers of h over

X . The real point of this section is to show what happens when there is an empty feasible region. In either case, the following result shows that we do find a Clarke stationary point for h .

Theorem 3.5 *Let assumptions A1 and A2 hold, and assume that the algorithm generates a refining subsequence $\{x_k^I\}_{k \in K}$ with $x_k^I \in I_k$ converging to a refined point \hat{x}^I in X near which h is Lipschitz. If $v \in T_X^H(\hat{x}^I)$ is a refining direction for \hat{x}^I , then $h^\circ(\hat{x}^I; v) \geq 0$.*

Proof. Let $\{x_k^I\}_{k \in K}$ with $x_k^I \in I_k$ be an infeasible refining subsequence converging to $\hat{x}^I \in X$ and $v = \lim_{k \in L} \frac{d_k}{\|d_k\|} \in T_X^H(\hat{x}^I)$ be a refining direction for \hat{x}^I , with $d_k \in D_k(x_k^I)$ for every $k \in L$. If $h(\hat{x}^I) = 0$, then the result is trivial. Otherwise, the remainder of the proof is identical to that of Theorem 3.3, with h and X playing the roles of f and Ω , respectively. ■

The next corollary's proof is essentially identical to that of Corollary 3.4.

Corollary 3.6 *Suppose that assumptions A1 and A2 hold, and the algorithm generates an infeasible refining subsequence $\{x_k^I\}_{k \in K}$ with $x_k^I \in I_k$ converging to a refined point \hat{x}^I in X , near which h is Lipschitz. If the set of refining directions for \hat{x}^I is dense in $T_X^H(\hat{x}^I) \neq \emptyset$, then \hat{x}^I is a Clarke stationary point for*

$$\min_{x \in X} h(x). \quad (8)$$

Proof. The assumptions ensure that $h^\circ(\hat{x}^I; v) \geq 0$ for a set of directions v which is dense in the closure of $T_X^H(\hat{x}^I)$. Furthermore, the closure of the hypertangent cone coincides with the Clarke tangent cone, wherever the hypertangent cone is nonempty [25]. ■

3.3 An external constraint qualification

The remaining case that needs to be analyzed further is when MADS generates an infeasible refining subsequence converging to a feasible point \hat{x} . Ideally, we would like to show that the Clarke derivative of f is nonnegative at \hat{x} for all hypertangent vectors. The following example shows that without additional assumptions on the constraints that there might be a descent direction $v \in T_\Omega^H(\hat{x})$ for f . After this motivating example, we will supply an adequate additional assumption and relate it to a common constraint qualification for SQP.

Example 3.7 Consider the optimization problem in \mathbb{R}^1 :

$$\begin{aligned} \min_{x \in \mathbb{R}} \quad & f(x) = x \\ \text{s.t.} \quad & c(x) \leq 0, \end{aligned} \tag{9}$$

where

$$c(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 4^{-\ell(x)} p^2(v(x)) & \text{if } x > 0, \end{cases}$$

with $p(x) = -1260 + 4440x - 6015x^2 + 3935x^3 - 1245x^4 + 153x^5$ and where $\ell(x) = -\lfloor \log_2(\frac{x}{3}) \rfloor$ and $v(x) = \frac{2^{\ell(x)}}{3}x$. These particular functions were chosen so that

$$\begin{aligned} p(1) &= 8, & p(\frac{4}{3}) &= 4, & p(2) &= 16 \\ p'(1) &= 0, & p'(\frac{4}{3}) &= 0, & p'(2) &= 0. \end{aligned}$$

and for any $x > 0$, the values $\ell(x) \in \mathbb{Z}$ and $v(x) \in]1, 2]$ are the unique integer and scalar such that $x = 3 \times 2^{-\ell(x)} v(x)$.

Figure 5 illustrates the function c for some $\ell \in \mathbb{Z}$. One can verify that c is differentiable on \mathbb{R} , that its derivative is Lipschitz continuous, and that for any $\ell \in \mathbb{Z}$, $x = 4 \times 2^{-\ell}$ is the unique minimizer of c on the interval $[3 \times 2^{-\ell}, 5 \times 2^{-\ell}]$. The feasible region for this problem is simply $\Omega =]-\infty, 0]$ with $X = \mathbb{R}$.

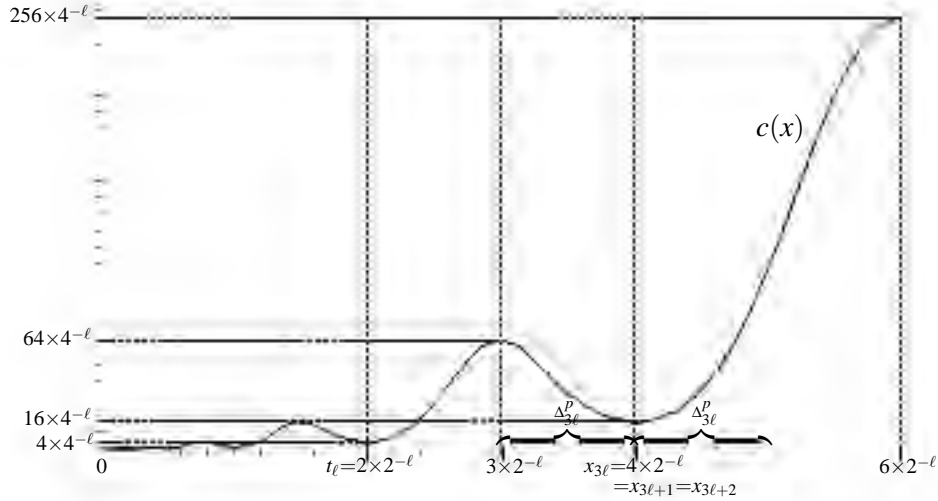


Figure 5: A differentiable constraint function $c(x)$.

The MADS instance considered here is tailored to produce a bad limit point. At each three iterations, starting at iteration 2, a SEARCH step is conducted. More

precisely, at iteration $k = 3\ell + 2$ for some $\ell \in \mathbb{N}$, the SEARCH generates the trial point $t_\ell = x_k - 2^{\ell+5}\Delta_k^m$ with $\Delta_k^m = (\Delta_k^p)^2$. At other iterations the SEARCH step is empty. The poll size parameter satisfies $\Delta_k^p = \sqrt{\Delta_k^m}$ is doubled at dominating iterations and halved at unsuccessful ones (i.e., $\tau = 4$, $w_k = w^+ = 1$ or $w_k = w^- = -1$ in equation (5)).

MADS is initiated at $x_0 = 4$ with $\Delta_0^m = \Delta_0^p = 1$ and $D = [1, -1]$. The initial point x_0 is the unique minimizer of c in the interval $[3, 5]$ and therefore the POLL step at iteration 0 fails to generate a new incumbent. Iteration 1 starts at $x_1 = x_0$ with an even smaller poll size parameter $\Delta_1^p = \frac{1}{2}$, thus iteration 1 is also unsuccessful. Iteration 2 starts at $x_2 = x_1$ with mesh and poll size parameters $\Delta_2^m = \frac{1}{16}$ and $\Delta_2^p = \frac{1}{4}$. The SEARCH step at iteration $k = 3\ell + 2$ with $\ell = 0$ generates the trial point $t_\ell = x_2 - 2^5\Delta_2^m = 4 - \frac{32}{16} = 2$, which is more feasible than x_2 and has a better objective function value f . Iteration 2 is thus dominating, and iteration 3 starts at $x_3 = 2$ with parameters $\Delta_3^m = \frac{1}{4}$ and $\Delta_3^p = \frac{1}{2}$.

For $\ell \in \mathbb{N}$, we will refer to iterations $3\ell, 3\ell + 1$ and $3\ell + 2$ as the ℓ^{th} cycle.

Proposition 3.8 For any integer $\ell \geq 0$, the iterates of cycle ℓ generated by the above instance of MADS satisfy $x_{3\ell} = x_{3\ell+1} = x_{3\ell+2} = 4 \times 2^{-\ell}$ and the poll size parameters satisfy $\Delta_{3\ell}^p = 2^{-\ell}$, $\Delta_{3\ell+1}^p = 2^{-\ell-1}$ and $\Delta_{3\ell+2}^p = 2^{-\ell-2}$.

Proof. The proof is done by induction on ℓ . We already verified in above that the result holds for cycle $\ell = 0$.

Suppose that iteration $k = 3\ell$ is initiated with $x_{3\ell} = 4 \times 2^{-\ell}$ and $\Delta_{3\ell}^p = 2^{-\ell}$. At that point $f(x_{3\ell}) = 2^{-\ell+2}$ and $c(x_{3\ell}) = 16 \times 4^{-\ell}$. As mentioned above, $x_{3\ell}$ is the unique minimizer of c in the interval $[x_k - \Delta_k^p, x_k + \Delta_k^p] = [3 \times 2^{-\ell}, 5 \times 2^{-\ell}]$ and therefore the POLL step at iteration 3ℓ fails to generate a new incumbent. Iteration $3\ell + 1$ starts at $x_{3\ell+1} = x_{3\ell}$ with even smaller poll size parameter $\Delta_{3\ell+1}^p = 2^{-\ell-1}$, thus iteration $3\ell + 1$ is also unsuccessful. Iteration $3\ell + 2$ starts at $x_{3\ell+2} = x_{3\ell+1}$ with poll size parameter $\Delta_{3\ell+2}^p = 2^{-\ell-2}$. The SEARCH step at iteration $k = 3\ell + 2$ generates the trial point

$$\begin{aligned} t_\ell &= x_{3\ell+2} - 2^{\ell+5}\Delta_{3\ell+2}^m &= x_{3\ell+2} - 2^{\ell+5}(\Delta_{3\ell+2}^p)^2 \\ &= 4 \times 2^{-\ell} - 2^{\ell+5} \times 2^{-2\ell-4} &= 2 \times 2^{-\ell}, \end{aligned}$$

where $c(t_\ell) = 4 \times 4^{-\ell} < c(x_{3\ell+2}) = 16 \times 4^{-\ell}$ and $f(t_\ell) < f(x_{3\ell+2})$. Therefore iteration $3\ell + 2$ is dominating, and iteration $3(\ell + 1)$ starts at $x_{3(\ell+1)} = t_\ell = 4 \times 2^{-\ell-1}$ with $\Delta_{3(\ell+1)}^p = 2^{-\ell-1}$. \blacksquare

The previous proposition shows that the entire sequence of MADS frame centers are infeasible and converge to $\hat{x} = 0$, a feasible point on the boundary of

$\Omega =] - \infty, 0]$. The hypertangent cone to Ω at \hat{x} is nonempty and contains descent directions for the objective function $f(x) = x$. In fact, every hypertangent direction is a descent direction for f since $f^\circ(\hat{x}; v) = f'(\hat{x}; v) < 0$ for every $v \in T_\Omega^H(\hat{x})$.

The above example shows that in order to derive stronger convergence results, one must make an additional assumption. We propose the following constraint qualification, where $B_\varepsilon(\cdot)$ denotes a ball of radius ε .

A3 : For every hypertangent direction $v \in T_\Omega^H(\hat{x}) \neq \emptyset$, there exists an $\varepsilon > 0$ for which $h^\circ(x; v) < 0$ for all $x \in \{x \in X \cap B_\varepsilon(\hat{x}) : h(x) > 0\}$.

Example 3.7 fails to satisfy assumption A3 since $h'(7 \times 2^{-\ell}; -1) > 0$ for any $\ell \in \mathbb{N}$. We discuss this assumption for the remaining of this subsection. In [6], we studied the case where the MADS algorithm treats all constraints by the barrier approach, i.e., $X = \Omega$. We assumed the existence of an hypertangent vector at a putative solution \hat{x} as a constraint qualification. We showed in the continuously differentiable case that this is equivalent to the Gould and Tolle or Mangasarian and Fromovitz constraint qualification with no equality constraints, see [16, 8]. We restate that result here because we will need it in our investigation of A3.

Theorem 3.9 (from [6]) *Let $C : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuously differentiable at a point $\hat{x} \in \Lambda = \{x \in \mathbb{R}^n : C(x) \leq 0\}$, and let $\mathcal{A}(\hat{x}) = \{i \in \{1, 2, \dots, m\} : c_i(\hat{x}) = 0\}$ be the active set at \hat{x} . If $v \in \mathbb{R}^n$ is a hypertangent vector to Λ at \hat{x} then $\nabla c_i(\hat{x})^T v < 0$ for each $i \in \mathcal{A}(\hat{x})$ such that $\nabla c_i(\hat{x}) \neq 0$. Furthermore, if $\nabla c_i(\hat{x})^T v < 0$ for each $i \in \mathcal{A}(\hat{x})$, then $v \in \mathbb{R}^n$ is a hypertangent vector to Λ at \hat{x} .*

As we saw in Section 3.1, the existence of a hypertangent vector was sufficient for us to prove strong results for a refining sequence of feasible iterates, and the previous theorem relates this assumption to assumptions on $C(x)$ that are weaker than those usually assumed for SQP.

The following theorem relates the constraint qualification A3 to assumptions on $C(x)$ under continuous differentiability. These assumptions are weaker than assuming that $\nabla c_i(\hat{x}) \neq 0$ for all $i \in \mathcal{A}(\hat{x})$, which is in turn weaker than a common SQP assumption $m \leq n$ and $C'(\hat{x})$ has full rank.

Theorem 3.10 *Let $C : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuously differentiable at a point $\hat{x} \in \Omega = \{x \in X : C(x) \leq 0\}$, and assume that $T_\Omega^H(\hat{x}) \neq \emptyset$. Assume that there is an $\varepsilon > 0$ for which*

$$\forall x \in X \cap B_\varepsilon(\hat{x}) \text{ with } C(x) \not\leq 0, \exists i \in \mathcal{A}(\hat{x}) \text{ for which } c_i(x) > 0 \text{ and } \nabla c_i(\hat{x}) \neq 0.$$

Then Assumption A3 holds.

Proof. Let $v \in T_{\Omega}^H(\hat{x})$, Then by Theorem 3.9, $\nabla c_i(\hat{x})^T v < 0$ for each $i \in \mathcal{A}(\hat{x})$ with $\nabla c_i(\hat{x}) \neq 0$. By continuity, $\exists \varepsilon > 0$ such that for every $x \in X \cap B_{\varepsilon}(\hat{x})$, we still have that $\nabla c_i(x)^T v < 0$ for each $i \in \mathcal{A}(\hat{x})$ with $\nabla c_i(\hat{x}) \neq 0$.

By taking ε even smaller if necessary, we can ensure that for $i \notin \mathcal{A}(\hat{x})$, $c_i(x) < 0$ for $x \in X \cap B_{\varepsilon}(\hat{x})$. Now let x be such a point for which $h(x) > 0$, which implies $C(x) \not\leq 0$. Then, by hypothesis, there must be at least one $i \in \mathcal{A}(\hat{x})$ for which $c_i(x) > 0$ and $\nabla c_i(\hat{x}) \neq 0$. Thus $\nabla c_i(\hat{x})^T v < 0$, and by the choice of ε , $\nabla c_i(x)^T v < 0$ as well.

Since $h(x) = \sum_1^m (\max(c_j(x), 0))^2$, we have from [14] that

$$h^{\circ}(x; v) = \nabla h(x)^T v = 2 \cdot v^T C'(x)^T W(x) C(x), \quad (10)$$

where $W(x)$ is a diagonal matrix with zeros in the i th position when $c_i(x) \leq 0$ and ones when $c_i(x) > 0$. Thus, (10) is nonpositive since it is the inner product of a nonpositive vector $2 \cdot v^T C'(x)^T$ and a nonnegative vector $W(x) C(x)$. Furthermore, it is nonzero because for at least one $i \in \mathcal{A}(\hat{x})$ the i th components of the two vectors are nonzero. Thus $h^{\circ}(x; v) < 0$. \blacksquare

3.4 A convergent infeasible refining subsequence: result on f and h

We show here that under Assumption A3, the algorithm generates infinitely many feasible points. Consequently, there exists a feasible refining subsequence, and thus the convergence results of Section 3.1 may be applied to that feasible subsequence. We first need the following lemma.

Lemma 3.11 *Let $v \in T_X^H(\hat{x}) \cap T_{\Omega}^H(\hat{x})$, be such that assumption A3 is satisfied. Then there exists a scalar $\delta > 0$ such that if $y \in X \cap B_{\delta}(\hat{x})$, and $h(y) > 0$ and $w \in B_{\delta}(v)$ and $0 < t < \delta$, then $h(y + tw) < h(y)$.*

Proof. Let $v \in T_X^H(\hat{x}) \cap T_{\Omega}^H(\hat{x})$, and $\varepsilon > 0$ be small enough so that assumption A3 is satisfied. Suppose that the result is false, i.e., that for any $\delta > 0$, there exists some $y_{\delta} \in X \cap B_{\delta}(\hat{x})$, with $h(y_{\delta}) > 0$ and some $w_{\delta} \in B_{\delta}(v)$ and some $0 < t_{\delta} < \delta$ such that $h(y_{\delta} + t_{\delta} w_{\delta}) \geq h(y_{\delta})$.

Then, if δ is sufficiently small, then $w_{\delta} \in T_X^H(\hat{x}) \cap T_{\Omega}^H(\hat{x})$, and the entire line segment $I = [y_{\delta}, y_{\delta} + t_{\delta} w_{\delta}]$ is contained in $X \cap B_{\delta}(\hat{x})$ (by definition of the hyper-tangent cone to X). Assumption A3 ensures that h is Lipschitz continuous on I . Theorem 2.3.7 of Clarke [10] ensures that there is some $u \in I$ and some ζ in the generalized gradient $\partial h(u)$ such that $t_{\delta} w_{\delta}^T \zeta = h(y_{\delta} + t_{\delta} w_{\delta}) - h(y_{\delta}) \geq 0$. Therefore, by definition of the generalized gradient, $h^{\circ}(u; w_{\delta}) \geq w_{\delta}^T \zeta \geq 0$. This contradicts

assumption A3. ■

The previous lemma provides sufficient conditions under which h decreases in some direction w . It will be used in the proof of the next result by substituting $y = x_k$, $t = \Delta_k^m \|d_k\|$ and $w = \frac{d_k}{\|d_k\|}$.

Theorem 3.12 *Let assumptions A1, A2 and A3 hold, and assume that the algorithm generates an infeasible refining subsequence $\{x_k^I\}_{k \in K}$ converging to a feasible refined point \hat{x} in Ω with refining direction $v \in T_X^H(\hat{x}) \cap T_\Omega^H(\hat{x})$. Then, there exists a feasible refining subsequence for which the conclusions of Theorem 3.3 and Corollary 3.4 hold:*

If $v \in T_\Omega^H(\hat{x}^F)$ is a refining direction for \hat{x}^F , then $f^\circ(\hat{x}^F; v) \geq 0$.

If the set of refining directions for \hat{x}^F is dense in $T_\Omega^H(\hat{x}^F) \neq \emptyset$, then \hat{x}^F is a Clarke stationary point for (1).

Proof. Let $\hat{x} \in \Omega$ be the feasible limit of an infeasible refining subsequence $\{x_k^I\}_{k \in K}$ with refining direction $v \in T_X^H(\hat{x}) \cap T_\Omega^H(\hat{x})$. But when $k \in K$ is sufficiently large, Assumption A3 and Lemma 3.11 ensures that $x_k^I + \Delta_k^m d_k \in X$ since $v \in T_X^H(\hat{x})$, and that $h(x_k^I + \Delta_k^m d_k) < h(x_k^I)$ for some polling direction $d_k \in D_k$.

If $h(x_k^I + \Delta_k^m d_k) > 0$ then iteration k would be either dominating or improving, as a new infeasible incumbent would be generated. Therefore, for all $k \in K \subseteq U$ sufficiently large, $h(x_k^I + \Delta_k^m d_k) = 0$ for some frame direction $d_k \in D_k$.

We have shown that infinitely many feasible points near \hat{x} are generated by the algorithm. Thus, there exists a feasible refining subsequence for which Theorem 3.3 and Corollary 3.4 hold. ■

To illustrate this last theorem, consider the simple example of minimizing the convex function $f(x) = (x + \pi)^2$ subject to a single linear constraint $x \leq 0$ with infeasible starting point $x_0 = 1$. The sequence of feasible frame centers x_k^F of any MADS-PB instance will converge to the strictly feasible global optimizer $\hat{x}^F = -\pi$. The entire sequence of infeasible frame centers x_k^I converges to the feasible solution $\hat{x}^I = 0$ on the boundary of Ω . Polling around the infeasible frame centers will generate some feasible points close to $\hat{x}^I = 0$, but these feasible points will usually not improve the current feasible incumbent (which will be located near the global minimizer $\hat{x}^F = -\pi$). However, there are some feasible descent directions for f at \hat{x}^I . The point of this last observation is that Theorem 3.3 and Corollary 3.4 may be applied to the limit of feasible frame centers $\hat{x}^F = -\pi$, and not to $\hat{x}^I = 0$.

3.5 A hierarchical convergence analysis

The convergence results presented above may be summarized as follows. Under assumption A1, the possible outcomes of applying the MADS-PB algorithm to problem (1) are

- i- The sequence of frame centers is unbounded.
- ii- Under assumption A2, there exists a convergent refining subsequence, converging to some refined point \hat{x} .
- iii- In addition to -ii-, if $\hat{x} \in X$ and if h is Lipschitz near \hat{x} , and if the set of refining directions is dense in $T_X^H(\hat{x}) \neq \emptyset$ then \hat{x} is a Clarke stationary point for the minimization of h over X .
- iv- In addition to -ii-, if $\hat{x} \in \Omega \subseteq X$ and if h is Lipschitz near \hat{x} , and if the set of refining directions is dense in $T_\Omega^H(\hat{x}) \neq \emptyset$, and if the refining subsequence contained infinitely many feasible frame centers³, then \hat{x} is a Clarke stationary point for the minimization of f over Ω .

The results -iii- and -iv- require that the set of refining directions of the both feasible and infeasible refining subsequences formed a dense set of directions. This is ensured by the LTMADS way of defining the polling directions [6].

The above convergence analysis may be pushed further by assuming more on the differentiability of f and on the nature of the tangent cones. We refer the reader to [6] for definitions of *strict differentiability*, *regularity* and of the *contingent cone*. With these notions, we may extend the hierarchy of convergence results to:

- v- In addition to -iii-, if h is strictly differentiable at \hat{x} , then \hat{x} is a Clarke KKT stationary point for the minimization of h over X .
- vi- In addition to -iv-, if f is strictly differentiable at \hat{x} , then \hat{x} is a Clarke KKT stationary point for the minimization of f over Ω .
- vii- In addition to -iii-, if X is regular at \hat{x} , then \hat{x} is a contingent stationary point for the minimization of h over X .
- viii- In addition to -iv-, if Ω is regular at \hat{x} , then \hat{x} is a contingent stationary point for the minimization of f over Ω .
- ix- If -v- and -vii- hold, then \hat{x} is a contingent KKT stationary point for the minimization of h over X .

³ Assumption A3 is sufficient, but not necessary, to ensure the existence of infinitely many feasible frame centers.

-x- If -vi- and -viii- hold, then \hat{x} is a contingent KKT stationary point for the minimization of f over Ω .

The proof of the results -v- through -x- are practically identical to the similar results in [6], and are omitted here.

4 Numerical results

We compare four types of runs. The first three use methods already in the literature: GPS under the extreme barrier approach [26, 4], GPS with the filter approach described in [5] and LTMADS with the extreme barrier approach [6]. The two other runs are both with the present MADS-PB approach with a standard primary poll set. They are differentiated by using either one or two secondary poll directions and labelled as LTMADS-PB 1 and LTMADS-PB 2, respectively. Due to the randomness present in the LTMADS algorithm, the reported results are the average of five distinct calls with different random seeds.

In all runs, the default parameters are used: $D = [I_k \ -I_k]$ is the standard $2n$ set of coordinate directions, in GPS the poll points are reordered by success, and in LTMADS the opportunistic SEARCH is performed (these strategies are detailed in [6]). The frame around the secondary poll center will be constructed using either the single direction $-b(\ell)$ from page 203 of [6], or the two opposite directions $-b(\ell)$ and $b(\ell)$.

We consider three different problems. The first two are there to compare the behavior of the algorithm on convex and non-convex problems of dimensions ranging from 5 to 50. These two problems can easily be solved analytically to ensure that we know the correct solution. The third problem is an engineering problem with a black box function.

For all three problems, we report results from both feasible and infeasible starting points. The runs that use the extreme barrier approach from an infeasible point are performed in two phases: First, a feasible point is found by solving the problem (8) using GPS-EB or LTMADS-EB and stopping as soon as a point with $h(x) = 0$ is found. Second, this feasible point is used as starting point for solving problem (1). The number of function evaluations of both steps are taken into account.

We give plots of the progression of the incumbent feasible objective function value versus the number of evaluations.

4.1 Linear optimization on an hypersphere

The following convex optimization problem was posed in [6].

$$\begin{aligned}
& \min_{x \in \mathbb{R}^n} \sum_{i=1}^n x_i \\
& \text{s.t.} \quad \sum_{i=1}^n x_i^2 \leq 3n,
\end{aligned}$$

Starting points :

Feasible	$(0, 0, 0, \dots, 0, 0)$
Infeasible	$(3, 3, 3, \dots, 3, 3)$.

There is a single global optimal solution to that problem: $x_i^* = -\sqrt{3}$ for all i and $f(x^*) = -\sqrt{3}n$. The purpose of this simple example is to illustrate the effect of the dimension. We will test the values $n = 5, 10, 20$ and 50 on two sets of runs. The algorithm terminates at the $600n^{\text{th}}$ function evaluation.

Figures 6 and 7 illustrate the behavior of the algorithm from the feasible and infeasible starting points, respectively.

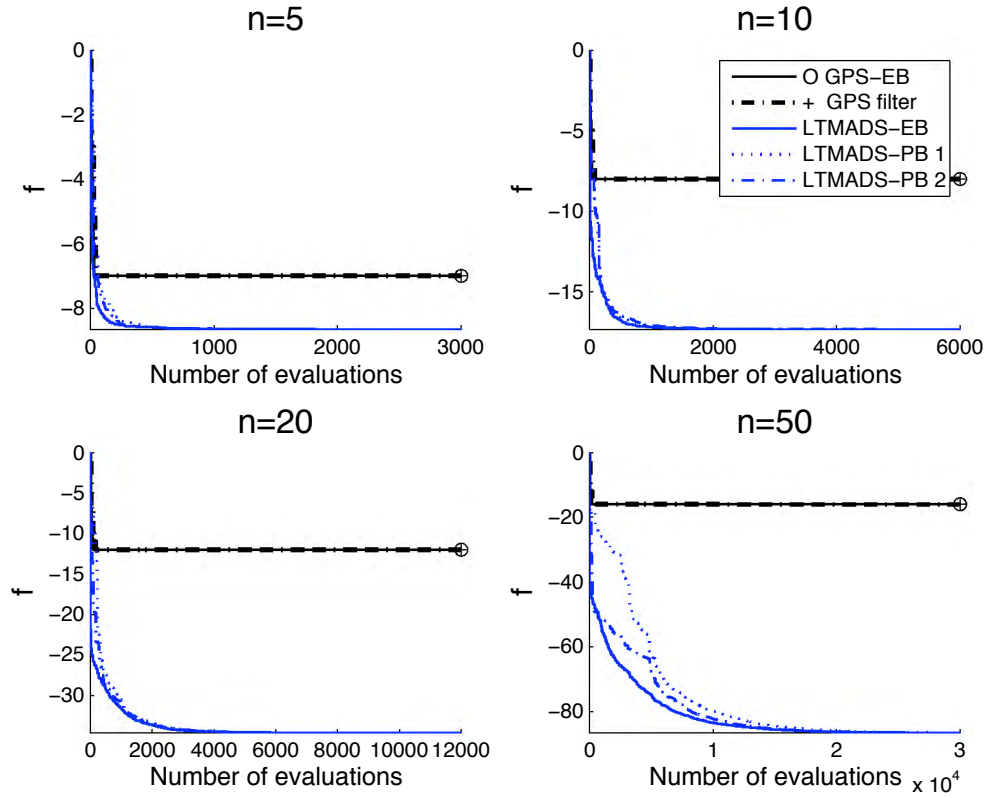


Figure 6: Progression of the objective function value vs the number of evaluations on a convex problem from a feasible starting point.

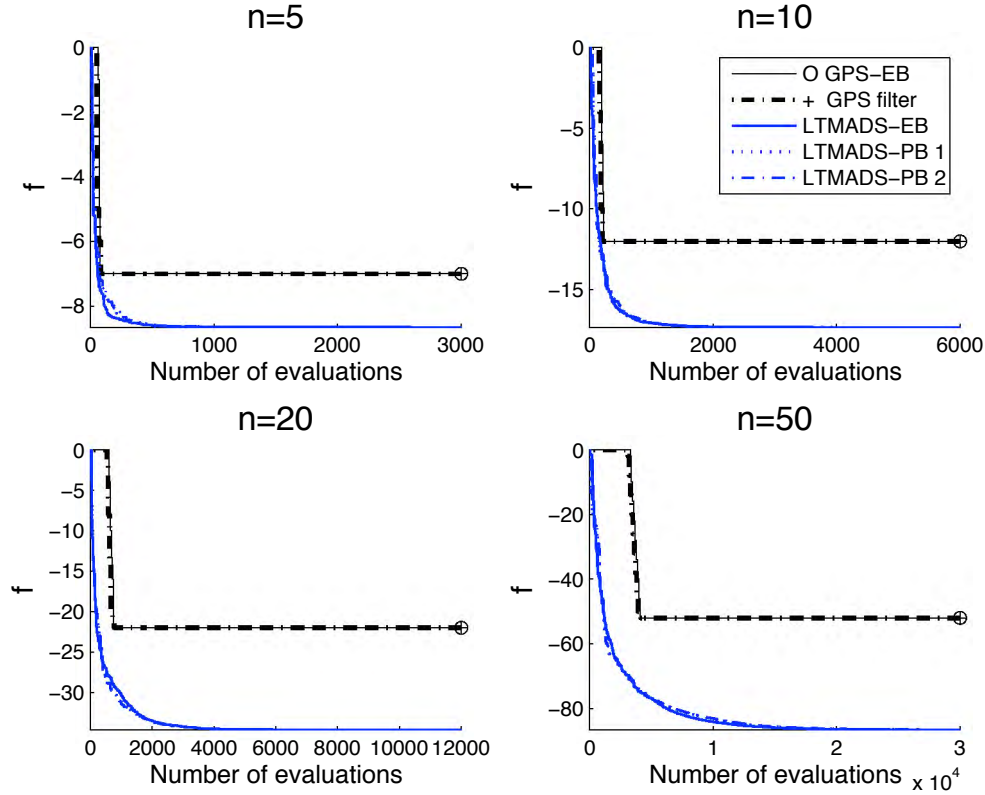


Figure 7: Progression of the objective function value vs the number of evaluations on a convex problem from an infeasible starting point.

One can observe that all runs involving LTMADS converge to the global minimizer. The GPS runs are very similar and converge to a suboptimal point on the boundary of the domain.

The feasible domain for this problem is convex and full-dimensional. Thus, LTMADS-EB has no difficulty finding a feasible point from an infeasible start. LTMADS-PB behaves similarly except for $n = 50$ when starting from a feasible point. The logs of the runs reveals that a similar behavior occurs in two of the five LTMADS-PB runs with a single secondary direction, and in one of the runs with two secondary directions. The behavior is that the first infeasible trial point generated has a large value of h . Then, for several iterations, the infeasible incumbents are the primary poll centers, and a lot of function evaluations are used to move back toward the domain.

To investigate the role of the initial barrier parameter, we have made some runs on the problem with $n = 50$ from the feasible starting point setting h_0^{\max} to 0, 100, 1000, 10000 and 100000. These are illustrated on Figure 8 zooming in on the first 15000 function evaluations. The first infeasible trial point generated by LTMADS always has an h value inferior to 10000, and therefore the runs with $h_0^{\max} = 10000, 100000$ or ∞ are identical. Setting $h_0^{\max} = 0$ is equivalent to applying LTMADS-EB, which in this case turns out to be among the best strategies. It also appears in this case that the use of a single secondary direction is preferable to using two such directions. This suggests the following strategy for the choice of h_0^{\max} : Set it to zero if there is no infeasible starting point, otherwise set it to infinity.

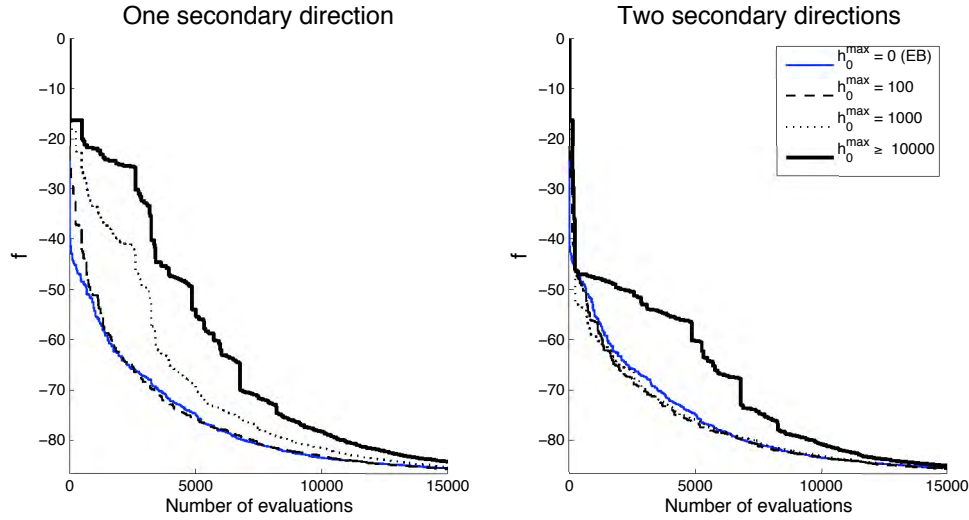


Figure 8: Progression of the objective function value vs the number of evaluations on a 50 variable convex problem from an feasible starting point with various values of h_0^{\max} .

4.2 Linear optimization over a non-convex set

Consider the optimization of a linear function over a non-convex domain:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x_n \\ \text{s.t.} \quad & \sum_{i=1}^n (x_i - 1)^2 \leq n^2 \leq \sum_{i=1}^n (x_i + 1)^2, \end{aligned}$$

Starting points :

Feasible $(n, 0, 0, \dots, 0, 0)$
Infeasible $(n, 0, 0, \dots, 0, -n)$.

There is a single optimal solution to that problem: $x^* = (1, 1, 1, \dots, 1, 1 - n)^T$ with $f(x^*) = 1 - n$. The algorithm terminates at the $600n^{\text{th}}$ function evaluation.

Figures 9 and 10 illustrate the behavior of the algorithm from the feasible and infeasible starting point, respectively.

Again, both GPS runs, from the feasible and infesasible starting points, fail to approach the global solution because. GPS always generates trial points along the same fixed directions.

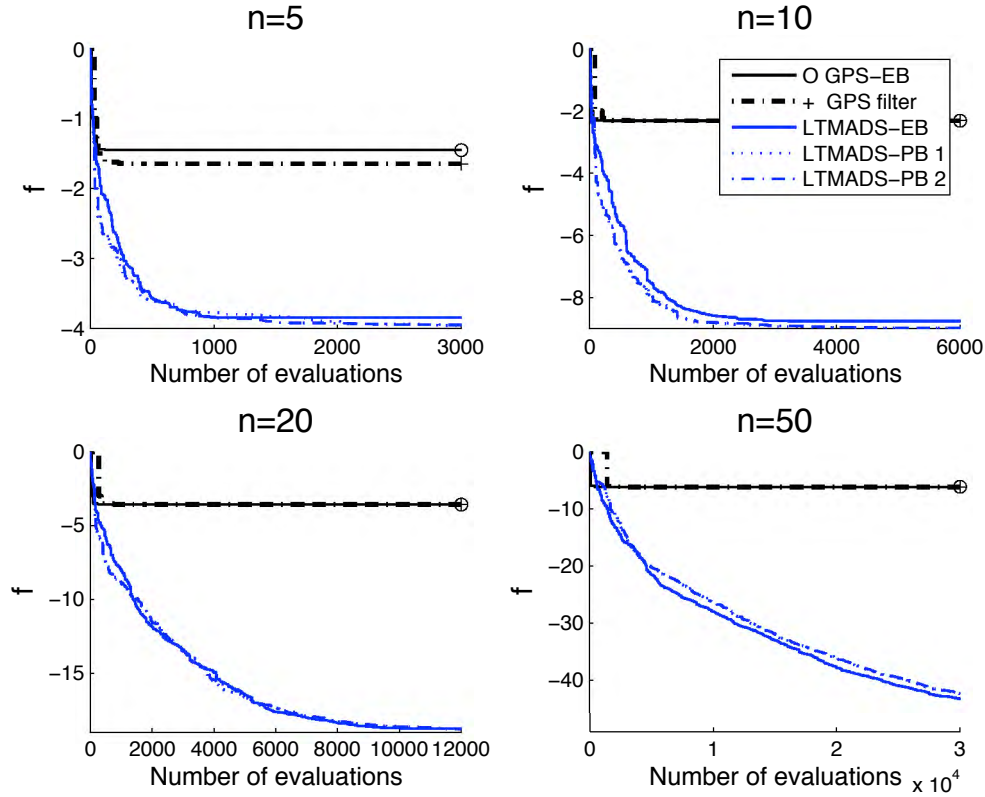


Figure 9: Progression of the objective function value vs the number of evaluations on a non-convex problem from a feasible starting point.

From the feasible starting point, both the extreme and progressive barrier approach produce similar results, as expected. However, the usefulness of the progressive barrier approach is confirmed when starting from the infeasible point. Table 4.2 gives the average number of function evaluations to generate a first feasible

solution and the objective function value at this feasible point. It also gives the best value found by the end of the run. These statistics are given for both LTMADS-EB and LTMADS-PB. Both strategies where there are a single and two secondary directions are combined since they give similar results.

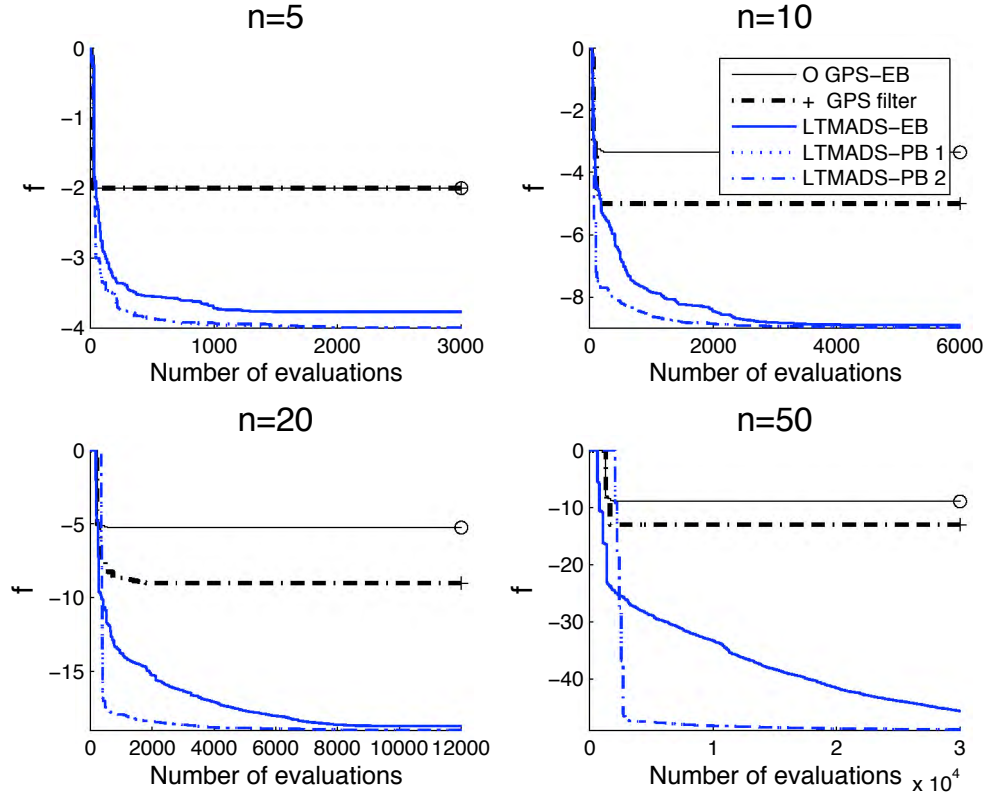


Figure 10: Progression of the objective function value vs the number of evaluations on a non-convex problem from an infeasible starting point

Let us analyze the situation where $n = 50$ in more detail because this is our first example to illustrate the effectiveness of the progressive barrier approach. The extreme barrier strategy required on average 1004 evaluations to generate a feasible point while solving (8). The average objective function value was -22.2 (all values were between -4 and -33). LTMADS-PB required 2402 evaluations (more than twice the number of evaluations) to reach feasibility. But, since the progressive barrier approach gives some importance to the objective function while searching for a feasible solution, it always generated a solution whose value is -46.0 . Ob-

n	LTMADS-EB			LTMADS-PB		
	First feasible sol		Best sol	First feasible sol		Best sol
	# evals	f -value	f -value	# evals	f -value	f -value
5	18.8	-0.4	-3.766	29.0	-2.6	-3.992
10	49.0	-1.4	-8.896	76.2	-7.2	-8.981
20	181.2	-8.0	-18.680	374.2	-16.8	-18.974
50	1004.0	-22.2	-45.902	2401.9	-46.0	-48.795

Table 1: Comparison of LTMADS with an extreme barrier and a progressive barrier

serve that this value is better than any generated by LTMADS-EB even after 50,000 evaluations, and the average function value at the 2402-th evaluation of LTMADS-EB is only -25.481 . Clearly the progressive barrier approach used its strategy of trying to decrease both f and h to go to a better part of the feasible region as we hoped.

4.3 Optimization of a styrene production process

In [1], we model the optimization of a styrene process production process with 8 continuous variables, and 4 closed yes-no constraints and 7 open constraints. Each call to the black box requires between 1 and 3 seconds and still may fail to return a value for some input parameters. The c++ code is freely available [12] and can be used by designers of other derivative-free methods. The starting points are

$$\begin{aligned} \text{Feasible} & \quad (0.54, 0.66, 0.86, 0.08, 0.29, 0.51, 0.32, 0.15) \\ \text{Infeasible} & \quad (0.44, 0.99, 0.76, 0.39, 0.39, 0.48, 0.43, 0.05). \end{aligned}$$

Figures 11 illustrate the behavior of the algorithm from both starting points.

Once again, the LTMADS runs outperform the GPS ones. The LTMADS-EB and LTMADS-PB runs from the feasible starting point again are similar to each other. The LTMADS-PB runs with one or two secondary directions are also similar. Once that feasibility is reached, LTMADS-PB reduces the feasible incumbent function value faster than LTMADS-EB.

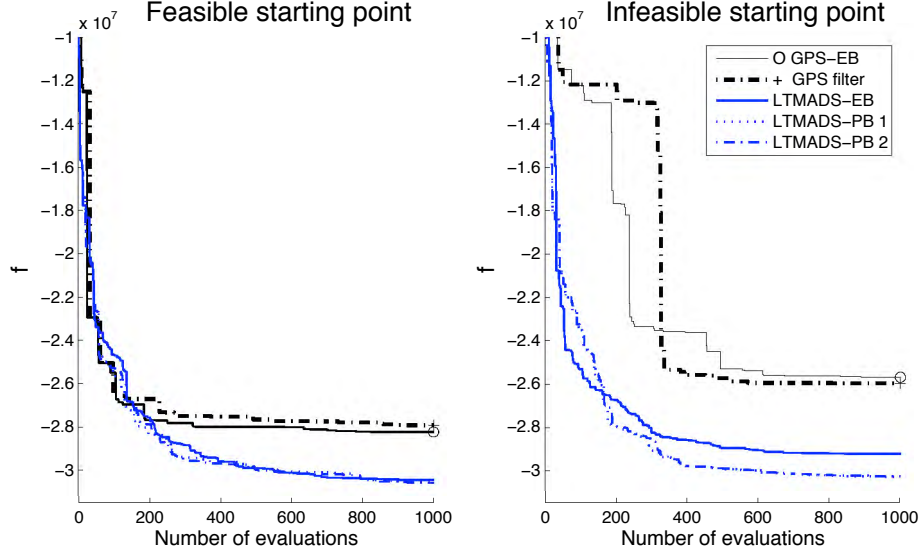


Figure 11: Progression of the objective function value vs the number of evaluations on the styrene problem.

5 Discussion

The objective of this paper was to present an alternative to the barrier approach to handle constraints in the context of the MADS algorithm. Our algorithm allows infeasible points whose constraint violation function value is below a threshold h_k^{\max} that depends adaptively on the iteration number k . This threshold is non-increasing with respect to k . When an initial feasible point is known, setting this value to zero reduces the algorithm to MADS-EB [6].

Our numerical experiments suggest that our new approach is not necessarily better than LTMADS-EB when a feasible starting point is known. Thus, a user might set h_0^{\max} to a small value, or perhaps even to 0, when a feasible starting point is given. In the test problems that we considered, the sequence of feasible and infeasible incumbents were converging to the same solution. There was a case where a lot of infeasible solutions were generated. This indicates the utility of h_0^{\max} as a control.

The main use of our new approach is for non-convex problems where no initial feasible point is known. In all these cases, LTMADS-PB converged faster than a two-phase LTMADS-EB approach. The two-phase approach neglects the objective function in the first phase and generates a first feasible point with a larger objective

function value. The LTMADS-PB approach takes more time to reach feasibility, but this first feasible point is usually much closer to the global solution.

We need more tests, but we tentatively conclude that since LTMADS-PB is better from infeasible starts and about the same from feasible starts, it is the better choice. The earlier GPS approaches seem to be noncompetitive. However, the artful use of surrogates can make all these algorithms more effective for difficult problems. See [2, 5] for some GPS-filter results illustrating this point.

References

- [1] C. Audet, V. Bécard, and S. Le Digabel. Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. Technical Report G-2006-61, Les Cahiers du GERAD, 2006.
- [2] C. Audet, A. J. Booker, J. E. Dennis, Jr., P. D. Frank, and D. W. Moore. A surrogate-model-based method for constrained optimization. AIAA Paper 2000-4891, Presented at the 8th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, California, 2000.
- [3] C. Audet, A.L. Custódio, and J. E. Dennis, Jr. Erratum : Mesh adaptive direct search algorithms for constrained optimization. Technical Report G-2006-53, Les Cahiers du Gerad, 2006.
- [4] C. Audet and J. E. Dennis, Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2003.
- [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 D. Orban. Finding optimal algorithmic parameters using the mesh adaptive direct search algorithm. *SIAM Journal on Optimization*, 17(3):642–664, 2006.
- [8] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1976.
- [9] 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.
- [10] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley, New York, 1983. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.
- [11] I. D. Coope and C. J. Price. Frame-based methods for unconstrained optimization. *Journal of Optimization Theory and Applications*, 107(2):261–274, 2000.

- [12] G. Couture, C. Audet, J. E. Dennis, Jr., and M. A. Abramson. The NOMAD project. <http://www.gerad.ca/NOMAD/>.
- [13] C. Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76:733–746, 1954.
- [14] J. E. Dennis, Jr., M. El-Alem M., and K. Williamson. A trust-region approach to non-linear systems of equalities and inequalities. *SIAM Journal on Optimization*, 9(2):291–315, 1999.
- [15] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, Series A, 91:239–269, 2002.
- [16] F. J. Gould and J. W. Tolle. Geometry of optimality conditions and constraint qualifications. *Mathematical Programming*, 2:1–18, 1972.
- [17] J. Jahn. *Introduction to the Theory of Nonlinear Optimization*. Springer, Berlin, 1994.
- [18] E. B. Leach. A note on inverse function theorem. In *Proceedings of the AMS*, volume 12, pages 694–697, 1961.
- [19] A. L. Marsden. *Aerodynamic Noise Control by Optimal Shape Design*. PhD thesis, Stanford University, 2004.
- [20] A. L. Marsden, M. Wang, J. E. Dennis, Jr., and P. Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*, 5(2):235–262, 2004.
- [21] A. L. Marsden, M. Wang, J. E. Dennis, Jr., and P. Moin. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *J. Fluid Mech.*, 572:13–36, 2007.
- [22] A. L. Marsden, M. Wang, J. E. Dennis, Jr., and P. Moin. Suppression of airfoil vortex-shedding noise via derivative-free optimization. *Physics of Fluids*, 16(10):L83–L86, 2004.
- [23] C. J. Price and I. D. Coope. Frame based ray search algorithms in unconstrained optimization. *Journal of Optimization Theory and Applications*, 116:259–377, 2003.
- [24] C. J. Price and I. D. Coope. Frames and grids in unconstrained and linearly constrained optimization: A nonsmooth approach. *SIAM Journal on Optimization*, 14(2):415–438, 2003.
- [25] R. T. Rockafellar. Generalized directional derivatives and subgradients of nonconvex functions. *Canad. J. Math.*, 32(2):257–280, 1980.
- [26] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, February 1997.