Mesh Adaptive Direct Search Algorithms for Constrained Optimization

Article *in* SIAM Journal on Optimization · January 2006

DOI: 10.1137/060671267 · Source: DBLP

2 authors:

Charles Audet
Polytechnique Montréal
177 PUBLICATIONS 7,987 CITATIONS

SEE PROFILE

J. E. Dennis
Rice University
173 PUBLICATIONS 29,954 CITATIONS

SEE PROFILE

SIAM J. OPTIM.
Vol. x, No. x, pp. xxx-xxx

MESH ADAPTIVE DIRECT SEARCH ALGORITHMS FOR CONSTRAINED OPTIMIZATION [*]

CHARLES AUDET [†] AND J.E. DENNIS JR. [‡]

Abstract. This paper introduces the Mesh Adaptive Direct Search (MADS) class of algorithms for nonlinear optimization. MADS extends the Generalized Pattern Search (GPS) class by allowing local exploration, called *polling*, in an asymptotically dense set of directions in the space of optimization variables. This means that under certain hypotheses, including a weak constraint qualification due to Rockafellar, MADS can treat constraints by the extreme *barrier* approach of setting the objective to infinity for infeasible points and treating the problem as unconstrained.

The main GPS convergence result is to identify limit points $\hat{x}$, where the Clarke generalized derivatives are nonnegative in a finite set of directions, called *refining directions*. Although in the unconstrained case, nonnegative combinations of these directions span the whole space, the fact that there can only be finitely many GPS refining directions limits rigorous justification of the barrier approach to finitely many linear constraints for GPS. The main result of this paper is that the MADS algorithms can generate an asymptotically dense set of refining directions.

For LTMADS, an implementable instance of MADS, the refining directions are dense in the hypertangent cone at $\hat{x}$ with probability 1. This result holds if the iterates associated with the refining directions converge to a single $\hat{x}$. We compare LTMADS to versions of GPS on some test problems. We also illustrate the limitation of our results with examples.

Key words. Mesh adaptive direct search algorithms (MADS), convergence analysis, constrained optimization, nonsmooth analysis, Clarke derivatives, hypertangent, contingent cone.

1. Introduction.

We present and analyze a new *Mesh Adaptive Direct Search* (MADS) class of algorithms for minimizing a nonsmooth function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ under general constraints $x \in \Omega \neq \emptyset \subseteq \mathbb{R}^n$. For the form of the algorithm given here, the feasible region Ω may be defined through blackbox constraints given by an oracle, such as a computer code that returns a yes or no indicating whether or not a specified trial point is feasible.

In the unconstrained case, where $\Omega = \mathbb{R}^n$, this new class of algorithms occupies a position somewhere between the Generalized Pattern Search (GPS) class [22], as organized in [6], and the Coope and Price frame-based methods [10]. A key advantage of MADS over GPS is that local exploration of the space of variables is not restricted to a finite number of directions (called *poll directions*). This is the primary drawback of GPS algorithms in our opinion, and our main motivation in defining MADS was to overcome this restriction. MADS algorithms are frame-based methods. We propose a less general choice of frames than the choices allowed by Coope and Price. Our MADS frames are easy to implement, and they are specifically aimed at ensuring an asymptotically dense set of polling directions. We illustrate our ideas with an example algorithm that we call LTMADS because it is based on a random lower triangular matrix.

The convergence analysis here is based on Clarke's calculus [8] for nonsmooth functions. The analysis evolved from our previous work on GPS [3] where we gave a hierarchy of convergence results for GPS that show the limitations inherent in the restriction to finitely many directions. Specifically, we showed that for unconstrained optimization, GPS produces a limit point at which the gradient is zero if the function at that point is strictly differentiable [17]. Strict differentiability is just the requirement that the generalized gradient is a singleton, i.e., that $\partial f(\hat{x}) = \{\nabla f(\hat{x})\}$ in addition to the requirement that f is Lipschitz near $\hat{x}$. But if the function f is only Lipschitz near such a limit point $\hat{x}$, then Clarke's generalized directional derivatives [8]

[*] Work of the first author was supported by FCAR grant NC72792 and NSERC grant 239436-01, and both authors were supported by AFOSR FA9550-04-1-0235, The Boeing Company, and ExxonMobil.

[†] GERAD and Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal (Québec), H3C 3A7 Canada (Charles.Audet@gerad.ca, http://www.gerad.ca/Charles.Audet)

[‡] Computational and Applied Mathematics Department, Rice University, 8419 42nd Ave SW, Seattle, WA 98136 (dennis@rice.edu, http://www.caam.rice.edu/∼dennis)

are provably nonnegative only for a finite set of directions $\hat{D} \subset \mathbb{R}^n$ whose nonnegative linear combinations span the whole space:

$$f^\circ(\hat{x};d) := \limsup_{y\to\hat{x},\ t\downarrow 0} \frac{f(y+td)-f(y)}{t} \ \geq\ 0 \quad \text{for all } d \in \hat{D}. \tag{1.1}$$

$\hat{D}$ is called the set of *refining directions*. This result 1.1 for GPS is not as strong as stating that the generalized derivative is nonnegative for every direction in $\mathbb{R}^n$, i.e., that the limit point is a Clarke stationary point, or equivalently that $0 \in \partial f(\hat{x})$, the generalized gradient of f at $\hat{x}$ defined by

$$f^\circ(\hat{x};v) \geq 0 \ \text{ for all } v \in \mathbb{R}^n \quad \Leftrightarrow \quad 0 \in \partial f(\hat{x}) := \{s \in \mathbb{R}^n : f^\circ(\hat{x};v) \geq v^T s \text{ for all } v \in \mathbb{R}^n\}. \tag{1.2}$$

Example F in [2] shows that indeed the GPS algorithm does not necessarily produce a Clarke stationary point for Lipschitz functions because of the restriction to finitely many poll directions. This is so even if the gradient exists at the limit point $\hat{x}$. For the unconstrained case, this restriction can be overcome by assuming more smoothness for f, e.g., strict differentiability at $\hat{x}$ [3] as mentioned above.

However, the directional dependence of GPS in the presence of even bound constraints cannot be overcome by any amount of smoothness, by using penalty functions, or by the use of the filter approach for handling constraints [4]. In contrast, MADS produces unconstrained limit points at which the Clarke derivatives are nonnegative for every direction in $\mathbb{R}^n$.

Besides the advantages for the unconstrained case of an asymptotically dense set of refining directions, MADS can also treat a wide class of nonlinear constraints by the “barrier” approach. By this we mean that the algorithm is not applied directly to f but to the barrier function f_Ω, defined to be equal to f on Ω and $+\infty$ outside Ω. This way of rejecting infeasible points was shown to be effective for GPS with finitely many linear constraints by Lewis and Torczon [18]. However, their proof requires that the tangent cone generators of the feasible region at boundary points near an iterate be known at each iteration. For LTMADS, no special effort is needed for the barrier approach to be provably effective with probability 1 on nonlinear constraints satisfying a reasonable constraint qualification due to Rockafellar [21] – that there exists a hypertangent vector at the limit point. A key advantage of the barrier approach is that one can avoid expensive function calls to f whenever a constraint is violated. Indeed, the question of feasibility of a trial point needs only a yes or no answer - the constraints do not need to be given by a known algebraic condition.

The class of algorithms presented here differs significantly from previous GPS extensions [4, 19] to nonlinear constraints. Treating constraints as we do motivates us to use the generalization of the Clarke derivative presented in Jahn [15]. Jahn's approach is aimed at a case like ours where the evaluation of f is restricted to points in the feasible domain Ω. Thus we use the following definition of the Clarke generalized derivative at $\hat{x} \in \Omega$ in the direction $v \in \mathbb{R}^n$:

$$f^\circ(\hat{x};v) := \limsup_{\substack{y\to\hat{x},\ y\in\Omega \\ t\downarrow 0,\ y+tv\in\Omega}} \frac{f(y+tv)-f(y)}{t}. \tag{1.3}$$

Both definitions (1.1) and (1.3) coincide when $\Omega = \mathbb{R}^n$ or when $\hat{x} \in int(\Omega)$.

The main theoretical objective of this paper is to show that under appropriate assumptions, any MADS algorithm produces a constrained Clarke stationary point, *i.e.*, a limit point $\hat{x} \in \Omega$ satisfying the following necessary optimality condition

$$f^\circ(\hat{x};v) \ \geq\ 0 \text{ for all } v \in T^{Cl}_\Omega(\hat{x}), \tag{1.4}$$

where $T^{Cl}_\Omega(\hat{x})$ is the Clarke tangent cone to Ω at $\hat{x}$ (see [8] or Definition 3.5).

The paper is organized into two main parts. First, Sections 2 and 3 present the abstract MADS algorithm class and its convergence analysis. The analysis revolves around three types of tangent cones. This allows us to tie some convergence results to local differentiability of the function f at limit points satisfying certain constraint qualifications. We present sufficient conditions under which (1.4) holds. We discuss the consequences of this when the algorithm is applied to an unconstrained problem, or when the set Ω is regular in the sense of Definition 3.7 or [8]. We also give a stronger contraint qualification ensuring that MADS produces a contingent KKT stationary point if f is strictly differentiable. The reader will find a quite different algorithm analyzed using the same concepts in [12].

Then in Sections 4 and 5, we give an implementable instance of MADS along with numerical experiments to compare MADS with standard GPS. On an artificial example where GPS and Nelder-Mead are well known to stagnate, we show that MADS reaches the global optimum. We give a comparison on a parameter fitting problem in catalytic combustion kinetics on which we know that GPS performs well [14]. We also give an example illustrating the power of being able to handle even simple nonlinear constraints by the barrier approach. We also use this example to illustrate that MADS can cope surprisingly well as the dimension of the problem increases. The final example shows the value of randomly generated polling directions for a problem with a narrowing feasible region.

Notation. $\mathbb{R}, \mathbb{Z}$ and $\mathbb{N}$ respectively denote the sets of real numbers, integers, and nonnegative integers. For $x \in \mathbb{R}^n$ and $\delta \in \mathbb{R}_+$, $B_\delta(x)$ denotes the open ball of radius δ centered at x. For a matrix D, the notation $d \in D$ indicates that d is a column of D. The iteration numbers are denoted by the index k.

2. Mesh Adaptive Direct Search algorithms. Given an initial iterate $x_0 \in \Omega$, a MADS algorithm attempts to locate a minimizer of the function f over Ω by evaluating f_Ω at some trial points. The algorithm does not require any derivative information for f. This is useful when there are several local optima. But it is essential when ∇f is unavailable, either because it does not exist, or it cannot be accurately estimated due to noise in f or other reasons.

MADS is an iterative algorithm where at each iteration a finite number of trial points are generated, and the infeasible trial points are discarded. The objective function values at the feasible trial points are compared with the current incumbent value $f_\Omega(x_k)$, *i.e.*, the best feasible objective function value found so far. Each of these trial points lies on the *current mesh*, constructed from a finite set of n_D directions $D \subset \mathbb{R}^n$ scaled by a *mesh size parameter* $\Delta_k^m \in \mathbb{R}_+$.

There are two restrictions on the set D. First, D must be a positive spanning set [11], *i.e.*, nonnegative linear combinations of its elements must span $\mathbb{R}^n$. Second, each direction $d_j \in D$ (for $j = 1, 2, \ldots, n_D$) must be the product Gz_j of some fixed non-singular generating matrix $G \in \mathbb{R}^{n \times n}$ by an integer vector $z_j \in \mathbb{Z}^n$. For convenience, the set D is also viewed as a real $n \times n_D$ matrix.

DEFINITION 2.1. *At iteration k, the current mesh is defined to be the following union:*

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

where S_k is the set of points where the objective function f had been evaluated by the start of iteration k.

In the definition above, the mesh is defined to be the union of sets over S_k. Defining the mesh this way ensures that all previously visited points lie on the mesh, and that new trial points can be selected around any of them using the directions in D. This definition of the mesh is identical to the one in [4] and generalizes the one in [3].

The mesh is conceptual in the sense that it is never actually constructed. In practice, one can easily make sure that the strategy for generating trial points is such that they all belong to the mesh. One simply has to verify in Definition 2.1 that x belongs to S_k and that z is an integer vector. The objective of the iteration is to find a trial mesh point with a lower objective function value than the current incumbent value $f_\Omega(x_k)$. Such a trial point is called an *improved mesh point*, and the iteration is called a *successful iteration*. There are no sufficient decrease requirements on the objective function value.

The evaluation of f_Ω at a trial point x is done as follows. First, the constraints defining Ω are tested to determine if x is feasible or not. Indeed, since some of the constraints defining Ω might be expensive or inconvenient to test, one would order the constraints to test the easiest ones first. If $x \notin \Omega$, then $f_\Omega(x)$ is set to $+\infty$ without evaluating $f(x)$, and perhaps without evaluation all the constraints defining Ω. In effect, this means we discard the infeasible trial points. On the other hand, if $x \in \Omega$, then $f(x)$ is evaluated. This remark may seem obvious, but it saves computation, and it is needed in the proof of Theorem 3.12.

Each iteration is divided into two steps. The first, called the SEARCH step, has the same flexibility as in GPS. It allows evaluation of f_Ω at any finite number of mesh points. Any strategy can be used in the SEARCH step to generate a finite number of trial mesh points. Restricting the SEARCH points to lie on the mesh is a way in which MADS is less general than the frame methods of Coope and Price [10]. The SEARCH is said to be empty when no trial points are considered. The drawback to the SEARCH flexibility is that it cannot be used in the convergence analysis – except to provide counterexamples as in [2]. More discussion of SEARCH steps is given in [1, 20, 6].

When an improved mesh point is generated, then the iteration may stop, or it may continue if the user hopes to find a better improved mesh point. In either case, the next iteration will be initiated with a new incumbent solution $x_{k+1} \in \Omega$ with $f_\Omega(x_{k+1}) < f_\Omega(x_k)$ and with a mesh size parameter Δ^m_{k+1} equal to or larger than Δ^m_k (the exact rules for updating this parameter are presented below). Coarsening the mesh when improvements in f_Ω are obtained can speed convergence.

Whenever the SEARCH step fails to generate an improved mesh point, then the second step, called the POLL, is invoked before terminating the iteration. The difference between the MADS and the GPS algorithms lies exactly in this POLL step. For this reason, our numerical comparisons in the sequel use empty, or very simple, SEARCH steps in order to isolate the value of the MADS POLL step.

When the iteration fails in generating an improved mesh point, then the next iteration is initiated from any point $x_{k+1} \in S_{k+1}$ with $f_\Omega(x_{k+1}) = f_\Omega(x_k)$; though there is usually a single such incumbent solution, and then x_{k+1} is set to x_k. The mesh size parameter Δ^m_{k+1} is reduced to increase the mesh resolution, and therefore to allow the evaluation of f at trial points closer to the incumbent solution. 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 follows:

$$\Delta^m_{k+1} \;=\; \tau^{w_k}\Delta^m_k \;\text{ for some } w_k \in \left\{ \begin{array}{cl} \{0,1,\ldots,w^+\} & \text{if an improved mesh point is found} \\ \{w^-,w^-+1,\ldots,-1\} & \text{otherwise.} \end{array} \right. \tag{2.1}$$

Everything up to this point in the section applies to both GPS and MADS. We now present the key difference between both classes of algorithms. For MADS, we introduce the *poll size parameter* $\Delta^p_k \in \mathbb{R}_+$ for iteration k. This new parameter dictates the magnitude of the distance from the trial points generated by the POLL step to the current incumbent solution x_k. In GPS, there is a single parameter to represent these quantities: $\Delta_k = \Delta^p_k = \Delta^m_k$. In MADS, the strategy for updating Δ^p_k must be such that $\Delta^m_k \leq \Delta^p_k$ for all k, and moreover, it must satisfy

$$\lim_{k\in K}\Delta^m_k = 0 \;\text{ if and only if }\; \lim_{k\in K}\Delta^p_k = 0 \;\text{ for any infinite subset of indices } K. \tag{2.2}$$

An implementable updating strategy satisfying these requirements is presented in Section 4.

We now move away from the GPS terminology, and toward that of Coope and Price. The set of trial points considered during the POLL step is called a *frame*. The frames of Coope and Price can be more general than MADS frames in a way not important to the present discussion. For this reason, we do not digress to discuss their general definition here [9].

The MADS frame is constructed using a current incumbent solution x_k (called the *frame center*) and the poll and mesh size parameters Δ_k^p and Δ_k^m to obtain a positive spanning set of directions D_k. Unlike GPS, generally the MADS set of directions D_k is not a subset of D.

DEFINITION 2.2. *At iteration k, the MADS frame is defined to be the set:*

$$P_k = \{x_k + \Delta_k^m d : d \in D_k\} \ \subset \ M_k \ ,$$

where D_k is a positive spanning set such that $0 \notin D_k$ and for each $d \in D_k$,

- *d can be written as a nonnegative integer combination of the directions in D: $d = Du$ for some vector $u \in \mathbb{N}^{n_{D_k}}$ that may depend on the iteration number k*
- *the distance from the frame center x_k to a frame point $x_k + \Delta_k^m d \in P_k$ is bounded by a constant times the poll size parameter: $\Delta_k^m \|d\| \leq \Delta_k^p \max\{\|d'\| : d' \in D\}$*
- *limits (as defined in Coope and Price [9]) of the normalized sets D_k are positive spanning sets.*

In GPS, the set D_k is a subset of the finite set D. There is more flexibility in MADS. We will present an instance of MADS where the closure of the cone generated by the set

$$\bigcup_{k=1}^{\infty} \left\{ \frac{d}{\|d\|} : d \in D_k \right\}$$

equals $\mathbb{R}^n$ with probability 1. We will say that the set of poll directions is asymptotically dense with probability 1.

If the POLL step fails to generate an improved mesh point then the frame is called a *minimal frame*, and the frame center x_k is said to be a *minimal frame center*. This leads to mesh refinement. At each iteration, the columns of D_k are called the poll directions.

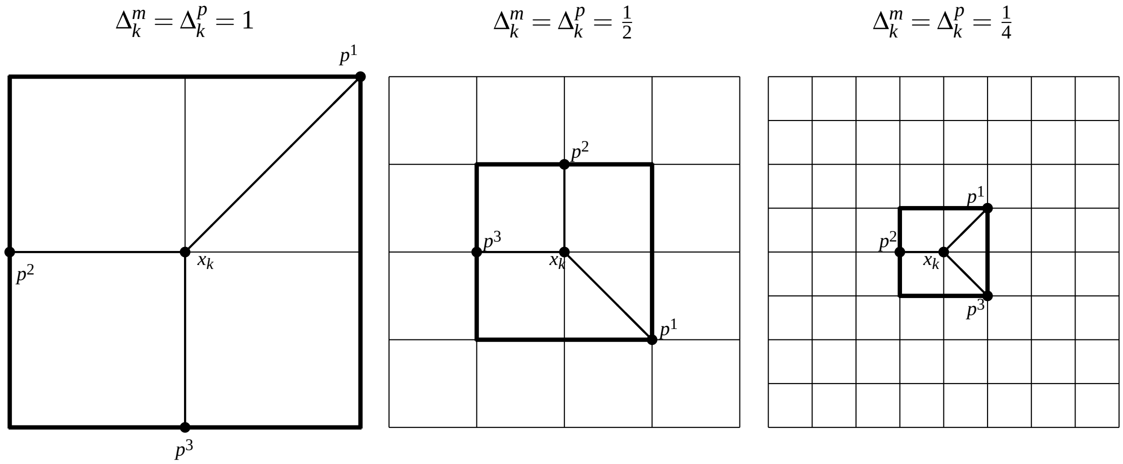
The algorithm is stated formally below. It is very similar to GPS, with differences in the POLL step, and in the new poll size parameter.

A GENERAL MADS ALGORITHM

- INITIALIZATION: Let $x_0 \in \Omega$, $\Delta_0^m \leq \Delta_0^p$, D, G, τ, w^- and w^+ satisfy the requirements given above.
  Set the iteration counter $k \leftarrow 0$.
- SEARCH AND POLL STEP: Perform the SEARCH and possibly the POLL steps (or only part of them) until an improved mesh point x_{k+1} is found on the mesh M_k (see Definition 2.1).
  - OPTIONAL SEARCH: Evaluate f_Ω on a finite subset of trial points on the mesh M_k.
  - LOCAL POLL: Evaluate f_Ω on the frame P_k (see Definition 2.2).
- PARAMETER UPDATE: Update Δ_{k+1}^m according to Equation (2.1), and Δ_{k+1}^p according to (2.2).
  Set $k \leftarrow k+1$ and go back to the SEARCH and POLL step.

The crucial distinction and advantage of MADS over GPS is that the MADS mesh size parameter Δ_k^m may go to zero more rapidly than Δ_k^p. Consequently, the directions in D_k used to define the frame may be selected in a way so that asymptotically they are not confined to a finite set. Note that in GPS both Δ_k^m and Δ_k^p are equal: a single parameter plays the role of the mesh and poll size parameters, and therefore, the number of positive spanning sets that can be formed by subsets of D is constant over all iterations.

For example, suppose that in $\mathbb{R}^2$ the set D is composed of the eight directions $\{(d_1, d_2)^T \neq (0,0)^T : d_1, d_2 \in \{-1,0,1\}\}$. There are a total of eight distinct positive bases containing three directions that can be constructed from D. Figures 2.1 and 2.2 illustrate some possible frames in $\mathbb{R}^2$ for three values of Δ_k^m. The

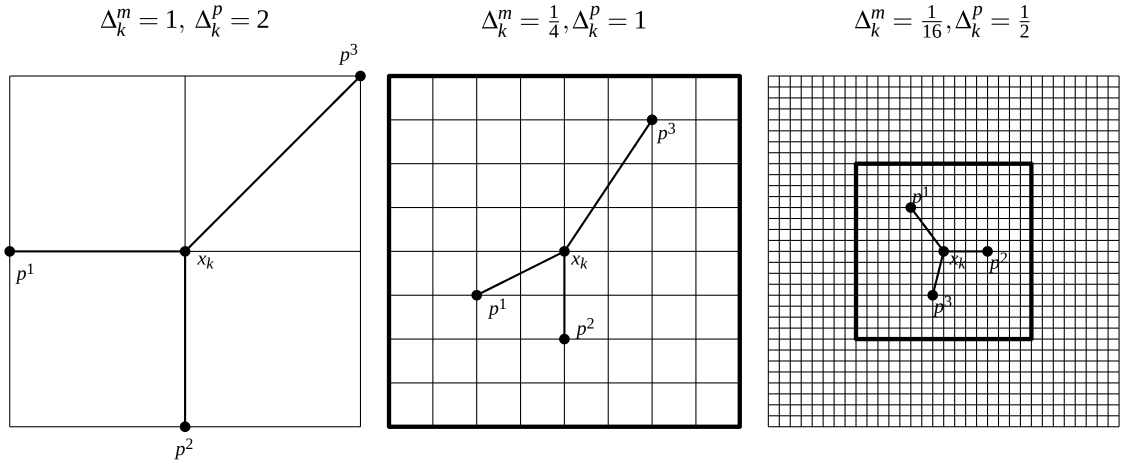


$\Delta_k^m = \Delta_k^p = 1$
$\Delta_k^m = \Delta_k^p = \frac{1}{2}$
$\Delta_k^m = \Delta_k^p = \frac{1}{4}$
p^1
p^2
p^3
x_k

FIG. 2.1. *Example of GPS frames $P_k = \{x_k + \Delta_k^m d : d \in D_k\} = \{p^1, p^2, p^3\}$ for different values of $\Delta_k^m = \Delta_k^p$. In all three figures, the mesh M_k is the intersection of all lines.*

frames in Figure 2.1 are generated by a GPS instance, and are such that $\Delta_k^p = \Delta_k^m$. Regardless of k and of the mesh or poll size parameters, each direction in D_k is confined to be selected in D.

The frames in Figure 2.2 are generated by an instance of MADS with $\Delta_k^p = n\sqrt{\Delta_k^m}$. One can see that the new MADS algorithm may select the directions of D_k from a larger set. With the new algorithm, the frame may be chosen among the mesh points lying inside the square with the dark contour. We will present in Section 4 an implementation of MADS ensuring that given any directions in $\mathbb{R}^n$, the algorithm generates arbitrarily close poll directions, i.e., that the set of poll directions is asymptotically dense in $\mathbb{R}^n$.



$\Delta_k^m = 1,\ \Delta_k^p = 2$
$\Delta_k^m = \frac{1}{4}, \Delta_k^p = 1$
$\Delta_k^m = \frac{1}{16}, \Delta_k^p = \frac{1}{2}$
p^1
p^2
p^3
x_k

FIG. 2.2. *Example of MADS frames $P_k = \{x_k + \Delta_k^m d : d \in D_k\} = \{p^1, p^2, p^3\}$ for different values of Δ_k^m and Δ_k^p. In all three figures, the mesh M_k is the intersection of all lines.*

We have presented above a general framework for MADS algorithms. The next section contains a detailed convergence analysis for that general framework. It presents sufficient conditions to ensure a hierarchy of convergence results based on the local differentiability of f (using the Clarke non-smooth calculus) and on the local properties of Ω (using three types of tangent cones). The results rely on the assumption that a specific set of directions (called the refining directions – see Definition 3.2) be dense in a tangent cone. Then, in Section 4 we propose a specific implementation called LTMADS, and give sufficient conditions to satisfy this assumption.

3. Convergence analysis of MADS. The convergence analysis below relies on the assumptions that $x_0 \in \Omega$, that $f(x_0)$ is finite, and that all iterates $\{x_k\}$ produced by the MADS algorithm lie in a compact set. Future work will relax the first assumption by incorporating the filter approach given in [4].

The section is divided into three subsections. The first recalls Torczon's [22] analysis of the behavior of the mesh size parameter and defines refining sequences as in [3]. It also defines the idea of a refining subsequence and a refining direction. The second subsection recalls the definitions of the hypertangent, Clarke, and contingent cones in addition to some results on generalized derivatives. The third contains a hierarchy of convergence results based on local properties of the feasible region Ω.

3.1. Preliminaries. Torczon [22] first showed the following result for unconstrained pattern search algorithms. Then Audet and Dennis [3] used the same technique for a description of GPS that is much closer to our description of MADS. The proof of this result for MADS is identical to that of GPS. The element necessary to the proof is that for any integer $N \geq 1$, the iterate x_N may be written as $x_N = x_0 + \sum_{k=0}^{N-1} \Delta_k^m D z_k$ for some vectors $z_k \in \mathbb{N}^{n_D}$. This is still true with our new way of defining the mesh and the frame (see Definitions 2.1 and 2.2).

PROPOSITION 3.1. *The poll and mesh size parameters produced by a MADS instance satisfy*

$$\liminf_{k \to +\infty} \Delta_k^p \ = \ \liminf_{k \to +\infty} \Delta_k^m \ = \ 0.$$

Since the mesh size parameter shrinks only at minimal frames, Proposition 3.1 guarantees that there are infinitely many minimal frame centers. The following definition specifies the subsequences of iterates and limit directions we use.

DEFINITION 3.2. *A subsequence of the MADS iterates consisting of minimal frame centers, $\{x_k\}_{k\in K}$ for some subset of indices K, is said to be a* refining subsequence *if $\{\Delta_k^p\}_{k\in K}$ converges to zero.*

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

It is shown in [3], that there exists at least one convergent refining subsequence. We now present some definitions that will be used later to guarantee the existence of refining directions.

3.2. Three types of tangent cones. Three different types of tangent cones play a central role in our analysis. Their definition, and equivalent ones, may be found in [21, 8, 15]. After presenting them, we supply an example where the three cones differ to illustrate some of our results. The first cone that we present is the hypertangent cone.

DEFINITION 3.3 (Hypertangent cone). *A vector $v \in \mathbb{R}^n$ is said to be a hypertangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point $x \in \Omega$ if there exists a scalar $\varepsilon > 0$ such that*

$$y + tw \in \Omega \qquad \text{for all } y \in \Omega \cap B_\varepsilon(x), \quad w \in B_\varepsilon(v) \ \text{ and } \ 0 < t < \varepsilon. \tag{3.1}$$

The set of hypertangent vectors to Ω at x is called the hypertangent cone to Ω at *x and is denoted by $T_\Omega^H(x)$.*

The hypertangent cone is a useful concept for understanding the behavior of the MADS algorithm. When analysing MADS, we will be concerned with specific subsequences:

- minimal frame centers $x_k \to \hat{x}$;
- mesh size parameters $\Delta_k^m \searrow 0$ and step size $\Delta_k \|d_k\| \searrow 0$;
- normalized refining directions $\frac{d_k}{\|d_k\|} \to v \neq 0$.

These subsequences will be chosen in a way so that $x_k \in \Omega$ and $x_k + (\Delta_k^m \|d_k\|) \frac{d_k}{\|d_k\|} \in \Omega$. The connection with the the hypertangent definition is obvious by noticing that the roles of y, t and w are played by $x_k, \Delta_k^m \|d_k\|$ and $\frac{d_k}{\|d_k\|}$, respectively. The connection with the Clarke derivative (1.3) will be made explicit in Theorem 3.12.

Since the definition of a hypertangent is rather technical and crucial to our results, we will pause for a short discussion. The reader could easily show that if Ω is a full dimensional polytope defined by linear constraints, then every direction from a point $\hat{x} \in \Omega$ into the interior of Ω is a hypertangent. That follows immediately from the following result relating hypertangents to the constraint qualification suggested by Gould and Tolle [13]. See also [5] for a discussion of the Gould and Tolle constraint qualification and the closely related one of Mangasarian and Fromovitz.

THEOREM 3.4. *Let $C : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable at a point $\hat{x} \in \Omega = \{x \in \mathbb{R}^n \,:\, C(x) \leq 0\}$, and let $\mathcal{A}(\hat{x}) = \{i \in \{1,2,\ldots,m\} : c_i(\hat{x}) = 0\}$ be the active set at $\hat{x}$. Then $v \in \mathbb{R}^n$ is a hypertangent vector to Ω at $\hat{x}$ if and only if $\nabla c_i(\hat{x})^T v < 0$ for each $i \in \mathcal{A}(\hat{x})$ with $\nabla c_i(\hat{x}) \neq 0$.*

Proof. Let v be a hypertangent vector to Ω at $\hat{x}$. Then, there exists an $\varepsilon > 0$ such that $\hat{x} + tv \in \Omega$ for any $0 < t < \varepsilon$. Let $i \in \mathcal{A}(\hat{x})$. Continuous differentiability of c_i at $\hat{x}$ implies that

$$\nabla c_i(\hat{x})^T v = \lim_{t \to 0} \frac{c_i(\hat{x} + tv) - c_i(\hat{x})}{t} \leq 0 \ .$$

It only remains to show that $\nabla c_i(\hat{x})^T v \neq 0$ when $\nabla c_i(\hat{x}) \neq 0$. Suppose by way of contradiction that $\nabla c_i(\hat{x})^T v = 0$ and $\nabla c_i(\hat{x}) \neq 0$. Since the hypertangent cone is an open set [21], for any nonnegative $\delta \in \mathbb{R}$ sufficiently small, $v + \delta \nabla c_i(\hat{x})$ is a hypertangent vector to Ω at $\hat{x}$. It follows that

$$0 \ \geq \ \nabla c_i(\hat{x})^T (v + \delta \nabla c_i(\hat{x})) \ = \ \delta \|\nabla c_i(\hat{x})\|_2^2 \ > \ 0,$$

which is a contradiction. Thus, $\nabla c_i(\hat{x})^T v < 0$ when $\nabla c_i(\hat{x}) \neq 0$.

To prove the converse, let $i \in \mathcal{A}(\hat{x})$ be such that $\nabla c_i(\hat{x}) \neq 0$ and $v \in \mathbb{R}^n$ be such that $\|v\| = 1$ and $\nabla c_i(\hat{x})^T v < 0$. The product $\nabla c_i(y)^T w$ is a continuous function at $(y;\ w) = (\hat{x};\ v)$, and so there is some $\varepsilon_1 > 0$ such that

$$\nabla c_i(y)^T w < 0 \ \text{ for all } \ y \in B_{\varepsilon_1}(\hat{x}) \text{ and } w \in B_{\varepsilon_1}(v). \tag{3.2}$$

Take $\varepsilon = \min\{1, \frac{\varepsilon_1}{3}\}$ and let y, w be in $B_\varepsilon(\hat{x})$ and $B_\varepsilon(v)$ respectively with $y \in \Omega$, and let $0 < t < \varepsilon$. We will show that $y + tw \in \Omega$. Our construction ensures that $c_i(y) \leq 0$ and $\varepsilon < \varepsilon_1$, and so by the mean value theorem, we have

$$c_i(y + tw) \leq c_i(y + tw) - c_i(y) = \nabla c_i(y + \theta tw)^T (tw) \ \text{ for some } \theta \in [0,1]. \tag{3.3}$$

But, $\|y + \theta tw - \hat{x}\| \leq \|y - \hat{x}\| + \theta t(\|w - v\| + \|v\|) < \varepsilon + \varepsilon(\varepsilon + 1) \leq 3\varepsilon \leq \varepsilon_1$, thus $y + \theta tw \in B_{\varepsilon_1}(\hat{x})$, and $w \in B_\varepsilon(v) \subseteq B_{\varepsilon_1}(v)$. It follows that equation (3.2) applies and therefore $\nabla c_i(y + \theta tw)^T w < 0$. Combining this with (3.3) and with the fact that $t > 0$ implies that $c_i(y + tw) \leq 0$. But c_i was any active component function, and so $C(y + tw) \leq 0$ and thus $y + tw \in \Omega$. ■

We would like to culminate our hierarchy of convergence results by providing necessary conditions to ensure contingent stationarity. In order to do so, we present two other types of tangent cones.

DEFINITION 3.5 (Clarke tangent cone). *A vector $v \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in the closure of Ω if for every sequence $\{y_k\}$ of elements of Ω that converges to x and for every sequence of positive real numbers $\{t_k\}$ converging to zero, there exists a sequence of vectors $\{w_k\}$ converging to v such that $y_k + t_k w_k \in \Omega$. The set $T^{Cl}_\Omega(x)$ of all Clarke tangent vectors to Ω at x is called the Clarke tangent cone to Ω at x.*

DEFINITION 3.6 (Contingent cone). *A vector $v \in \mathbb{R}^n$ is said to be a tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in the closure of Ω if there exists a sequence $\{y_k\}$ of elements of Ω that converges to x and a sequence of positive real numbers $\{\lambda_k\}$ for which $v = \lim_k \lambda_k(y_k - x)$. The set $T^{Co}_\Omega(x)$ of all tangent vectors to Ω at x is called the contingent cone (or sequential Bouligand tangent cone) to Ω at x.*

DEFINITION 3.7. *The set Ω is said to be regular at x provided $T^{Cl}_\Omega(x) = T^{Co}_\Omega(x)$.* Any convex set is regular at each of its points [8]. Both $T^{Co}_\Omega(x)$ and $T^{Cl}_\Omega(x)$ are closed cones, and both $T^{Cl}_\Omega(x)$ and $T^{H}_\Omega(x)$ are convex cones. Moreover, $T^{H}_\Omega(x) \subseteq T^{Cl}_\Omega(x) \subseteq T^{Co}_\Omega(x)$. Rockafellar [21] showed that $T^{H}_\Omega(x) = int(T^{Cl}_\Omega(x))$ whenever $T^{H}_\Omega(x)$ is nonempty.

3.3. Generalized derivatives.

Recall that we are using Jahn's definition (1.3) of the Clarke derivative instead of (1.1), and therefore we cannot directly use the calculus theory developed in [8]. The next lemma and proposition extend previously known calculus results in the unconstrained case.

LEMMA 3.8. *Let f be Lipschitz near $\hat{x} \in \Omega$ with Lipschitz constant λ. If u and v belong to $T^{H}_\Omega(\hat{x})$, then*

$$f^\circ(\hat{x};u) \;\geq\; f^\circ(\hat{x};v) - \lambda\|u - v\|.$$

Proof. Let f be Lipschitz near $\hat{x} \in \Omega$ with Lipschitz constant λ and let u and v belong to $T^{H}_\Omega(\hat{x})$. Let $\varepsilon > 0$ be such that $y + tw \in \Omega$ whenever $y \in \Omega \cap B_\varepsilon(\hat{x})$, $w \in B_\varepsilon(u) \cup B_\varepsilon(v)$ and $0 < t < \varepsilon$. This can be done by taking ε to be the smaller of the values for u and v guaranteed by the definition of a hypertangent. In particular, if $y \in \Omega \cap B_\varepsilon(\hat{x})$ and if $0 < t < \varepsilon$, then both $y + tu$ and $y + tv$ belong to Ω. This allows us to go from the first to the second equality of the following chain:

$$\begin{aligned}
f^\circ(\hat{x};u) &= \limsup_{\substack{y \to \hat{x},\, y \in \Omega \\ t \downarrow 0,\, y+tu \in \Omega}} \frac{f(y+tu)-f(y)}{t} = \limsup_{\substack{y \to \hat{x},\, y \in \Omega \\ t \downarrow 0,\, y+tv \in \Omega}} \frac{f(y+tu)-f(y)}{t} \\
&= \limsup_{\substack{y \to \hat{x},\, y \in \Omega \\ t \downarrow 0,\, y+tv \in \Omega}} \frac{f(y+tv)-f(y)}{t} + \frac{f(y+tu)-f(y+tv)}{t} \\
&= f^\circ(\hat{x};v) + \limsup_{\substack{y \to \hat{x},\, y \in \Omega \\ t \downarrow 0,\, y+tv \in \Omega}} \frac{f(y+tu)-f(y+tv)}{t} \;\geq\; f^\circ(\hat{x};v) - \lambda\|u-v\|.
\end{aligned}$$

■

Based on the previous lemma, the next proposition shows that the Clarke generalized derivative is continuous with respect to v on the Clarke tangent cone. The result is necessary to the proofs of Theorems 3.12 and 3.13.

PROPOSITION 3.9. *Let f be Lipschitz near $\hat{x} \in \Omega$. If $T^{H}_\Omega(\hat{x}) \neq \emptyset$ and if $v \in T^{Cl}_\Omega(\hat{x})$ then*

$$f^\circ(\hat{x};v) \;=\; \lim_{\substack{w \to v, \\ w \in T^{H}_\Omega(\hat{x})}} f^\circ(\hat{x};w).$$

Proof. Let λ be a Lipschitz constant for f near $\hat{x} \in \Omega$ and let $\{w_k\} \subset T_\Omega^H(\hat{x})$ be a sequence of directions converging to a vector $v \in T_\Omega^{Cl}(\hat{x})$. By definition of the hypertangent cone, let $0 < \varepsilon_k < \frac{1}{k}$ be such that

$$y + tw \in \Omega \text{ whenever } y \in \Omega \cap B_{\varepsilon_k}(\hat{x}), w \in B_{\varepsilon_k}(w_k) \text{ and } 0 < t < \varepsilon_k. \tag{3.4}$$

We first show the inequality $f^\circ(\hat{x}; v) \le \lim_k f^\circ(\hat{x}; w_k)$. Equation (3.4) implies that

$$\begin{aligned} f^\circ(\hat{x};v) \;=\; \limsup_{\substack{y \to \hat{x},\ y \in \Omega \\ t \downarrow 0,\ y+tv \in \Omega}} \frac{f(y+tv)-f(y)}{t} &= \limsup_{\substack{y \to \hat{x},\ y \in \Omega \\ t \downarrow 0,\ y+tv \in \Omega \\ y+tw_k \in \Omega}} \frac{f(y+tv)-f(y)}{t} \\ &\le \limsup_{\substack{y \to \hat{x},\ y \in \Omega \\ t \downarrow 0,\ y+tw_k \in \Omega}} \frac{f(y+tw_k)-f(y)}{t} - \frac{f(y+tw_k)-f(y+tv)}{t} \\ &= f^\circ(\hat{x};w_k) \;+\; \limsup_{\substack{y \to \hat{x},\ y \in \Omega \\ t \downarrow 0,\ y+tw_k \in \Omega}} \frac{f(y+tw_k)-f(y+tv)}{t}. \end{aligned}$$

As k goes to infinity, $\left|\frac{f(y+tw_k)-f(y+tv)}{t}\right| \le \lambda \|w_k - v\|$ goes to zero. Since $\{w_k\}$ was arbitrary in the hypertangent cone, it follows that

$$f^\circ(\hat{x};v) \;\le\; \lim_{\substack{w \to v, \\ w \in T_\Omega^H(\hat{x})}} f^\circ(\hat{x};w).$$

Second, we show the reverse inequality: $f^\circ(\hat{x}; v) \ge \lim_k f^\circ(\hat{x}; w_k)$. Let us define $u_k = \frac{1}{k} w_k + (1 - \frac{1}{k})v = w_k + (1 - \frac{1}{k})(v - w_k)$. Since the hypertangent cone is a convex set, and since v lies in the closure of the hypertangent cone, then it follows that $u_k \in T_\Omega^H(\hat{x})$ for every $k = 1, 2, \ldots$

We now consider the generalized directional derivative

$$f^\circ(\hat{x};u_k) = \limsup_{\substack{y \to \hat{x},\ y \in \Omega \\ t \downarrow 0,\ y+tu_k \in \Omega}} \frac{f(y+tu_k)-f(y)}{t}.$$

The fact that $u_k \in T_\Omega^H(\hat{x})$ implies that there exists $y_k \in \Omega \cap B_{\varepsilon_k}(\hat{x})$ and $0 < \frac{t_k}{k} < \varepsilon_k$ such that $y_k + t_k u_k \in \Omega$ and

$$\frac{f(y_k + t_k u_k) - f(y_k)}{t_k} \ge f^\circ(\hat{x}; u_k) - \varepsilon_k, \tag{3.5}$$

where ε_k is the constant from equation (3.4). We now define the sequence $z_k = y_k + \frac{t_k}{k} w_k \in \Omega$ converging to $\hat{x}$, and the sequence of scalars $h_k = (1 - \frac{1}{k})t_k > 0$ converging to zero. Notice that

$$z_k + h_k v = y_k + t_k \left(\frac{1}{k} w_k + (1 - \frac{1}{k}) v \right) \;=\; y_k + t_k u_k \;\in\; \Omega,$$

and therefore

$$\begin{aligned}
f^\circ(\hat{x};v) \;=\; \limsup_{\substack{z \to \hat{x},\; z \in \Omega \\ h \downarrow 0,\; z+hv \in \Omega}} \tfrac{f(z+hv)-f(z)}{t} &\geq \lim_k \tfrac{f(z_k+h_k v)-f(z_k)}{h_k} \\
&= \lim_k \tfrac{f(y_k+t_k u_k)-f(y_k)}{(1-\frac{1}{k})t_k} + \tfrac{f(y_k)-f(y_k+\frac{t_k}{k}w_k)}{(1-\frac{1}{k})t_k} \\
\text{by equation (3.5):} \quad &\geq \lim_k f^\circ(\hat{x};u_k) - \varepsilon_k + \tfrac{f(y_k)-f(y_k+\frac{t_k}{k}w_k)}{(1-\frac{1}{k})t_k} \\
\text{by Lemma 3.8:} \quad &\geq \lim_k f^\circ(\hat{x};w_k) - \lambda\|u_k-w_k\| - \varepsilon_k + \tfrac{\frac{\lambda}{k}\|w_k\|}{(1-\frac{1}{k})} \\
&= \lim_k f^\circ(\hat{x};w_k) - \lambda\|v-w_k\| + \tfrac{1}{k}\|v\| \;=\; f^\circ(\hat{x};w_k).
\end{aligned}$$

■

Unfortunately, the above Proposition is not necessarily true when the hypertangent cone is empty: $f^\circ(\hat{x};v)$ may differ from $\lim_{w\to v} f^\circ(\hat{x};w)$. The above proof breaks as we cannot show in Equation (3.4) that $y+tw_k$ belongs to Ω when $y\in\Omega$ is close to $\hat{x}$ and when $t>0$ is small. The following example in $\mathbb{R}^2$ illustrates that in this case, the Clarke generalized derivative is not necessarily upper semi-continuous on the contingent cone.

EXAMPLE 3.10. *Consider a feasible region $\Omega \subset \mathbb{R}^2$ that is the union of*

$$\Omega_1 = \{(a,b)^T : a \geq 0,\ b \geq 0\} \ \textit{ with } \ \Omega_2 = \{(-a,b)^T : b = -a^2,\ a \geq 0\}.$$

One can verify that at the origin

$$T^H_\Omega(0) = \emptyset,\ \ T^{Cl}_\Omega(0) = \{(a,0)^T : a \geq 0\} \subset \Omega_1 \quad \textit{and} \quad T^{Co}_\Omega(0) = \Omega_1 \cup \{(-a,0)^t : a \geq 0\},$$

and therefore Ω is not regular at the origin.

Consider the continuous concave function in $\mathbb{R}^2$: $f(a,b) = -\max\{0,a\}$. Notice that $f(a,b)=0$ for $(a,b)^T\in\Omega_2$, and $f(a,b) = -a \leq 0$ on Ω_1. We will show that $f^\circ(0;w)$ is nonnegative for w in the interior of the contingent cone but $f^\circ(0;e_1) = -1$ with $e_1 = (1,0)^T$ in the Clarke tangent cone.

Let $w = (w_1,w_2)^T$ be any direction in $int(T^{Co}_\Omega(0)) = int(\Omega_1)$. We will construct appropriate subsequences in order to compute a valid lower bound on $f^\circ(0;w)$. For every positive integer k, define

$$y_k = \left(\frac{-w_1}{k},\ \frac{-w_1^2}{k^2}\right)^T \ \textit{ and } \ t_k = \frac{1}{k}.$$

One can easily check that $y_k\in\Omega_2\subset\Omega$, and hence $f(y_k)=0$ for every k. Also, for every $k > \frac{w_1^2}{w_2}$, we have $y_k + t_k w = \left(0,\ \frac{1}{k^2}(kw_2 - w_1^2)\right)^T \in \Omega_1 \subset \Omega$ is on the nonnegative b axis. It follows that $f(y_k+t_kw)=0$ for every such k, and so

$$f^\circ(0;w) \geq \lim_{k\to\infty} \tfrac{f(y_k+t_kw)-f(y_k)}{t_k} \;=\; \lim_{k\to\infty} k\cdot(0-0) \;=\; 0.$$

In particular, taking $w=(1,\varepsilon)$, we have that $f^\circ(0;(1,\varepsilon)^T)$ is nonnegative for any $\varepsilon>0$.

However, let us compute the Clarke generalized directional derivative $f^\circ(0;e_1)$ at the origin in the direction $e_1 = (1,0)^T \in T^{Cl}_\Omega(0)$. The origin cannot be approached by points $y_k = (a_k,b_k)^T\in\Omega$ with the *properties that $b_k < 0$, and $y_k + t_k e_1 \in \Omega$ with $t_k > 0$. This is easy to see from a picture because y_k would have to be in Ω_2, and then $y_k + t_k e_1$ cannot possibly be in Ω. A necessary condition for both sequences to be in Ω is that y_k belongs to Ω_1, where $f(a,b) = -a$. But then every difference quotient in the definition of $f^\circ(0;e_1)$ is -1, and therefore $f^\circ(0;e_1) = -1$.*

This example shows that when the hypertangent cone at $\hat{x}$ is empty, but the Clarke tangent cone is not, it is possible that $f^\circ(\hat{x};w)$ is nonnegative for every w in the interior of the contingent cone and drops discontinuously to a negative value on the boundary of the contingent cone: $f^\circ(\hat{x};e_1) < \limsup_{w\to e_1} f^\circ(\hat{x};w)$.

3.4. A hierarchy of convergence results for MADS. We now present different necessary optimality conditions based on the tangent cone definitions.

DEFINITION 3.11. *Let f be Lipschitz near $\hat{x} \in \Omega$. Then, $\hat{x}$ is said to be a Clarke, or contingent, stationary point of f over Ω, if $f^\circ(\hat{x};v) \geq 0$ for every direction v in the Clarke, or contingent, cone of Ω at $\hat{x}$, respectively.*

In addition, $\hat{x}$ is said to be a Clarke, or contingent KKT stationary point of f over Ω, if $-\nabla f(\hat{x})$ exists and belongs to the polar of the Clarke, or contingent cone of Ω at $\hat{x}$, respectively.

This leads to our basic result on refining directions from which all our hierarchy of results are derived. The proof of this results also illustrates the close connection between the MADS framework, the Clarke calculus and the definition of a hypertangent vector.

THEOREM 3.12. *Let f be Lipschitz near a limit $\hat{x} \in \Omega$ of a refining subsequence, and let $v \in T^H_\Omega(\hat{x})$ be a refining direction for $\hat{x}$. Then the generalized directional derivative of f at $\hat{x}$ in the direction v is nonnegative,* i.e., $f^\circ(\hat{x};v) \geq 0$.

Proof. Let $\{x_k\}_{k\in K}$ be a refining subsequence converging to $\hat{x}$ and $v = \lim_{k\in L} \frac{d_k}{\|d_k\|} \in T^H_\Omega(\hat{x})$ be a refining direction for $\hat{x}$, with $d_k \in D_k$ for every $k \in L$. Since f is Lipschitz near $\hat{x}$, Proposition 3.9 ensures that $f^\circ(\hat{x};v) = \lim_{k\in L} f^\circ(\hat{x};\frac{d_k}{\|d_k\|})$. But, for any $k \in L$, one can apply the definition of the Clarke generalized derivative with the roles of y and t played by x_k and $\Delta^m_k\|d_k\|$, respectively. Note that this last quantity indeed converges to zero since Definition 2.2 ensures that it is bounded above by $\Delta^p_k \max\{\|d'\| : d' \in D\}$, where D is a finite set of directions, and Equation (2.2) states that Δ^p_k goes to zero. Therefore

$$f^\circ(\hat{x};v) \ \geq\ \limsup_{k\in L} \frac{f(x_k+\Delta^m_k\|d_k\|\frac{d_k}{\|d_k\|})-f(x_k)}{\Delta^m_k\|d_k\|} \ =\ \limsup_{k\in L} \frac{f(x_k+\Delta^m_k d_k)-f(x_k)}{\Delta^m_k\|d_k\|} \ \geq\ 0.$$

The last inequality follows from the fact that for each sufficiently large $k \in L$, $x_k + \Delta^m_k d_k \in \Omega$ and $f(x_k + \Delta^m_k d_k) = f_\Omega(x_k + \Delta^m_k d_k)$ was evaluated and compared by the algorithm to $f(x_k)$, but x_k is a minimal frame center, so the inequality holds. ■

We now show that Clarke directional derivatives of f at the limit $\hat{x}$ of minimal frame centers, for meshes that get infinitely fine, are nonnegative for all directions in the hypertangent cone, i.e., we show that MADS generates a Clarke stationary point.

THEOREM 3.13. *Let f be Lipschitz near a limit $\hat{x} \in \Omega$ of a refining subsequence, and assume that $T^H_\Omega(\hat{x}) \neq \emptyset$. If the set of refining directions for $\hat{x}$ is dense in $T^H_\Omega(\hat{x})$, then $\hat{x}$ is a Clarke stationary point of f on Ω.*

Proof. The proof follows directly from Theorem 3.12 and Proposition 3.9. ■

Note that even though the algorithm is applied to f_Ω instead of f, the convergence results are linked to the local smoothness of f and not f_Ω, which is obviously discontinuous on the boundary of Ω. This is because we use (1.3) as the definition of the Clarke generalized derivative instead of (1.1). The constraint qualification used in these results is that the hypertangent cone is non-empty at the feasible limit point $\hat{x}$. Further discussion on non-empty hypertangent cones is found in Rockafellar [21].

A corollary to this last theorem is that if f is strictly differentiable at $\hat{x}$, then it is a Clarke KKT point.

COROLLARY 3.14. *Let f be strictly differentiable at a limit $\hat{x} \in \Omega$ of a refining subsequence, and assume that $T_\Omega^H(\hat{x}) \neq \emptyset$. If the set of refining directions for $\hat{x}$ is dense in $T_\Omega^H(\hat{x})$, then $\hat{x}$ is a Clarke KKT stationary point of f over Ω.*

Proof. Strict differentiability ensures that the gradient $\nabla f(\hat{x})$ exists and that $\nabla f(\hat{x})^T v = f^\circ(\hat{x}; v)$ for all directions. It follows directly from the previous proposition that $-\nabla f(\hat{x})^T v \leq 0$ for every direction v in $T_\Omega^{Cl}(\hat{x})$, thus $\hat{x}$ is a Clarke KKT stationary point. ■

Our next two results are based on the definition of set regularity (see Definition 3.7).

PROPOSITION 3.15. *Let f be Lipschitz near a limit $\hat{x} \in \Omega$ of a refining subsequence, and assume that $T_\Omega^H(\hat{x}) \neq \emptyset$. If the set of refining directions for $\hat{x}$ is dense in $T_\Omega^H(\hat{x})$, and if Ω is regular at $\hat{x}$, then $\hat{x}$ is a contingent stationary point of f over Ω.*

Proof. The definition of regularity of the set Ω ensures that $f^\circ(\hat{x}; w) \geq 0$ for all w in $T_\Omega^{Co}(\hat{x})$. ■

The following result is the counterpart to Corollary 3.14 for contingent stationarity. The proof is omitted since it is essentially the same.

COROLLARY 3.16. *Let f be strictly differentiable at a limit $\hat{x} \in \Omega$ of a refining subsequence, and assume that $T_\Omega^H(\hat{x}) \neq \emptyset$. If the set of refining directions for $\hat{x}$ is dense in $T_\Omega^H(\hat{x})$, and if Ω is regular at $\hat{x}$, then $\hat{x}$ is a contingent KKT stationary point of f over Ω.*

Example F in [2] presents an instance of a GPS algorithm such that when applied to a given unconstrained optimization problem, it generates a single limit point $\hat{x}$ which is not a Clarke stationary point. In fact, it is shown that f is differentiable but not strictly differentiable at $\hat{x}$ and $\nabla f(\hat{x})$ is nonzero. This unfortunate circumstance is due to the fact that GPS uses a finite number of poll directions. MADS can use infinitely many.

The following result shows that the algorithm ensures strong optimality conditions for unconstrained optimization, or when $\hat{x}$ is in the interior of Ω.

THEOREM 3.17. *Let f be Lipschitz near a limit $\hat{x}$ of a refining subsequence. If $\Omega = \mathbb{R}^n$, or if $\hat{x} \in int(\Omega)$, and if the set of refining directions for $\hat{x}$ is dense in $\mathbb{R}^n$, then $0 \in \partial f(\hat{x})$.*

Proof. Let $\hat{x}$ be as in the statement of the result, then $T_\Omega^H(\hat{x}) = \mathbb{R}^n$. Combining Definition 3.11 and Theorem 3.13 with equation (1.2) yields the result. ■

Newton's method uses first derivatives, and the standard analysis of Newton's method assumes Lipschitz continuity of the first derivatives. Correspondingly, MADS is an algorithm that uses only function values, and we assume only that the function f is Lipschitz near $\hat{x}$.

The SQP method, which is a constrained analog of Newton's method, makes an assumption on the constraints such as linear independence of the constraint gradients at a limit point of the iteration. Such assumptions are called constraint qualifications, and they are meant to ensure that the limit points produced by the algorithm satisfy necessary conditions for optimality. The weaker constraint qualification we need for MADS is that the hypertangent cone is nonempty at the limit point of a refining sequence produced by MADS.

To ensure that the algorithm produces a limit point satisfying necessary conditions, SQP makes assumptions on the initial guess and the method for choosing the next iterate. So far, we have made the main algorithmic assumption that the set of refining directions is dense in $T_\Omega^H(\hat{x})$. Under these conditions, we guarantee a Clarke stationary point $\hat{x}$. In a paper on SQP, it is common to suggest practical algorithmic procedures to achieve the algorithmic requirements for convergence to a stationary point. That is our next topic for MADS.

In the general statement of the algorithm we did not present a strategy that would guarantee a dense set of refining directions in the hypertangent cone. We want to keep the algorithm framework as general as possible. There are different strategies that could be used to generate a dense set of poll directions. The selection of the set D_k could be done in a deterministic way or may use some randomness. In the remainder of the paper, we present, analyze, and test one MADS strategy that uses some randomness. We do this because we have not found a deterministic strategy that achieves a good distribution of poll directions when the process is terminated after a reasonable number of iterations.

4. Practical implementation – LTMADS. We now present two variants of a stochastic implementation of the MADS algorithm. We call either variant LTMADS, because of the underlying lower triangular basis construction, and we show that with probability 1, the set of poll directions generated by the algorithm is dense in the whole space, and in particular in the hypertangent cone.

4.1. Implementable instances of a MADS algorithm. Let $G = I$, the identity matrix, and let $D = [I \ -I], \tau = 4, w^- = -1$ and $w^+ = 1$ be the fixed algorithmic parameters. Choose $\Delta_0^m = 1, \Delta_0^p = 1$ to be the initial mesh and poll size parameters, and define the update rules as follows:

$$\Delta_{k+1}^m = \begin{cases} \frac{\Delta_k^m}{4} & \text{if } x_k \text{ is a minimal frame center} \\ 4\Delta_k^m & \text{if an improved mesh point is found, and if } \Delta_k^m \leq \frac{1}{4} \\ \Delta_k^m & \text{otherwise.} \end{cases}$$

A consequence of these rules is that the mesh size parameter is always a power of 4 and never exceeds 1. Thus, $\frac{1}{\sqrt{\Delta_k^m}} \geq 1$ is always a nonnegative power of 2 and hence integral.

We now present a strategy to randomly generate the poll directions. In what follows, every random generation is done uniformly with equal probabilities. In order to ensure that the set of refining directions is dense in the hypertangent cone, one of these directions must be selected in a different way. This direction must depend only on the value of the mesh size parameter, and not on the iteration number. The direction is denoted by $b(\ell)$ where ℓ is an integer related to the mesh size parameter. An additional counter, called ℓ_c is initially set to zero. It is used to keep track of the values of ℓ for which $b(\ell)$ was created. The construction of $b(\ell)$ is as follows.

GENERATION OF THE DIRECTION $b(\ell)$ FOR A GIVEN NONNEGATIVE INTEGER ℓ.

- VERIFICATION IF $b(\ell)$ WAS ALREADY CREATED:
  If $\ell_c > \ell$, then exit this procedure with the existing vector $b(\ell) \in \mathbb{Z}^n$.
  Otherwise, set $\ell_c \leftarrow \ell_c + 1$, and continue to the next step.
- INDEX OF ENTRY WITH LARGEST COMPONENT:
  Let $\hat{\imath}$ be an integer randomly chosen in the set $N = \{1, 2, \ldots, n\}$.
- CONSTRUCTION OF $b(\ell)$:
  Randomly set $b_{\hat{\imath}}(\ell)$ to either plus or minus 2^ℓ, and $b_i(\ell)$ for $i \in N \setminus \{\hat{\imath}\}$ to be an integer in $\{-2^\ell + 1, -2^\ell + 2, \ldots, 2^\ell - 1\}$. Record $b(\ell)$ and exit this procedure.

The above procedure returns a vector $b(\ell) \in \mathbb{Z}^n$ such that all elements but one are integers between $-2^\ell + 1$ and $2^\ell - 1$. The other element is either -2^ℓ or 2^ℓ. Moreover, when two iterations have the same mesh size parameter, then the corresponding vectors $b(\ell)$ are identical.

To each mesh size parameter Δ_k^m, we assign an integer $\ell = -\log_4(\Delta_k^m) \in \mathbb{N}$ so that $\Delta_k^m = 4^{-\ell}$. Note that the mesh size parameter in LTMADS takes the values $1, \frac{1}{4}, \frac{1}{16}, \ldots$, and therefore ℓ is necessarily a nonnegative integer.

We now present a procedure that extends $b(\ell)$ to a positive spanning set of either $2n$ or $n+1$ poll directions. The procedure first generates an $(n-1) \times (n-1)$ lower triangular matrix, and then combines it with $b(\ell)$ to create a basis in $\mathbb{R}^n$. Finally, this basis is extended to a positive basis by either mirroring the directions (for a maximal $2n$ basis), or by taking the negative sum of the directions (for a $n+1$ basis).

GENERATION OF THE POSITIVE BASIS D_k AND UPDATE OF Δ_k^p.

- CONSTRUCTION OF THE DIRECTION $b(\ell)$ AND INDEX $\hat{\imath}$:
  Let $\ell = -\log_4(\Delta_k^m)$, and construct $b(\ell)$ by the above procedure.
  Set $\hat{\imath}$ to be the integer in N such that $|b_{\hat{\imath}}(\ell)| = 2^\ell$.
- BASIS CONSTRUCTION IN $\mathbb{R}^{n-1}$:
  Let L be a lower triangular $(n-1)\times(n-1)$ matrix where each term on the diagonal is either plus or minus 2^ℓ, and the lower components are randomly chosen in $\{-2^\ell+1, -2^\ell+2, \ldots, 2^\ell-1\}$.
  L is a basis in $\mathbb{R}^{n-1}$ with $|\det(L)| = 2^{\ell(n-1)}$.
- PERMUTATION OF THE LINES OF L, AND COMPLETION TO A BASIS IN $\mathbb{R}^n$:
  Let $\{p_1, p_2, \ldots, p_{n-1}\}$ be random permutations of the set $N \setminus \{\hat{\imath}\}$. Set

$$\begin{array}{ll} B_{p_i,j} = L_{i,j} & \text{for } i,j = 1,2,\ldots,n-1 \\ B_{\hat{\imath},j} = 0 & \text{for } j = 1,2,\ldots,n-1 \\ B_{i,n} = b_i(\ell) & \text{for } i = 1,2,\ldots,n. \end{array}$$

  B is a basis in $\mathbb{R}^n$ with $|\det(B)| = 2^{\ell n}$.
- PERMUTATION OF THE COLUMNS OF B:
  Let $\{q_1, q_2, \ldots, q_n\}$ be random permutations of the set N.
  Set $B'_{i,q_j} = B_{i,j}$ for each i and j in N. B' is a basis in $\mathbb{R}^n$ with $|\det(B')| = 2^{\ell n}$.
- COMPLETION TO A POSITIVE BASIS:
  - Minimal positive basis: Set $D_k = [B' \; d]$ with $d_i = -\sum_{j\in N} B'_{ij}$.
    Set the poll size parameter to $\Delta_k^p = n\sqrt{\Delta_k^m} \geq \Delta_k^m$.
  - Maximal positive basis: Set $D_k = [B' \; -B']$.
    Set the poll size parameter to $\Delta_k^p = \sqrt{\Delta_k^m} \geq \Delta_k^m$.

The rows of a lower triangular matrix L are randomly permuted, and a line of zeroes is inserted in position $\hat{\imath}$. This results in a $n\times(n-1)$ matrix. The column $b(\ell)$ is appended to it, and this leads to a basis B in $\mathbb{R}^n$. The permutation of the rows ensures that the zeroes of the triangular matrix are not mostly located in the upper part of B. Afterwards, the columns of B are randomly permuted to ensure that the zeroes are not mostly located in the right part of B'. This construction ensures that $|\det(B)| = |\det(B')| = 2^{\ell n}$. The completion to a positive basis D_k appends to B' either the negative sum of the columns of B', or the negative of each column.

The construction also ensures that $b(\ell)$ is necessarily a column of the positive basis D_k. Our convergence analysis will show that as k goes to infinity, the union of all directions $b(\ell)$ is dense in $\mathbb{R}^n$ with probability one. We will also show that if the entire sequence of iterates converges, then the set of refining directions is also dense in $\mathbb{R}^n$ with probability one.

We now present an example in $\mathbb{R}^5$ to highlight the features of the positive basis construction.

EXAMPLE 4.1. *Consider an iteration k with $\Delta_k^m = \frac{1}{16}$. The step* CONSTRUCTION OF THE DIRECTION $b(\ell)$ AND INDEX $\hat{\imath}$ *fixed $\ell = -\log_4(\Delta_k^m) = 2$. Suppose that the randomly defined vector $b(\ell)$ is $(-3,2,4,-1,0)^T$. It follows that $\hat{\imath} = 3$ since $b_3(\ell) = 4$. Observe that all other components of $b(\ell)$ are integers between -2^2+1 and 2^2-1.*

Suppose that the step BASIS CONSTRUCTION IN $\mathbb{R}^{n-1}$ *generates the random lower triangular matrix*

$$L = \begin{bmatrix} -4 & 0 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ -1 & 2 & -4 & 0 \\ 1 & -2 & 0 & 4 \end{bmatrix} \in \mathbb{Z}^{4\times 4}.$$

Now, if the two permutation steps generate the line permutation vector $(p_1, p_2, p_3, p_4) = (4, 1, 2, 5)$*, and the column permutation vector* $(q_1, q_2, q_3, q_4, q_5) = (5, 1, 3, 2, 4)$ *then the bases constructed from L and* $b(\ell)$ *are*

$$B = \begin{bmatrix} \mathbf{3} & \mathbf{4} & \mathbf{0} & \mathbf{0} & -3 \\ \mathbf{-1} & \mathbf{2} & \mathbf{-4} & \mathbf{0} & 2 \\ 0 & 0 & 0 & 0 & 4 \\ \mathbf{-4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -1 \\ \mathbf{1} & \mathbf{-2} & \mathbf{0} & \mathbf{4} & 0 \end{bmatrix} \quad \text{and } B' = \begin{bmatrix} \mathbf{4} & \mathbf{0} & \mathbf{0} & -3 & \mathbf{3} \\ \mathbf{2} & \mathbf{0} & \mathbf{-4} & 2 & \mathbf{-1} \\ 0 & 0 & 0 & 4 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -1 & \mathbf{-4} \\ \mathbf{-2} & \mathbf{4} & \mathbf{0} & 0 & \mathbf{1} \end{bmatrix}$$

(the entries copied from L appear in boldface characters). One may easily verify that $|\det(B)| = |\det(B')| = 4^5$ *and that the four terms* B'_{p_i,q_i} *for* $i = 1, 2, 3, 4$ *as well as* B'_{3,q_5} *are equal to either* 4 *or* -4*.*

Finally, depending on if the minimal or maximal positive basis is selected, the COMPLETION TO A POSITIVE BASIS *step generates the set* D_k *composed of the columns of either*

$$\begin{bmatrix} 4 & 0 & 0 & -3 & 3 & -4 \\ 2 & 0 & -4 & 2 & -1 & 1 \\ 0 & 0 & 0 & 4 & 0 & -4 \\ 0 & 0 & 0 & -1 & -4 & 5 \\ -2 & 4 & 0 & 0 & 1 & -3 \end{bmatrix} \quad or \quad \begin{bmatrix} 4 & 0 & 0 & -3 & 3 & -4 & 0 & 0 & 3 & -3 \\ 2 & 0 & -4 & 2 & -1 & -2 & 0 & 4 & -2 & 1 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -1 & -4 & 0 & 0 & 0 & 1 & 4 \\ -2 & 4 & 0 & 0 & 1 & 2 & -4 & 0 & 0 & -1 \end{bmatrix}.$$

A key point of this construction is that any iteration with a mesh size parameter equal to $\frac{1}{16}$ *will have* $b(\ell)$ *as the* q_5^{th} *column of* D_k*. In this particular example,* $b(\ell)$ *is the fourth column of* D_k*. The other columns will usually differ from one iteration to another.*

Since MADS is allowed to be *opportunistic* and end a POLL step as soon as a better point is found, we want to randomize the POLL directions. Thus, the purpose of the second step is to permute the lines of the matrix B so that the zeroes in the upper triangular part of the matrix are randomly positioned, and to permute the columns so that the dense column is not always the first in D_k. The name LTMADS is based on the lower triangular matrix at the heart of the construction of the frames.

The following result shows that the frames generated by the LTMADS algorithm satisfy the conditions of Definition 2.2.

PROPOSITION 4.2. *At each iteration k, the procedure above yields a* D_k *and a MADS frame* P_k *such that:*

$$P_k = \{x_k + \Delta_k^m d : d \in D_k\} \subset M_k ,$$

where M_k *is given by Definition 2.1 and* D_k *is a positive spanning set such that for each* $d \in D_k$*,*

- *d can be written as a nonnegative integer combination of the directions in D:* $d = Du$ *for some vector* $u \in \mathbb{N}^{n_D}$ *that may depend on the iteration number k*
- *the distance from the frame center* x_k *to a frame point* $x_k + \Delta_k^m d \in P_k$ *is bounded by a constant times the poll size parameter:* $\Delta_k^m \|d\| \le \Delta_k^p \max\{\|d'\| : d' \in D\}$

- *limits (as defined in Coope and Price [9]) of the normalized sets D_k are positive spanning sets.*

Proof. The first n columns of D_k form a basis of $\mathbb{R}^n$ because they are obtained by permuting rows and columns of the lower triangular matrix B, which is nonsingular because it has nonzero terms on the diagonal. Moreover, taking the last direction to be the negative of the sum of the others leads to a minimal positive basis, and combining the first n columns of D_k with their negatives gives a maximal positive basis [11].

Again by construction, D_k has all integral entries in the interval $[-2^\ell, 2^\ell]$ (with $2^\ell = \frac{1}{\sqrt{\Delta_k^m}}$), and so clearly each column d of D_k can be written as a nonnegative integer combination of the columns of $D = [I, -I]$. Hence, the frame defined by D_k is on the mesh M_k.

Now the ℓ_∞ distance from the frame center to any frame point is $\|\Delta_k^m d\|_\infty = \Delta_k^m \|d\|_\infty$. There are two cases. If the maximal positive basis construction is used, then $\Delta_k^m \|d\|_\infty = \sqrt{\Delta_k^m} = \Delta_k^p$. If the minimal positive basis construction is used, then $\Delta_k^m \|d\|_\infty \le n\sqrt{\Delta_k^m} = \Delta_k^p$. The proof of the second bullet follows by noticing that $\max\{\|d'\|_\infty : d' \in [I\ -I]\} = 1$.

The frame can be rewritten in the equivalent form $\{x_k + \sqrt{\Delta_k^m} v : v \in \mathcal{V}\}$ where $\mathcal{V}$ is a set whose columns are the same as those of B after permutation and multiplication by $\sqrt{\Delta_k^m}$.

Coope and Price [9] show that a sufficient condition for the third bullet to hold is that each element of $\mathcal{V}$ is bounded above and below by positive constants that are independent of k. This is trivial to show with our construction. Indeed, each entry of $\mathcal{V}$ lies between -1 and 1 and every term on the diagonal is ± 1. B is a triangular matrix, and therefore $|det(\mathcal{V})| = 1$. ■

The frames given in Figure 2.2 were generated using minimal positive bases with directions sets D_k: $\{(-1,0)^T, (0,-1)^T, (1,1)^T\}$, $\{(-2,-1)^T, (0,-2)^T, (2,3)^T\}$ and $\{(-3,4)^T, (4,0)^T, (-1,-4)^T\}$. One can see that as Δ_k^m and Δ_k^p go to zero, the number of candidates for frame points increases rapidly. For the example illustrated in the figure, In the rightmost figure, there are a total of 56 distinct possible frames that MADS may choose from. The first line of the matrix B is $[\pm 4\ 0]$, the second line is $[B_{21}\ \pm 4]$ where B_{21} is an integer between -3 and 3. It follows that there are $2 \times 2 \times 7 = 28$ possibilities for B. Permuting the lines doubles the number of possibilities to 56. Permuting the columns does not change the points in the frames.

In addition to an opportunistic strategy, *i.e.*, terminating a POLL step as soon as an improved mesh point is detected, a standard trick we use in GPS to improve the convergence speed consists in promoting a successful poll direction to the top of the list of directions for the next POLL step. We call this *dynamic ordering* of the polling directions. This strategy can not be directly implemented in MADS since at a successful iteration $k-1$, the poll size parameter is increased, and therefore a step of Δ_k^m in the successful direction will often be outside the mesh. The way we mimic GPS dynamic ordering in MADS is that when the previous iteration succeeded in finding an improved mesh point, we execute a simple one point *dynamic search* in the next iteration as follows. Suppose that $f_\Omega(x_k) < f_\Omega(x_{k-1})$ and that d is the direction for which $x_k = x_{k-1} + \Delta_{k-1}^m d$. Then, the trial point produced by the SEARCH step is $s_k = x_{k-1} + 4\Delta_{k-1}^m d$. Note that with this construction, if $\Delta_{k-1}^m < 1$, then $s_k = x_{k-1} + \Delta_k^m d$ and otherwise, $s_k = x_{k-1} + 4\Delta_k^m d$. In both cases s_k lies on the current mesh M_k. If this SEARCH finds a better point, then we go on to the next iteration, but if not, then we proceed to the POLL step. The reader will see in the numerical results below that this seems to be a good strategy.

4.2. Convergence analysis. The convergence results in Section 3.4 are based on the assumption that the set of refining directions for the limit of a refining sequence is asymptotically dense in the hypertangent cone at that limit. The following result shows that the above instances of LTMADS generates an asymptotically dense set of poll directions with probability 1. Therefore, the convergence results based on the local smoothness of the objective function f and on the local topology of the feasible region Ω can be applied to LTMADS.

THEOREM 4.3. *Let $\hat{x} \in \Omega$ be the limit of a refining subsequence produced by either instance of LTMADS. Then the set of poll directions for the subsequence converging to $\hat{x}$ is asymptotically dense in $T^H_\Omega(\hat{x})$ with probability 1.*

Proof. Let $\hat{x}$ be the limit of a refining subsequence $\{x_k\}_{k\in K}$ produced by one of the above instances of LTMADS (either with the minimal or maximal positive basis). Consider the sequence of positive bases $\{D_k\}_{k\in K}$. Each one of these bases is generated independently.

We use the notation $P[E]$ to denote the probability that E occurs. Let v be a direction in $\mathbb{R}^n$ with $\|v\|_\infty = 1$ such that $P[|v_j| = 1] \geq \frac{1}{n}$ and $P[v_j = 1 \mid |v_j| = 1] = P[v_j = -1 \mid |v_j| = 1] = \frac{1}{2}$. We will find a lower bound on the probability that a normalized direction in D_k is arbitrarily close to the vector v.

Let k be an index of K, and let $\ell = -\log_4(\Delta^m_k)$. Recall that in the generation of the positive basis D_k, the column $b(\ell)$ is such that $|b_{\hat{\imath}}(\ell)| = 2^\ell$, and the other components of $b(\ell)$ are random integers between $-2^\ell+1$ and $2^\ell - 1$. Set $u = \frac{b(\ell)}{\|b(\ell)\|_\infty}$. It follows by construction that $u = 2^{-\ell}b(\ell)$ and $\|u\|_\infty = |u_{\hat{\imath}}| = 1$. We will now show for any $0 < \varepsilon < 1$, that the probability that $\|u - v\|_\infty < \varepsilon$ is bounded below by some nonnegative number independent of k, as $k \in K$ goes to infinity. Let us estimate the probability that $|u_j - v_j| < \varepsilon$ for each j. For $j = \hat{\imath}$ we have

$$\begin{aligned} P[|u_{\hat{\imath}} - v_{\hat{\imath}}| < \varepsilon] &\geq P[u_{\hat{\imath}} = v_{\hat{\imath}} = 1] + P[u_{\hat{\imath}} = v_{\hat{\imath}} = -1] \\ &= P[u_{\hat{\imath}} = 1] \times P[v_{\hat{\imath}} = 1] + P[u_{\hat{\imath}} = -1] \times P[v_{\hat{\imath}} = -1] \;\geq\; \frac{1}{2} \times \frac{1}{2n} + \frac{1}{2} \times \frac{1}{2n} \;=\; \frac{1}{2n}. \end{aligned}$$

For $j \in N \setminus \{\hat{\imath}\}$ we have

$$P[|u_j - v_j| < \varepsilon] = P[v_j - \varepsilon < u_j < v_j + \varepsilon] \;=\; P\left[2^\ell(v_j - \varepsilon) < b_j(\ell) < 2^\ell(v_j + \varepsilon)\right].$$

We will use the fact that the number of integers in the interval $\left[2^\ell(v_j - \varepsilon), 2^\ell(v_j + \varepsilon)\right] \cap \left[-2^\ell + 1, 2^\ell - 1\right]$ is bounded below by the value $2^\ell\varepsilon - 1$. Now, since the bases D_k are independently generated, and since $b_j(\ell)$ is an integer randomly chosen with equal probability among the $2^{\ell+1} - 1$ integers in the interval $\left[-2^\ell + 1, 2^\ell - 1\right]$, then it follows that

$$P[|u_j - v_j| < \varepsilon] \geq \frac{2^\ell\varepsilon - 1}{2^{\ell+1} - 1} \;>\; \frac{2^\ell\varepsilon - 1}{2^{\ell+1}} \;=\; \frac{\varepsilon - 2^{-\ell}}{2}.$$

Recall that $\hat{x}$ is the limit of a refining subsequence, and so, there exists an integer α such that $\sqrt{\Delta^m_k} = 2^{-\ell} \leq \frac{\varepsilon}{2}$ whenever $\alpha \leq k \in K$, and so

$$P[|u_j - v_j| < \varepsilon] \geq \frac{\varepsilon - \sqrt{\Delta^m_k}}{2} \;\geq\; \frac{\varepsilon}{4} \qquad \text{for any } k \in K \text{ with } k \geq \alpha.$$

It follows that

$$P[\|u - v\|_\infty < \varepsilon] \;=\; \prod_{j=1}^{n} P[|u_j - v_j| < \varepsilon] \geq \frac{\left(\frac{\varepsilon}{4}\right)^{n-1}}{2n} \qquad \text{for any } k \in K \text{ with } k \geq \alpha.$$

We have shown when k is sufficiently large, that $P[\|u - v\|_\infty < \varepsilon]$ is larger than a strictly positive constant which is independent of Δ^m_k. Thus, there will be a poll direction in D_k for some $k \in K$ arbitrarily close to any direction $v \in \mathbb{R}^n$, and in particular to any direction $v \in T^H_\Omega(\hat{x})$. ■

The proof of the previous result shows that the set of directions consisting of the $b(\ell)$ directions over all iterations is dense in $\mathbb{R}^n$. Nevertheless, we require the algorithm to use a positive spanning set at each iteration instead of a single poll direction. This ensures that any limit of a refining subsequence is the limit of minimal frame centers on meshes that get infinitely fine. At this limit point, the set of refining directions is generated from the set of poll directions which is dense in LTMADS and finite in GPS. Therefore with both MADS and GPS, the set of directions for which the Clarke generalized derivatives are nonnegative positively span the whole space. However, GPS does not allow the possibility that the set of refining directions is dense, since it is finite.

Finally, we give a condition that ensures dense MADS refining directions with probability 1.

THEOREM 4.4. *Suppose that the entire sequence of iterates produced by either instance of LTMADS converges to $\hat{x} \in \Omega$. Then the set of refining directions for the entire sequence of iterates is asymptotically dense in $T^H_\Omega(\hat{x})$ with probability one.*

Proof. Let K be the set of indices of iterations that are minimal frame centers. If the entire sequence of iterates produced by an instance of LTMADS converges to $\hat{x} \in \Omega$, then the subsequence $\{x_k\}_{k\in K}$ also converges to $\hat{x}$. Therefore, $\{b(\ell)\}_{\ell=1}^{\infty}$ is a subsequence of refining directions. This subsequence was shown in Theorem 4.3 to be asymptotically dense in $T^H_\Omega(\hat{x})$ with probability 1. ∎

5. Numerical results. We consider four test problems in this section. Each problem is intended to make a point about MADS. Three of the problems are artificial, and for these problems we give results for GPS with a POLL step only and with a simple randomized SEARCH . The GPS results all use a POLL ordering we have found to be advantageous in our experience using GPS.

The first problem is unconstrained, but GPS is well known to stagnate on this problem if it is given an unsuitable set of directions. MADS has no problem converging quickly to a global optimizer. The second problem is a bound constrained chemical engineering problem where GPS is known to behave well enough to justify publication of the results [14]. Still, on the whole, MADS does better. The third is a simple nonlinearly constrained problem where GPS and our filter version of GPS are both known to converge short of an optimizer. As the theory given here predicts, MADS has no difficulty. We also use this problem to show that MADS does well as the number of variables increases.

The last example is such that the feasible region gets narrow very quickly. This is meant to be a test for any derivative-free feasible point algorithm - like GPS or MADS with the extreme barrier approach to constraints. MADS does better than GPS with the filter or the barrier, both of which stagnate due to the limitation of finitely many POLL directions. MADS stops making progress when the mesh size gets smaller than the precision of the arithmetic.

Of course, even when one tries to choose carefully, four examples are not conclusive evidence. However, we believe that these numerical results coupled with the more powerful theory for MADS make a good case for MADS versus GPS.

5.1. An unconstrained problem where GPS does poorly. Consider the unconstrained optimization problem in $\mathbb{R}^2$ presented in [16] where GPS algorithms are known to converge to non-stationary points:

$$f(x) \;=\; \left(1-\exp(-\|x\|^2)\right) \times \max\{\|x-c\|^2, \|x-d\|^2\},$$

where $c = -d = (30,40)^T$. Figure 5.1 shows level sets of this function. It can be shown that f is locally Lipschitz and and strictly differentiable at its global minimizer $(0,0)^T$.

The GPS and MADS runs are initiated at $x_0 = (-2.1,1.7)^T$, depicted by a diamond in the right part of Figure 5.1. The gradient of f exists and is non-zero at that point, and therefore both GPS and MADS will move away from it. Since there is some randomness involved in the MADS instance described in Section 4.1, we ran it a total of 5 times, to see how it compares to our standard NOMAD implementation of GPS. Figure 5.2 shows a log plot of the progress of the objective function value for each set of runs. All POLL were opportunistic, and the runs were stopped when a minimal frame with poll size parameter

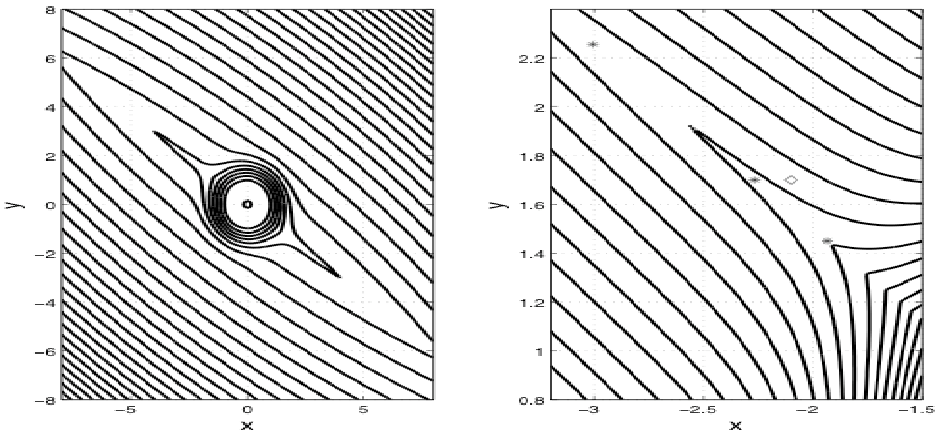


FIG. 5.1. *Level sets of* $f(x) = \left(1-\exp(-\|x\|^2)\right) \times \max\{\|x-c\|^2, \|x-d\|^2\}$.

less than 10^{-10} was detected. For GPS, the maximal $2n$ positive basis refers to the set of positive and negative coordinate directions, and the two minimal $n+1$ positive bases are $\{(1,0)^T, (0,1)^T, (-1,-1)^T\}$ and $\{(1,0)^T, (-0.5, 0.866025)^T, (-0.5, -0.866025)^T\}$.

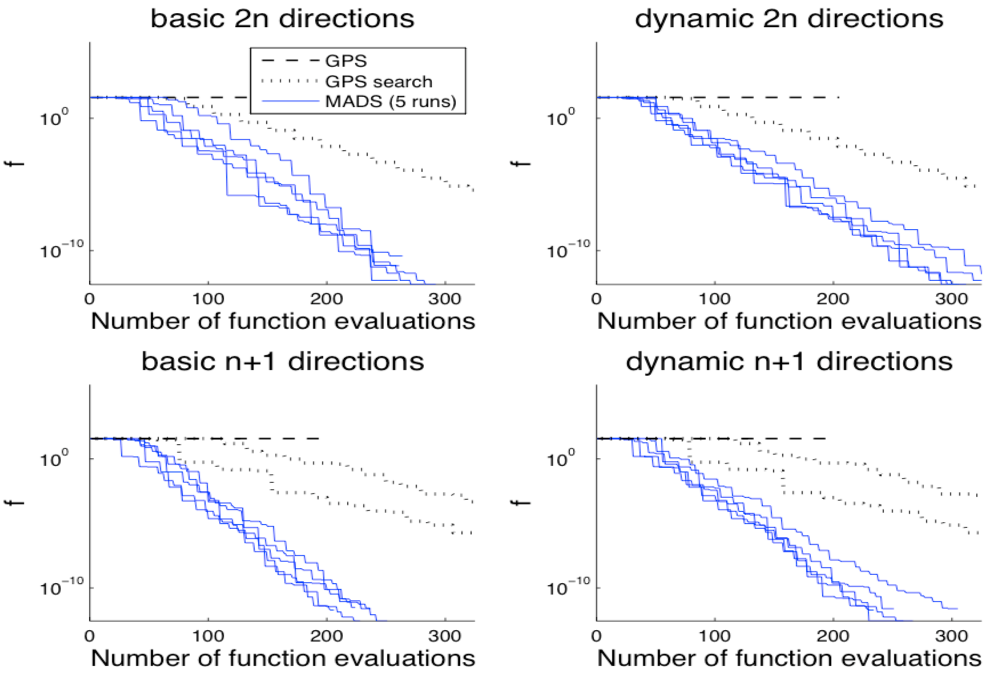


FIG. 5.2. *Progression of the objective function value vs the number of evaluations.*

Without a search strategy, every GPS run converged to a point on the line $x_2 = -\frac{3}{4}x_1$, where f is not differentiable. These three limit points are denoted by stars in Figure 5.1. As proved in [3], the limit points for GPS satisfy the necessary optimality condition that the Clarke generalized directional derivatives are nonnegative for D at these limit points, but, they are not local optimizers. One can see by looking at the level sets of f that no descent directions can be generated by the GPS algorithm using the above directions.

However, when adding a search strategy (by randomly selecting $2n$ mesh points at each SEARCH step) or when using LTMADS, all runs eventually generated good directions and converged to the origin, the global optimal solution. Figure 5.2 suggests that the MADS convergence is faster than GPS. Also, even if randomness appears in these instances of LTMADS, the behavior of the algorithm is very stable in converging quickly to the origin.

5.2. A test problem where GPS does well. The academic example above was one of our motivations for developing MADS. We now apply MADS to an example from the chemical engineering literature for which GPS was shown to be preferable to a conjugate-direction approach. Hayes *et al.* [14] describe a method for evaluating the kinetic constants in a rate expression for catalytic combustion applications using experimental light-off curves. The method uses a transient one-dimensional single channel monolith finite element reactor model to simulate reactor performance. The objective is to find the values of four parameters in a way such that the model estimates as closely as possible (in a weighted least square sense) an experimental conversion rate. This is a bound constrained nonsmooth optimization problem in $\mathbb{R}^4_+$, where the objective function measures the error between experimental data and values predicted by the model.

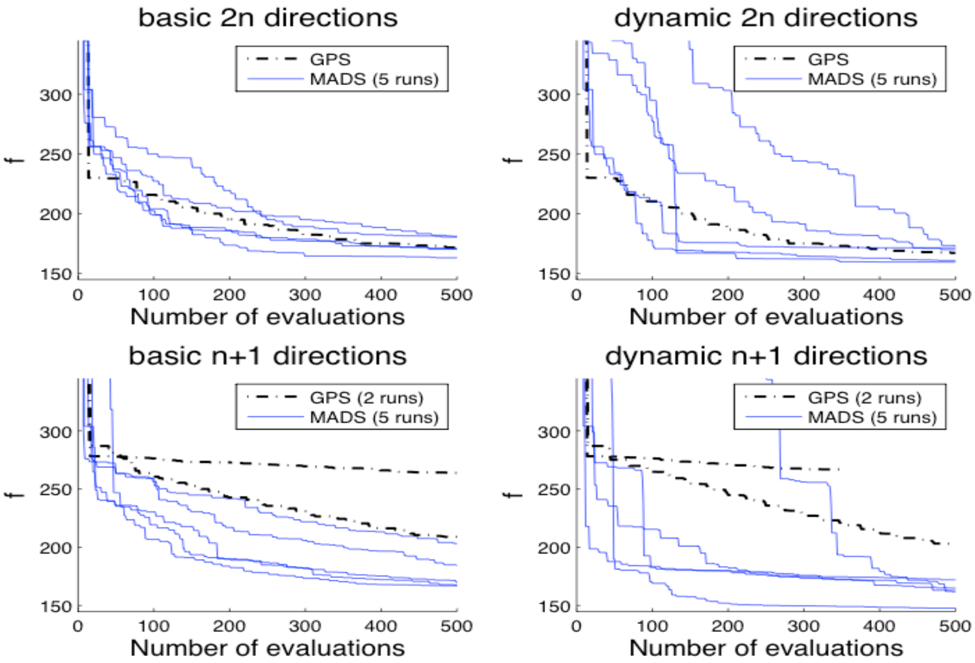


FIG. 5.3. *Data set 1 – Progression of the objective function value vs the number of evaluations.*

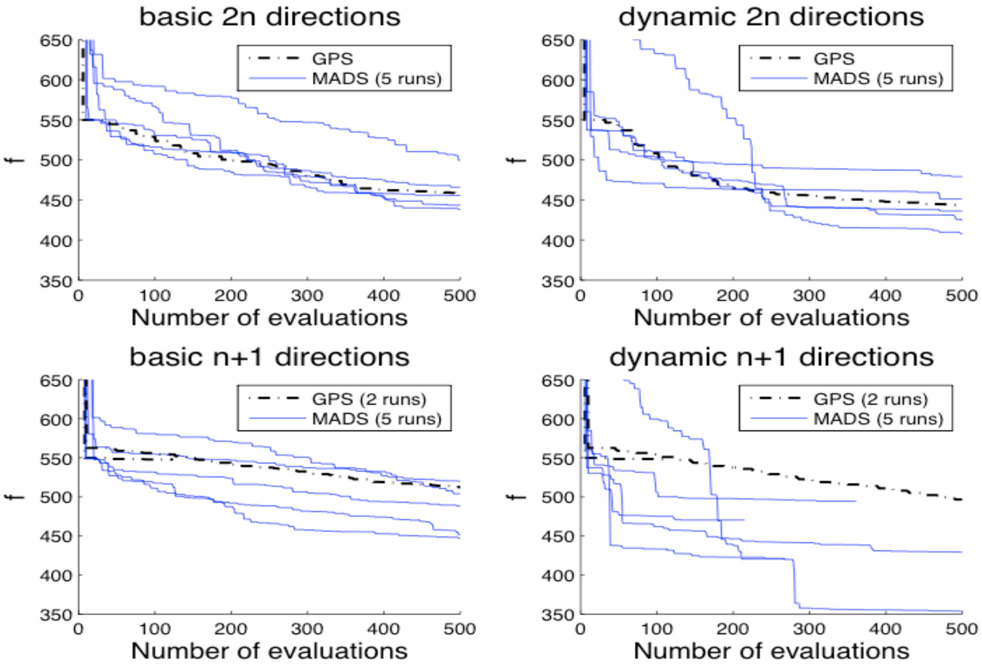


FIG. 5.4. *Data set 2 – Progression of the objective function value vs the number of evaluations.*

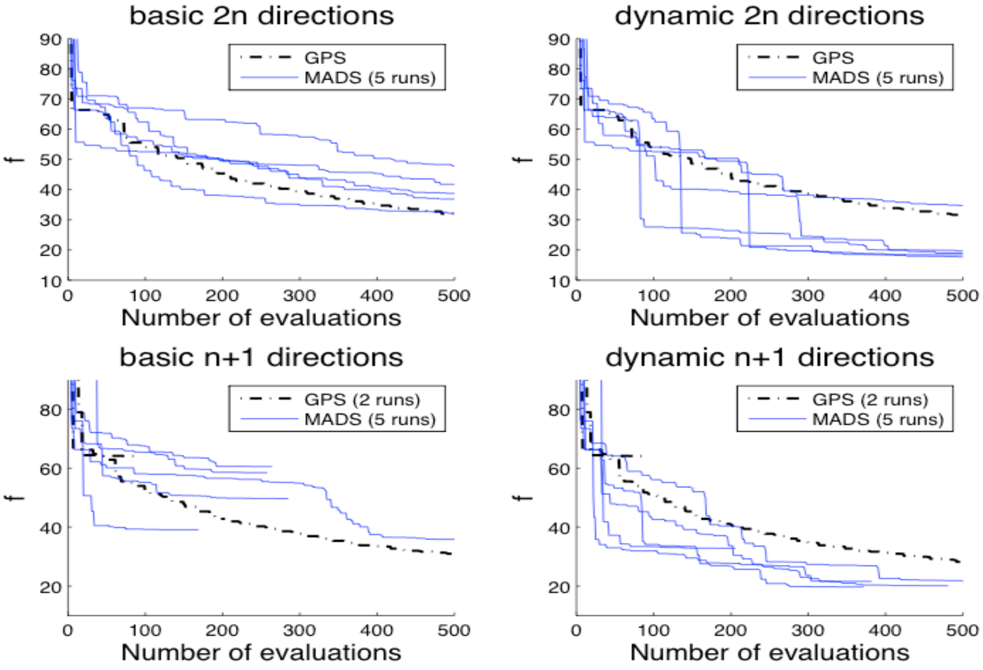


FIG. 5.5. *Data set 3 – Progression of the objective function value vs the number of evaluations.*

For the three sets of experimental data analyzed in [14], we compared the instances of GPS and MADS discussed above. The algorithms terminate whenever a minimal frame center with poll size parameter equal to 2^{-6} is detected, or whenever 500 functions evaluations are performed, whichever comes first. Figures 5.3, 5.4, and 5.5 show the progression of the objective function value versus the number of evaluations for each data set.

The plots suggest that the objective function value decreases more steadily with GPS than with MADS. This is because GPS uses a fixed set of poll directions that we know to be an excellent choice for this problem. By allowing more directions, MADS eventually generates a steeper descent direction, and the dynamic runs capitalize on this by evaluating f further in that direction thus sharply reducing the objective function value in a few evaluations. In general, if the number of function evaluations is limited to a fixed number, then it appears that MADS with the dynamic strategy gives a better result than GPS. For all three data sets, the dynamic runs are preferable to the basic runs. It also appears that for this problem, MADS runs with minimal $n+1$ directions perform better than the maximal $2n$ runs. In each of the three data sets, the best overall solution was always produced by MADS with the dynamic $n+1$ directions.

The quality of the best solutions produced by GPS and MADS can be visualized in Figure 5.6 where the difference between the experimental and predicted conversions are plotted versus time. A perfect model with perfectly tuned parameters would have had a difference of zero everywhere. The superiority of the solution produced by MADS versus GPS is mostly visible for the second data set near the time 160sec and the third data set near the time 200sec where in both cases the fit is better by approximately 1% .

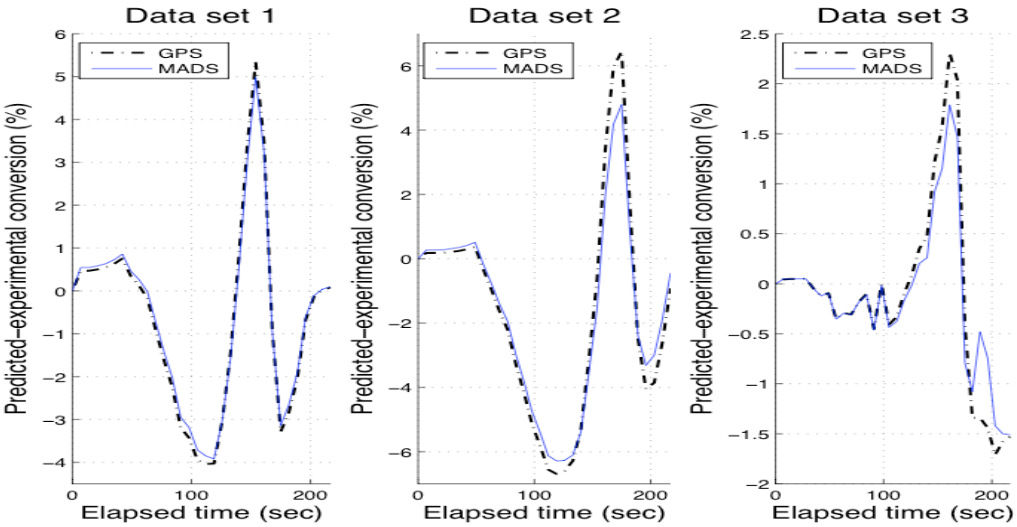


FIG. 5.6. *Conversion rate error versus time.*

5.3. Linear optimization on an hypersphere. The third example shows again the difficulty caused by being restricted to a finite number of polling directions. It also illustrates the effect of dimension. This is a problem with a linear objective and strictly convex full-dimensional feasible region, surely the simplest nonlinearly constrained problem imaginable.

$$\begin{array}{cl} \min\limits_{x\in\mathbb{R}^n} & \sum\limits_{i=1}^{n} x_i \\ \text{s.t.} & \sum\limits_{i=1}^{n} x_i^2 \leq 3n. \end{array}$$

There is a single optimal solution to that problem: every component of the vector x is $-\sqrt{3}$ and the optimal value is $-\sqrt{3}n$.

The starting point is the origin, and the algorithm terminates when $\Delta^p < 10^{-12}$, or when the number of function evaluation exceeds $600n$, whichever comes first. The algorithm was run with four values of n. For the GPS method we always used $D_k = D = [I, -I]$ with dynamic ordering. The GPS filter method is described in [4]. We used a search strategy, which we often use with the GPS filter method, consisting of a $5n$ point latin hypercube sample at the first iteration, and a $n/5$ random search at other iterations.

The behavior of the method is comparable for every value of n. In every case, the MADS algorithm converged to the global optimal solution. The GPS barrier approach quickly moved to a point on the boundary of the domain and stalled there. The GPS filter approach was able to move away from that point, but it converged to another sub-optimal solution. The absence of a SEARCH strategy, and the restriction to a finite number of POLL directions traps the iterates at a non-optimal solution. The addition of a random SEARCH strategy allows GPS, when n is $10, 20$ or 50, to move away from this solution, but it still was short of finding the optimal solution in the number of function calls allowed. The progression of the runs is illustrated in Figure 5.7.

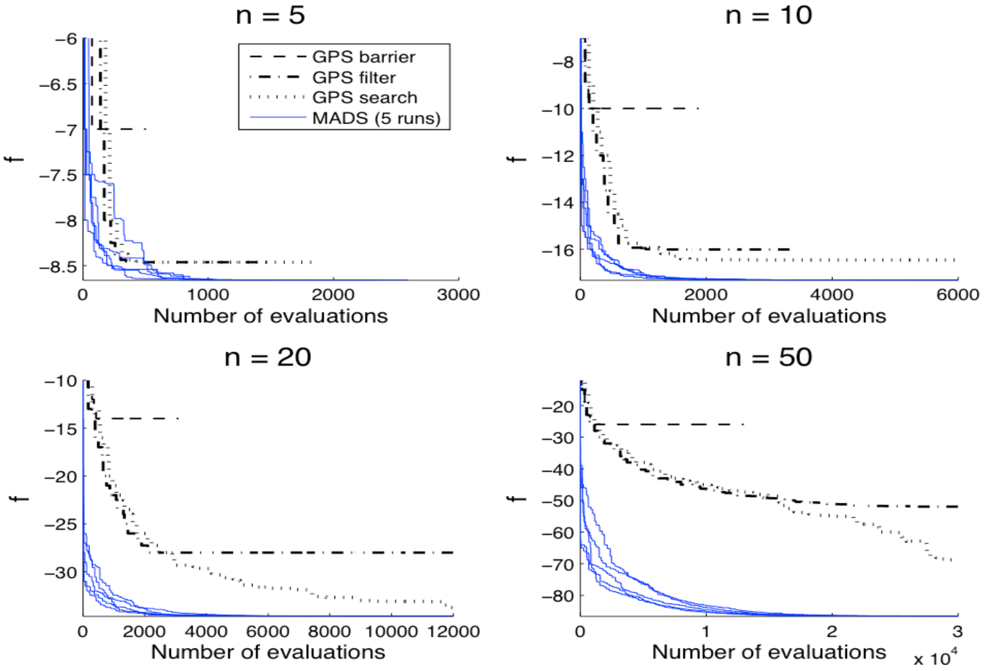


FIG. 5.7. *Progression of the objective function value vs the number of evaluations on an easy nonlinear problem.*

5.4. Numerical limitations. This last example does not satisfy the hypotheses of any GPS or MADS theorems because the optimizer is at $-\infty$. However, it is intended to see how well the various algorithms track a feasible region that gets narrow quickly. Consider the following problem in $\mathbb{R}^2$:

$$\begin{array}{cl} \displaystyle\min_{x=(a,b)^T} & a \\ \text{s.t.} & e^a \le b \le 2e^a. \end{array}$$

The starting point is $(0,1)^T$, and the algorithm terminates when $\Delta_k^m < 10^{-323}$ *i.e.*, when the mesh size parameter drops below the smallest positive representable number in double precision arithmetic. We admit that this is excessive, but we wanted to run the algorithms to their limits. The same strategies as in Section 5.3 are used.

The progression of the algorithms is illustrated in Figure 5.8. GPS with both the barrier and filter approaches to constraints converged quickly to points where the standard 2n basis does not contain a feasible descent direction. The filter GPS approach to constraints did better than the GPS barrier approach because it is allowed to become infeasible.

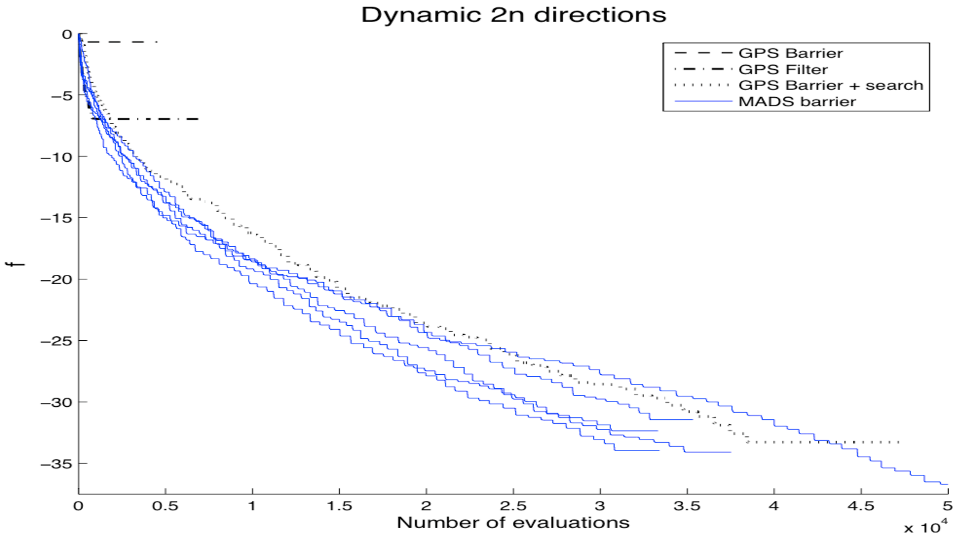


FIG. 5.8. *Progression of the objective function value vs the number of evaluations on a difficult nonlinear problem.*

All 5 runs of the LTMADS method of the previous section ended with roughly the same solution, a point where $a \pm \Delta_k^p = a$, which is all one can ask. The same behaviour is observed for GPS with a random SEARCH. The fact that LTMADS generates an asymptotically dense set of poll directions, and that a SEARCH step is conducted at each GPS iteration explain why both the GPS with a search and LTMADS do better than the GPS barrier or filter approach.

The feasible region is very narrow, and therefore it gets quite improbable that the MADS poll directions generate a feasible point. When such a feasible point is generated it is always very close to the frame center since the mesh and poll parameters are very small.

Even if the algorithm instances failed to solve this problem to optimality and converged to points that are not Clarke stationary points, the GPS and MADS convergence theory is not violated – yet. In all cases, there is a set of directions that positively span $\mathbb{R}^2$ such that for each direction either the Clarke generalized derivative is nonnegative or is an infeasible direction.

6. Discussion. GPS is a valuable algorithm, but the application of nonsmooth analysis techniques in [3] showed its limitations due to the finite choice of directions in [2]. MADS removes the GPS restriction to finitely many poll directions. We have long felt that this was the major impediment to stronger proofs of optimality for GPS limit points (and better behavior), and in this paper we find more satisfying optimality conditions for MADS in addition to opening new possibilities for handling nonlinear constraints.

We described a stochastic version of MADS, LTMADS, which performed well, especially for a first implementation. We expect others will find more, and perhaps better, implementations. The structure of this paper is intended to facilitate the introduction of new MADS instances, and back them with our convergence analysis. One only need to show that the new instance generates a dense set of refining directions. We chose to develop a stochastic way of generating these directions. We could have used a deterministic strategy, but felt that it lacked a desirable property. Indeed, the deterministic ways that we considered were such that when the algorithm terminated after a finite number of iterations, the set of poll directions was often far from being uniformly distributed in $\mathbb{R}^n$.

When n is small, our examples suggested that GPS with a random SEARCH behaved similarly to MADS. The similitude breaks down as the number of variables increases since the relative cost of the POLL step becomes more important. In MADS the POLL directions change at each iteration, but they are static in GPS.

We think that the work here is readily applied to choosing templates for implicit filtering [7], another very successful algorithm for nasty nonlinear problems.

7. Acknowledgments. Finally, we wish to thank Gilles Couture for coding NOMAD, the C++ implementation of MADS and GPS, and to acknowledge useful discussions with Andrew Booker, Mark Abramson and Sébastien Le Digabel.

REFERENCES

[1] M. A. Abramson, C. Audet, and J. E. Dennis, Jr. Generalized pattern searches with derivative information. *Mathematical Programming*, Series B, 100:3–25, 2004.

[2] C. Audet. Convergence results for pattern search algorithms are tight. *Optimization and Engineering*, 5(2):101–122, 2004.

[3] C. Audet and J. E. Dennis, Jr. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903, 2003.

[4] 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.

[5] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[6] 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.

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

[8] 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.

[9] I. D. Coope and C. J. Price. Frame-based methods for unconstrained optimization. *Journal of Optimization Theory and Applications*, 107(2):261–274, 2000.

[10] I. D. Coope and C. J. Price. On the convergence of grid-based methods for unconstrained optimization. *SIAM Journal on Optimization*, 11(4):859–869, 2001.

[11] C. Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76:733–746, 1954.

[12] D.E. Finkel and C.T. Kelley. Convergence analysis of the direct algorithm. Technical report, NCSU Mathematics Department, 2004.

[13] F. J. Gould and J. W. Tolle. Geometry of optimality conditions and constraint qualifications. *Mathematical Programming*, 2:1–18, 1972.

[14] R. E. Hayes, F. H. Bertrand, C. Audet, and S. T. Kolaczkowski. Catalytic combustion kinetics: Using a direct search algorithm to evaluate kinetic parameters from light-off curves. *The Canadian Journal of Chemical Engineering*, 81(6):1192–1199, 2003.

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

[16] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev.*, 45(3):385–482 (electronic), 2003.

[17] E. B. Leach. A note on inverse function theorem. In *Proceedings of the AMS*, volume 12, pages 694–697, 1961.

[18] R. M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. Technical Report 96–71, Institute for Computer Applications in Science and Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia 23681–2199, 1996.

[19] R. M. Lewis and V. Torczon. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12(4):1075–1089, 2002.

[20] M. D. McKay, W. J. Conover, and R. J. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[21] R. T. Rockafellar. Generalized directional derivatives and subgradients of nonconvex functions. *Canad. J. Math.*, 32(2):257–280, 1980.

[22] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, February 1997.