Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization

3 authors, including:

Charles Audet
Polytechnique Montréal
143 PUBLICATIONS   **4,626** CITATIONS

SEE PROFILE

Sébastien Le Digabel
Polytechnique Montréal
57 PUBLICATIONS   **1,242** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Book: Derivative-Free and Blackbox Optimization View project

Parameter Optimization View project

CrossMark

Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization

Charles Audet[1] · **Sébastien Le Digabel**[1] ·
Christophe Tribes[1]

Abstract Blackbox optimization deals with situations in which the objective function and constraints are typically computed by launching a time-consuming computer simulation. The subject of this work is the mesh adaptive direct search (MADS) class of algorithms for blackbox optimization. We propose a way to dynamically scale the mesh, which is the discrete spatial structure on which MADS relies, so that it automatically adapts to the characteristics of the problem to solve. Another objective of the paper is to revisit the MADS method in order to ease its presentation and to reflect recent developments. This new presentation includes a nonsmooth convergence analysis. Finally, numerical tests are conducted to illustrate the efficiency of the dynamic scaling, both on academic test problems and on a supersonic business jet design problem.

✉ Christophe Tribes
christophe.tribes@polymtl.ca

Charles Audet
Charles.Audet@gerad.ca;
https://www.gerad.ca/Charles.Audet

Sébastien Le Digabel
Sebastien.Le.Digabel@gerad.ca;
https://www.gerad.ca/Sebastien.Le.Digabel

[1] Département de mathématiques et génie industriel, GERAD, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal, Québec H3C 3A7, Canada

🖄 Springer

1 Introduction

Scaling variables can greatly affect the performance of an optimization algorithm (Dennis and Schnabel 1983; Fletcher 1980; Gill et al. 1981; Nocedal and Wright 1999; Papalambros and Wilde 2000). The mesh adaptive direct search (MADS) algorithm for constrained blackbox optimization is no exception (Audet and Dennis 2006). With the MADS default form, an initial scaling of the space of variables can be proposed explicitly by the user for each variable, or is obtained implicitly by considering the bounds on each variable and the magnitude of the components of the initial point.

The initial scaling is then used to construct a discretization of the space of variables, called the *mesh*. MADS algorithms are iterative, and each iteration attempts to improve the current best solution by generating trial points on the mesh. Unlike most meshes required to obtain numerical solutions of partial differential equations on complex physical domains (e.g., Liseikin 2009), the mesh used by the MADS algorithm at a given iteration is simply defined by orthogonal coordinate lines that intersect at some points. For the MADS algorithm, the mesh is a convenient support for the set of all potential trial points (feasible or not) that can be evaluated in $\mathbb{R}^n$ at a given level of granularity, n being the dimension of the space of variables. The granularity of the mesh is updated at the end of each iteration. The coarseness or fineness of all subsequent meshes is dictated by a single strictly positive real-valued parameter called the *mesh size parameter*. Hence, with the MADS default form, the relative mesh aspect ratios remain invariant as the algorithm progresses.

In the event of a successful iteration, i.e., when a trial point turns out to be a new incumbent solution, then the mesh size parameter is increased or kept constant and the search for a better solution is restarted. This allows the next iteration to generate trial points further away from this new incumbent solution. Alternatively, if none of the trial points generated during an iteration improves the incumbent, then the mesh size parameter is reduced. The set of trial points form some sort of discrete neighbourhood around the incumbent solution. Failing to improve the current solution implies that the incumbent solution is better than its neighbours, which is a crude sort of local optimality tied to the mesh size parameter.

The convergence analysis of the MADS algorithm studies convergent subsequences of incumbent solutions that were shown to be better than their neighbours, and for which the mesh size parameter goes to zero. These are called refining subsequences. At the limit of such a subsequence, the crude optimality conditions become the standard local optimality conditions. If the problem is sufficiently smooth, these conditions correspond to *(i)* a null gradient in the unconstrained case, and *(ii)* nonnegative directional derivatives in the directions tangent to the domain in the constrained case. In the nonsmooth case, these conditions state that *(iii)* zero belongs to the generalized gradient in the unconstrained case, and *(iv)* nonnegative directional generalized derivatives in the directions hypertangent to the domain in the constrained case. The details of this hierarchical convergence analysis based on decreasing requirements on the objective and constraints are found in Audet and Dennis (2006).

These theoretical results hold as the number of function evaluations goes to infinity. In practice however, the number of function calls is always limited, often by a given time-related budget. Furthermore, if the scaling of the variables is inappropriate, the progress made by the algorithm within a budget of evaluations might be marginal.

The contribution of the present work is to propose a way to improve the MADS algorithm by considering a mesh size parameter specific to each variable. This allows MADS to dynamically adapt the mesh in an anisotropic way: the relative mesh aspect ratios are allowed to vary. The idea of individual scaling of the variables to adapt to the curvature of the objective function is not new in direct search methods, as mentioned in the comprehensive survey by Kolda et al. 2003. Such approaches are proposed for methods requiring sufficient decrease conditions (García-Palomares and Rodríguez 2002; Kolda et al. 2006; Lucidi and Sciandrone 2002). Furthermore, Hough et al. (2001) report that increasing the mesh size parameter at every successful iteration could be counterproductive.

The main idea of the proposed contribution is that when a new incumbent solution is found, instead of coarsening the mesh for all variables using the same ratio, we coarsen it only with respect to the variables that were significantly modified in the new solution. Specifically, this is done by replacing the classical mesh size parameter in $\mathbb{R}_+$ that dictates the coarseness of the mesh, by a *mesh size vector* in $\mathbb{R}_+^n$, whose dimension equals the number of optimization variables. Since the entire convergence analysis of this type of methods is based on the mesh size parameter rather than in its new vectorial expression, the convergence analysis needs to be entirely revisited.

The paper is divided as follows. Section 2 presents the MADS class of algorithms with dynamic scaling. The technical part of the nonsmooth convergence analysis is deferred to the appendix section. Section 3 discusses a first implementation of this new class of algorithms. Numerical experiments are conducted in Sect. 4 on academic test problems and on a supersonic business jet design problem. Finally, the last section discusses future research directions.

Notation Vectors can be considered row or column vectors according to context. The superscript $k \in \mathbb{N}$ is not an exponent, but an index associated to the iteration number. The value of a raised to the power b is written with parenthesis: $(a)^b$ to avoid confusion with superscripts. If δ is a vector in $\mathbb{R}^n$, then $\mathrm{diag}(\delta)$ is the $n \times n$ diagonal matrix with $\delta \in \mathbb{R}^n$ as its diagonal. The set of m directions $D = \{d_1, d_2, \ldots, d_m\} \subset \mathbb{R}^n$ is said to be a positive spanning set if for any $v \in \mathbb{R}^n$, there exists an $\alpha \in \mathbb{R}_+^m$ such that $v = \sum_{i=1}^{m} \alpha_i d_i$. The unit sphere in $\mathbb{R}^n$ is denoted $\mathbb{S}^n = \{d \in \mathbb{R}^n : \|d\| = 1\}$.

2 MADS with dynamic scaling

Consider the general optimization problem

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

where f is a single-valued objective function, and Ω is the set of feasible solutions in $\mathbb{R}^n$.

Blackbox optimization targets problems in which properties of the objective function f as well as the functions defining the feasible set are not easily exploitable. This typically occurs when the output of these functions is obtained by launching a computer code with some $x \in \mathbb{R}^n$ as input. These functions are typically nonsmooth, time-consuming to evaluate, contaminated by numerical noise, which renders the application of classical optimization methods difficult. Direct search optimization methods are designed to interact directly with the outputs of the blackbox. An introduction to derivative-free optimization can be found in Conn et al. (2009).

In the present work, we focus on an extension of the MADS algorithm which dynamically scales the mesh for each variable independently. This new extension impacts the fundamental building blocks of the algorithm, and therefore we present it from scratch. This is also an opportunity to revisit some concepts and to present some aspects differently in order to ease the exposition.

2.1 High-level description of MADS

MADS is an iterative algorithm designed to solve problems of the form (1). It is launched from an initial point, or more generally from a finite collection of initial points $V^0 \subset \mathbb{R}^n$. The iteration number is represented by the superscript k and, at the start of iteration k, the set of trial points where the blackbox was previously evaluated is denoted by V^k. The current iterate x^k corresponds to the current best feasible point inside V^k and is called the incumbent solution, or the incumbent. It is simply a point of the set argmin$\{f(x) : x \in V^k \cap \Omega\}$.

In the present paper, constraints are handled with the *Extreme Barrier* approach from Audet and Dennis (2006) which consists of launching the algorithm on the unconstrained extended valued function

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

The goal at each iteration of a MADS algorithm is to replace the incumbent by a better one. In order to achieve this, finitely many tentative trial points are generated on a discretization of the space of variables called the mesh M^k. The formal definition of the mesh requires additional notation, and is deferred to the next subsection. Some of these trial points may be far from the incumbent, but a subset of these trial points needs to be distributed nearby the incumbent solution. These neighbouring trial points are called the poll points while the remaining trial points are called the search points, as detailed in the next section.

At the end of iteration k, if a better incumbent solution was produced, then the iteration is declared successful and M^{k+1} is updated to be at least as coarse as M^k. The rationale behind this is to allow the next iteration to take larger steps.

Alternatively, iteration k failed at producing a better incumbent solution, and therefore the incumbent solution is better than its neighbouring poll points. This can be seen as some crude form of local optimality for the incumbent. At the conclusion of a failed iteration, the mesh is refined so that the poll points generated during the next iteration will be closer to the incumbent solution.

2.2 The SEARCH and POLL steps

Let us focus on two important parameters related to the high-level description of the MADS algorithm from the previous subsection. In previous work, the coarseness of the mesh was controlled by a single positive scalar. In the present work, we extend this by defining $\delta_j^k > 0$ as the *mesh size parameter for the j-th variable at iteration k*, for $j = 1, 2, \ldots, n$. Together with the set of trial points visited thus far V^k, the vector $\delta^k \in \mathbb{R}_+^n$, called the *mesh size vector*, allows one to define the mesh at iteration k. The mesh is central to the algorithm since it contains all potential trial points:

$$M^k = V^k + \left\{ \text{diag}(\delta^k) z \; : \; z \in \mathbb{Z}^n \right\}. \tag{2}$$

With this definition, it follows that $t \in \mathbb{R}^n$ belongs to the mesh at iteration k if and only if there exists a trial point $v \in V^k$ at which the blackbox was previously evaluated, and an integer vector $z \in \mathbb{Z}^n$ such that $t = v + \text{diag}(\delta^k) z$. Furthermore, by recursively applying this observation, it follows that for any $t \in M^k$, there exists an element x^0 from the set of starting points V^0 such that

$$t = x^0 + \sum_{\ell=0}^{k} \text{diag}(\delta^\ell) z^\ell \tag{3}$$

for some integer vectors $\{z^0, z^1, \ldots, z^k\} \subset \mathbb{Z}^n$.

Iteration k attempts to improve the incumbent solution by performing a pair of steps called the SEARCH and the POLL. The SEARCH can be viewed as the diversification step of the algorithm since it attempts to improve the incumbent by generating finitely many trial points that are not required to be close to the incumbent. For doing this in practice, the user may indicate the trial points by exploiting some specific knowledge of the problem, or by using generic tools such as Latin Hypercube Sampling (McKay et al. 1979), Variable Neighborhood Search (Audet et al. 2008; Mladenović and Hansen 1997), or any other method that the user could come up with, as long as there are finitely many candidates in M^k.

Note that the SEARCH is optional and not necessary for the convergence analysis. When the SEARCH succeeds at improving the incumbent solution, then the POLL is not executed. The POLL step may be viewed as the intensification step of the algorithm as it aims at finding local optima. It uses the second important parameter which delimits the region in which the neighboring poll points are generated. This parameter is $\Delta_j^k \geq \delta_j^k$ and is called the *poll size parameter for the j-th variable at iteration k*, for $j = 1, 2, \ldots, n$. Similarly to the mesh size vector, we denote the vector $\Delta^k \in \mathbb{R}_+^n$ as the *poll size vector*. This vector is used to delimit a region around

the current incumbent solution $x^k \in V^k$ where the poll points will be selected. This region is called the *frame* (Coope and Price 2000) and is defined as follows.

Definition 2.1 The frame at iteration k is $\{x \in \mathbb{R}^n : |x_j - x_j^k| \leq \Delta_j^k, \ \forall j = 1, 2, \ldots, n\}$.

The POLL also generates a finite list of trial points, but more rigid rules are used than for the SEARCH. These poll points need to be distributed around the current incumbent solution in such a way that candidates are present in every half-space. Formally, the set of poll points at iteration k around the incumbent solution $x^k \in V^k$ is called the *poll set* and is given by:

$$P^k = \{x^k + \text{diag}(\delta^k)d : d \in D^k\} \subseteq M^k$$

where $D^k \subset \mathbb{Z}^n$ is a positive spanning set of directions. The next subsection describes links between the mesh and poll size vectors, and gives a condition ensuring that any poll trial point $x \in P^k$ belongs to the frame.

2.3 The poll size and mesh size vectors

At the first iteration, the poll size vector $\Delta^0 \in \mathbb{R}_+^n$ is either supplied by the user, or fixed by some procedure to reflect the scale of the variables. At the end of any iteration, the poll size parameter associated to the j-th variable is updated by multiplying the previous one by an integer power of some fixed real number $\beta_j > 1$, whose square is a rational number (the proof of Proposition 6.1 uses this requirement on rationality). The power is greater than or equal to zero on successful iterations, and strictly negative on unsuccessful iterations. This implies that the poll size parameters decrease if and only if the iteration is unsuccessful. In the actual implementation described in Sect. 3, $\beta_j = 2$ for every j.

The following definition fixes the value of the mesh size parameter in terms of the poll size parameter.

Definition 2.2 At iteration k, the mesh size parameter associated to the variable $j \in \{1, 2, \ldots, n\}$ is

$$\delta_j^k = \frac{\left(\min\left\{\Delta_j^0, \ \Delta_j^k\right\}\right)^2}{\sqrt{n}\ \Delta_j^0}.$$

With this definition, $\delta_j^k \leq \frac{1}{\sqrt{n}}\Delta_j^k$ and furthermore, the initial mesh size vector is automatically fixed to

$$\delta^0 = \frac{\Delta^0}{\sqrt{n}}. \tag{4}$$

The term $\sqrt{n}$ in the denominator accounts for the fact that Euclidean distances between objects grow with the dimension n. For example, in Definition 2.1, the furthest frame point x from the incumbent solution x^k satisfies $\|x - x^k\|_\infty = \|\Delta^k\|_\infty$ but its Euclidean distance satisfies $\|x - x^k\|_2 \leq \sqrt{n}\|\Delta^k\|_\infty$. Preliminary experimental results suggest that scaling the mesh size parameter using $\sqrt{n}$ in the denominator as in Definition 2.2 gives better results than not scaling it.

A consequence of this way of updating the poll size vector is summarized in the next Lemma.

Lemma 2.3 At iteration $k \geq 0$, and for every index $j \in \{1, 2, \ldots, n\}$, there exists an integer $r_j^k \in \mathbb{Z}$ such that the poll and mesh size parameters satisfy

$$\Delta_j^k = \Delta_j^0 (\beta_j)^{r_j^k} \tag{5}$$

and

$$\delta_j^k = \delta_j^0 (\beta_j)^{r_j^k - |r_j^k|} \leq \delta_j^0. \tag{6}$$

Proof Let j be an index in $\{1, 2, \ldots, n\}$. The result is clearly true at the initial iteration by setting $r_j^0 = 0$. At the end of every iteration, the poll size parameter is multiplied by an integer power of β_j. It follows that there exists a sequence of integers r_j^k such that $\Delta_j^k = \Delta_j^0 (\beta_j)^{r_j^k}$ for every k. Therefore

$$
\begin{aligned}
\delta_j^k &= \frac{\left(\min\left\{ \Delta_j^0, \ \Delta_j^0 (\beta_j)^{r_j^k} \right\} \right)^2}{\sqrt{n}\, \Delta_j^0} \\
&= \delta_j^0 \left(\min\left\{ 1, \ (\beta_j)^{r_j^k} \right\} \right)^2 \\
&= \delta_j^0 (\beta_j)^{\min\{0, 2r_j^k\}} \\
&= \delta_j^0 (\beta_j)^{r_j^k - |r_j^k|}.
\end{aligned}
$$

$\square$

The integer r_j^k from the previous lemma is called the *mesh index*, and the last expression gives the mesh size parameter as a function of the mesh index. One can easily see that for a given value of j, the poll and mesh size parameters go simultaneously to zero:

$$\{\Delta_j^k\}_{k \in K} \to 0 \quad \Leftrightarrow \quad \{r_j^k\}_{k \in K} \to -\infty \quad \Leftrightarrow \quad \{\delta_j^k\}_{k \in K} \to 0,$$

for some infinite subset K of indices. But, since the mesh size parameter is constructed using the square of the poll size parameter, it will converge to zero faster. Furthermore, we impose that the algorithm possesses a mechanism ensuring that if one of the poll size parameters goes to zero, then the norm of the poll size vector also converges to zero:

$$\{\Delta_j^k\}_{k \in K} \to 0 \Leftrightarrow \{\|\Delta^k\|\}_{k \in K} \to 0 \tag{7}$$

for any $j \in \{1, 2, \ldots, n\}$ and subset of indices K. Such a mechanism is described in the implementation of Sect. 3.

Finally, the following proposition prescribes conditions on the directions used to construct the poll set P^k, ensuring that the poll points are within the frame.

Proposition 2.4 Let $x \in P^k$ be a poll point at iteration k, and let $d \in D^k \subset \mathbb{Z}^n$ be such that $x = x^k + \mathrm{diag}(\delta^k)d$ where δ^k is the mesh size vector at iteration k. If

$$|d_j| \le (\beta_j)^{|r_j^k|} \tag{8}$$

for every $j = 1, 2, \ldots, n$, then x belongs to the poll frame, i.e., $|x_j - x_j^k| \le \Delta_j^k$.

Proof If $x = x^k + \mathrm{diag}(\delta^k)d$, then for each $j = 1, 2, \ldots, n$,

$$
\begin{aligned}
|x_j - x_j^k| &= \delta_j^k |d_j| \\
&\le \delta_j^k (\beta_j)^{|r_j^k|} && \text{by (8)} \\
&= \delta_j^0 (\beta_j)^{r_j^k - |r_j^k|}(\beta_j)^{|r_j^k|} && \text{by (6)} \\
&= \delta_j^0 (\beta_j)^{r_j^k} \\
&= \frac{\Delta_j^0}{\sqrt{n}}(\beta_j)^{r_j^k} && \text{by Definition 2.2} \\
&\le \Delta_j^k && \text{by (5).}
\end{aligned}
$$

$\square$

2.4 The modified MADS algorithm

Algorithm 1 summarizes the main steps of the modified MADS method. Similarly to the original MADS, it must be seen as a general family of methods defining the blocks necessary for convergence. Different implementations can be defined, as the one presented in Sect. 3.

Algorithm 1: MADS with dynamic scaling

Input : $V^0 \neq \emptyset$ the set of initial points in $\mathbb{R}^n$;
          $\Delta^0 \in \mathbb{R}^n_+$ the initial poll size vector;
          $\beta > 1 \in \mathbb{R}^n$ a vector of real numbers whose squares are rational;

for $k = 0, 1, 2, \ldots$ **do**
  Let $x^k \in \operatorname{argmin}\{f_\Omega(x) : x \in V^k\}$ be the incumbent solution.
  Compute the mesh size vector δ^k from Δ^k using Definition 2.2.
  Perform the SEARCH and possibly the POLL steps (or only part of them if a new incumbent point x^{k+1} is found).

  - OPTIONAL SEARCH: Test finitely many trial points in M^k.

  - LOCAL POLL: Test trial points in P^k.

  PARAMETER UPDATE: Update Δ^{k+1} through the mesh index r^{k+1} as follows. If the iteration was unsuccessful, then set $r_j^{k+1} < r_j^k$ to an integer for each $j = 1, 2, \ldots, n$. Otherwise the iteration was successful: the incumbent x^k was dominated by x^{k+1}. For each $j = 1, 2, \ldots, n$, set $r_j^{k+1} \geq r_j^k$ to an integer.
end

There are two important differences with the original MADS method. First there is no more the general direction matrix D (see Audet and Dennis 2006). Instead, it is implicitly fixed to $D = [-I_n \, I_n]$ where I_n is the identity matrix. In practice, it means that the mesh (2) is always based on orthogonal directions. Historically, the flexibility in the choice of the directions defining the mesh was introduced in the context of the *Generalized Pattern Search* algorithm, in which a fixed number of poll directions was used (Torczon 1997). With MADS and its dense sets of directions, this flexibility has never been exploited in practice simply because it was not necessary. We abandon this requirement on D in order to simplify the presentation.

The second difference constitutes the main contribution of the present work. A generalization of the method is obtained by only allowing the increase of selected mesh size parameters in the update step, at the end of a successful iteration. A practical rule for choosing which parameters to increase is presented in Sect. 3. It consists of allowing the mesh size parameters associated to the variables whose value have changed from iteration k to $k + 1$ to increase: For each $j = 1, 2, \ldots, n$, set $r_j^{k+1} = r_j^k$ if $x_j^{k+1} = x_j^k$ and set $r_j^{k+1} \geq r_j^k$ to an integer if $x_j^{k+1} \neq x_j^k$. In the case of a failed iteration, the behaviour remains the same as all the mesh size parameters must be decreased. A convergence analysis of MADS with dynamic scaling is presented in the appendix section. The main results are identical to those of MADS (Audet and Dennis 2006) but the analysis differs.

3 Anisotropic MADS, an implementation of MADS with dynamic scaling

The previous section gave a general framework for the MADS algorithm with dynamic scaling. In the present section, we propose a first instantiation of this algorithm. We make some algorithmic choices to simplify the presentation. For example, we set $\beta_j = 2$ for every index j.

The other details of the implementation are given in the following subsections. They include the way to initialize the poll size vector, how to update it after a failure and after a success, and how to define the poll directions.

3.1 Poll size initialization

The initial components of the poll size vector are a key point for the efficiency of the method. When no dynamic scaling is used, these values simply define the scaling of the problem, which is crucial as every practitioner knows. We must then choose values that reflect the best scaling that we can infer a priori. For that, our best guidelines are the lower and upper bounds on the variables, whose presence is usually beneficial with blackbox optimization methods. If the bounds are not available, then the initial point is considered. The proposed strategy for each $j = 1, 2, \ldots, n$ is to set the initial poll size parameter as follows:

$$\Delta_j^0 = \begin{cases} \dfrac{u_j - l_j}{10} & \text{if variable } j \text{ is bounded by the finite values } l_j < u_j, \\[2ex] \dfrac{|x_j^0 - b_j|}{10} & \text{if variable } j \text{ has only one bound } b_j \text{ which differs from } x_j^0, \\[2ex] \dfrac{|x_j^0|}{10} & \text{if variable } j \text{ has only one bound } b_j \text{ with } x_j^0 = b_j \neq 0, \\[2ex] \dfrac{|x_j^0|}{10} & \text{if variable } j \text{ has no bound and } x_j^0 \neq 0, \\[2ex] 1 & \text{otherwise.} \end{cases}$$

The factor ten in the denominator of the poll size initialization comes from previous versions of NOMAD (Le Digabel 2011) that work reasonably well in the absence of additional information provided by the user.

3.2 Updating the poll size vector at the end of an iteration

3.2.1 Unsuccessful iterations

At the end of an unsuccessful iteration k, each direction of a positive spanning set D^k failed to improve an incumbent solution. Decrease each poll size parameter as follows:

$$\Delta_j^{k+1} = \frac{\Delta_j^k}{2},$$

i.e., by setting $r_j^{k+1} = r_j^k - 1$, for each j in $\{1, 2, \ldots, n\}$.

3.2.2 Successful iterations

At the end of a successful iteration k, a new incumbent solution was generated in a nonzero direction $d \in \mathbb{R}^n$. Algorithm 1 states that for each index $j = 1, 2, \ldots, n$ the mesh size parameter can be increased or kept to its current value, i.e., $r_j^{k+1} \geq r_j^k$. In the proposed implementation, the mesh index is updated according to the following rule:

$$
r_j^{k+1} = \begin{cases} r_j^k + 1 & \text{if } |d_j| > \dfrac{1}{n} \|d\|_\infty \ \text{ or } \ r_j^k \geq -2, \\ r_j^k & \text{otherwise.} \end{cases}
$$

The rule compares $|d_j|$ to $\frac{1}{n} \|d\|_\infty$. The factor $\frac{1}{n}$ makes it more likely as n increases that the mesh index be incremented. The mesh index remains at r_j^k only when $|d_j|$ is small.

With this way of modifying the mesh index, it is possible that some of them became too small relative to the others. Let $\bar{r}^k = \max\{r_j^k : j = 1, 2, \ldots n\}$ be the largest mesh index at iteration k.

Consider each index $j = 1, 2, \ldots, n$. If the mesh index r_j^{k+1} became relatively small so that it is less than twice the largest one, then it is increased. Formally, we apply the following rule: if $r_j^{k+1} < -2$ and if $r_j^{k+1} < 2\bar{r}^k$, then increment r_j^{k+1} by one.

In our numerical experiments, isotropic MADS is identical to anisotropic MADS except that the mesh index is increased at all times and for all indices j on successful iterations.
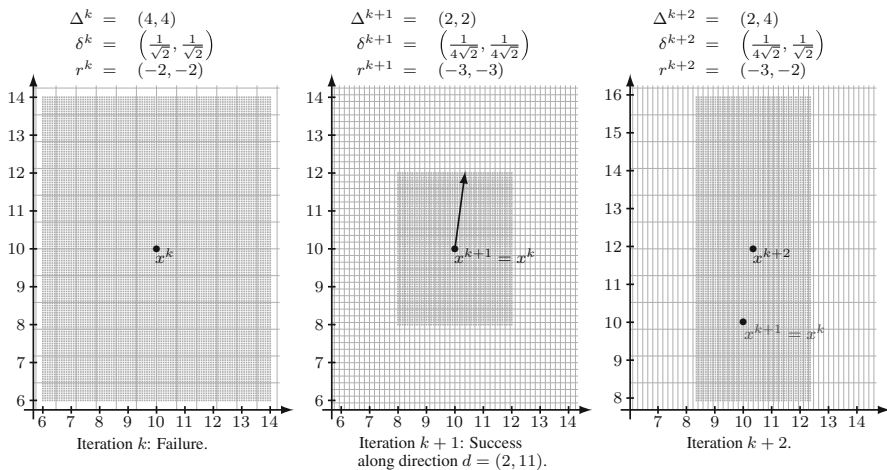

Fig. 1 Example of poll and mesh sizes. The incumbent is represented by the *central dot*. The mesh is represented by the intersection of the *lines*. The *poll points* need to be chosen on the mesh within the frame, represented by the *shaded box*

3.3 Example of poll and mesh sizes

Figure 1 illustrates the modifications of the poll and mesh size vectors on an unconstrained problem in $\mathbb{R}^2$ starting at iteration k with incumbent $x^k = (10, 10)$, poll size vector $\Delta^k = (4, 4)$, mesh size vector $\delta^k = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ and mesh index $r^k = (-2, -2)$. In the figure, the intersections of the horizontal and vertical lines represent the trial points on the mesh M^k and the shaded rectangle delimits the frame from Definition 2.1.

Iteration k fails at improving the incumbent x^k. Consequently, each component of the poll size vector is divided by 2, those of the mesh size vector are divided by 4, and the next mesh index is set to $r^{k+1} = (-3, -3)$.

At iteration $k + 1$, the trial point $x^{k+1} + \text{diag}(\delta^{k+1})d$, with $d = (2, 11)$, is successful. The next incumbent is set to that value $x^{k+2} = \left(10 + \frac{1}{2\sqrt{2}}, 10 + \frac{11}{4\sqrt{2}}\right) \approx (10.35, 11.94)$. The mesh index update rule of Sect. 3.2 yields $r^{k+2} = (-3, -2)$. The value r_1^{k+2} remains at r_1^{k+1} because $d_1 = 2$ is relatively small compared to $\frac{1}{n}\|d\|_\infty = 11/2$. The frame at iteration $k + 2$ is deformed, when compared to those of iterations k and $k + 1$.

At iterations k and $k + 1$, the ratio between the two mesh size parameters was 1. At iteration $k + 2$, the prior success was obtained by mainly modifying the second variable. At this point the ratio between the two mesh size parameters is 1/4. This ratio can be interpreted as a measure of the mesh anisotropy. The mesh is four times coarser with respect to the second variable.

3.4 Poll set construction

This subsection describes a way to generate the poll directions. It is inspired from the ORTHOMADS implementation (Abramson et al. 2009) except that it is no more restrained to the use of the quasi-random Halton sequence (Halton 1960). Another distinction is that directions are no longer orthogonal because of the anisotropic nature of the mesh. However, since the Householder orthogonal transformation is still used, we continue to qualify these directions as ORTHOMADS directions.

Let $\{u^\ell\}_{\ell=0}^{\infty}$ be a set of normalized directions in $\mathbb{R}^n$ such that their union is dense in the unit sphere. One option for this step is to consider random directions, which has been chosen for our numerical tests as detailed in Sect. 4. Another possibility is to consider the QR-MADS type of directions described in Van Dyke and Asaki (2013). Randomly generated directions are used together with a sufficient decrease strategy (Martínez and Sobral 2013; Vicente and Custódio 2012).

To ease the presentation, let us consider the iteration k and let v be one of these normalized directions. The Householder matrix

$$H = I - 2vv^T$$

is an orthonormal basis of $\mathbb{R}^n$. We will write $h \in H$ to denote a column of the basis H.

By multiplying the j-th row of H by Δ_j^k, we obtain a new basis $H' = \text{diag}(\Delta^k)H$ of $\mathbb{R}^n$ scaled by the poll size parameter values. Next, let us round each entry of the new basis as follows: The j-th entry of the ℓ-th column of the matrix H' can be written as $H'_{j\ell} = a_{j\ell} \times \delta_j^k$ for some real number $a_{j\ell}$. Let $\alpha_{j\ell} = \text{round}(a_{j\ell})$ and set $H''_{j\ell} = \alpha_{j\ell} \times \delta_j^k$. Then the set $\{x^k + h : h \in H''\}$ belongs to M^k and the poll set P^k is defined to be $\{x^k \pm h : h \in H''\}$.

All of this can be compactly written as

$$H'' = [H''_{jl}] \quad \text{where} \quad H''_{jl} = \text{round}\left(\frac{\Delta_j^k \times H_{jl}}{\delta_j^k}\right) \times \delta_j^k, \tag{9}$$

and Fig. 2 illustrates the different steps leading to the poll directions. The left part of the figure shows the $2n$ orthogonal directions forming a maximal positive basis. They correspond to the positive and negative columns of the Householder matrix H. The central part of the figure stretches these directions using the poll size vector. It represents the positive and negative columns of the matrix H'. It remains a maximal positive basis, but the directions are not orthogonal. The right part of the figure shows the actual poll points, obtained by rounding to the mesh.

The following result shows that the poll set obtained by rounding to the mesh is formed by a maximal positive basis when the poll size vector is sufficiently small.

Proposition 3.1 At iteration k, if $\Delta^k \leq \frac{1}{\sqrt{n}}\Delta^0$ then the matrix H'' obtained by Eq. (9) is such that $[-H'' \ H'']$ forms a maximal positive basis of $\mathbb{R}^n$.
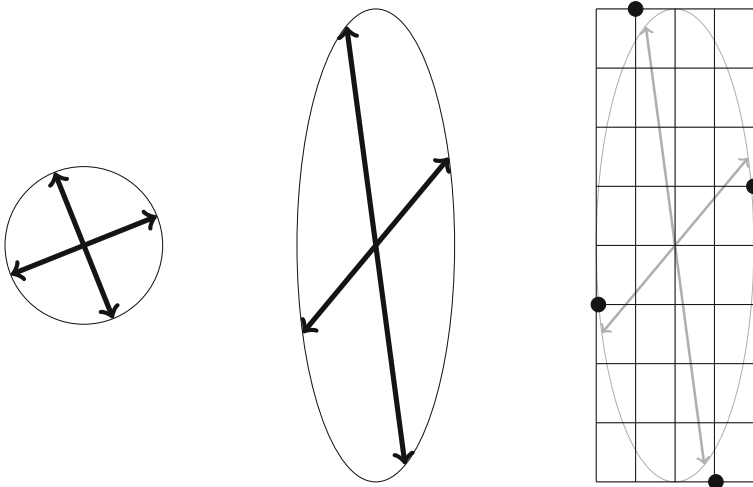

Fig. 2 Stretching the set of poll directions and rounding to the mesh. An orthogonal positive basis $[-H \ H]$ is represented to the *left*. The basis is stretched in the *center* of the figure to $[-H' \ H']$. The stretched positive basis is rounded to the current mesh, and the resulting directions $[-H'' \ H'']$ are represented in the *right* part of the figure by the *dark circles*

Proof Singular value decomposition theory (Van Loan 1985) ensures that the minimal perturbation on a basis $H \in \mathbb{R}^{n \times n}$ to make it singular is tied to its smallest singular value σ_n as follows

$$\min_{B \in \mathbb{R}^{n \times n}} \left\{ \|H - B\|_F \ : \ \mathrm{rank}(B) < n \right\} \ = \ \sigma_n.$$

In our context, the Householder matrix H is orthonormal and therefore all of its singular values are equal to one. The rounding operation in Eq. (9) ensures that

$$\left(\frac{\Delta_j^k \times H_{j\ell}}{\delta_j^k} - \frac{1}{2} \right) \times \delta_j^k \ \leq \ H_{j\ell}'' \ \leq \ \left(\frac{\Delta_j^k \times H_{j\ell}}{\delta_j^k} + \frac{1}{2} \right) \times \delta_j^k,$$

which may be rewritten more simply as

$$\left| H_{j\ell}'' - \Delta_j^k \times H_{j\ell} \right| \leq \frac{\delta_j^k}{2}.$$

Let $B = \mathrm{diag}(\Delta^k)^{-1} H''$ be the matrix obtained by dividing the j-th row of H'' by the poll size parameter Δ_j^k, i.e., $[B_{j\ell}] = \left[\frac{H_{j\ell}''}{\Delta_j^k} \right]$. Equations (4), (5) and (6) allow us to reformulate the last inequality as

$$\left| B_{j\ell} - H_{j\ell} \right| = \left| H_{j\ell} - B_{j\ell} \right| \ \leq \ \frac{\delta_j^k}{2\Delta_j^k} \ = \ \frac{\delta_j^0 (\beta_j)^{-|r_j^k|}}{2\Delta_j^0} \ = \ \frac{(\beta_j)^{-|r_j^k|}}{2\sqrt{n}}.$$

Therefore, if $\Delta_j^k \leq \frac{1}{\sqrt{n}} \Delta_j^0$ for each $j \in \{1, 2, \ldots, n\}$ then this implies that r_j^k is negative, $(\beta_j)^{-|r_j^k|} = \frac{\Delta_j^k}{\Delta_j^0} \leq \frac{1}{\sqrt{n}}$ and consequently $\left| H_{j\ell} - B_{j\ell} \right| \leq \frac{1}{2n}$. The Frobenius norm of $H - B$ satisfies

$$\|H - B\|_F \ \leq \ \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{4n^2}} \ = \ \frac{1}{2},$$

which is strictly less than one, the smallest singular value of the matrix H.

It follows that the perturbation on H is sufficiently small so that both B and $H'' = \mathrm{diag}(\Delta^k) B$ are non-singular matrices. $\qquad \square$

4 Numerical results

Numerical tests are conducted using the beta version 3.7 of the NOMAD (Le Digabel 2011) software package freely available at https://www.gerad.ca/nomad. All tests use the ORTHOMADS strategy with $2n$ poll directions (Abramson et al. 2009), or the $n + 1$ poll directions constructed by pruning half of the $2n$ directions based on the last direction of success, and by adding the negative sum as the $(n + 1)$-th direction (Audet et al. 2014). These two strategies are denoted by

MADS 2n and MADS(suc,neg), respectively. The opportunistic strategy is enabled, which consists in sorting a given list of poll trial points before it is evaluated, using the last direction of success, and to interrupt the evaluations as soon as a new improvement is made. Tests are also conducted in the presence and absence of quadratic models.

In order to obtain a dense set of normalized directions $\{u^\ell\}_{\ell=0}^\infty$ as defined in Sect. 3.4, we randomly generate points in the unit sphere using n normal variables as follows (Marsaglia 1972):

$$u^\ell = \frac{X_1 + X_2 + \ldots + X_n}{\sqrt{X_1^2 + X_2^2 + \ldots + X_n^2}}$$

for some index ℓ and $X_j \sim N(0,1)$ for all $j \in \{1, 2, \ldots, n\}$.

In practice, each normal random value is computed using the polar form of the Box-Muller transformation (Box and Muller 1958) based on draws from the uniform random generator found at http://madrabbit.org/~ray/code/xorshf96.c. This strategy to generate a dense set of poll directions is used for both the new strategy with dynamic scaling and for the previous isotropic one. These two versions are called aniso and iso,[1] for the anisotropic and isotropic variants, respectively. The only difference between the two variants resides in the strategy for updating the poll size vector for each component separately or all together. In practice, this is set by the `anisotropic_mesh(yes|no)` parameter in NOMAD.

In the next two subsections, two settings are considered to test the efficiency of the new strategy: A set of analytical functions, and a realistic application. The stopping criteria are set to be a maximal number of evaluations and a minimal mesh size of 10^{-13}, the default NOMAD value.

In both cases, many instances on different problems are employed allowing the presentation of results with *data profiles* (Moré and Wild 2009) using the following convergence test:

$$\bar{f}_{\text{feas}} - f(x_e) \geq (1 - \tau)(\bar{f}_{\text{feas}} - f^*)$$

where x_e denotes the best feasible iterate obtained after e evaluations of one of the algorithmic variants on one of the problems, τ is the convergence tolerance and f^* is the objective function value of the best feasible solution obtained by all tested variants of NOMAD, for all the instances of the considered problem. These different instances are obtained by different starting points or different random generator seeds. To obtain a common reference for the problem, we have determined the average of the first feasible objective function values, noted $\bar{f}_{\text{feas}}$, on all instances of that problem. The convergence test is equivalent to testing if the objective function is in a range of values with f^* as a lower bound.

In the graph of the data profiles, the horizontal axis represents the convergence in terms of number of evaluations, while the vertical axis corresponds to the proportion of problems solved according to the convergence test within a given convergence

---

[1] Our numerical experiments show that the performances of MADS 2N(iso) and the algorithm in Abramson et al. (2009) are practically identical.

tolerance τ. One curve is drawn for each algorithm, which allows to graphically compare the convergence of the different methods. For example, if one plot is always above the other, it means that the corresponding method dominates the other as it systematically solves more instances than the other for the same budget of function evaluations.

4.1 Test problems from the derivative-free optimization literature

We test 212 analytical problems from the derivative-free optimization literature (Moré and Wild 2009). These 212 instances are generated from 22 different CUTEst functions (Gould et al. 2015) with some different starting points and four variants (smooth, nonsmooth, deterministically noisy, and randomly noisy). The number of variables ranges from 2 to 12, and the problems are unconstrained except for eight of them in which the variables are nonnegative.

Data profiles display the fraction of problems solved for a given tolerance depending on the progression of the algorithm. Here, the convergence tolerance for matching the best obtained solution is fixed at $\tau \in \{10^{-3}, 10^{-5}, 10^{-7}\}$, and the convergence is represented by the equivalent number of simplex gradient evaluations, i.e., the number of groups of $(n+1)$ function calls. For these problems, the budget of evaluations is expressed with the same convention and has been taken as $400(n+1)$ calls.

Preliminary tests have shown an influence of the random number generator seed on the data profiles. The random number generator is used to obtain the poll directions and it affects the solutions obtained for these problems. To make the data profiles less sensitive to the seed selection and more conclusive, a series of 10 different seeds are used for each problem and each algorithm. This leads to a total of 4,240 executions, summarized in the data profiles of Figs. 3 and 4.

The profiles with MADS 2n and with MADS(suc,neg) show that there is a performance gain when the dynamic anisotropic mesh is enabled. For a given budget of evaluations, using the dynamic scaling allows to solve approximately 4 % more problems.

The data profiles in Fig. 4 show that using quadratic models for ordering trial points before evaluation and also for the SEARCH step of MADS improves the performance in finding the best solution. This confirms the results of previous study (Conn and Le Digabel 2013). In addition, it can be noted that the dynamic scaling is beneficial whether quadratic models are enabled or not. However, although quadratic models are used by default in NOMAD, they are disabled during the remaining numerical tests to focus solely on the effect of dynamic scaling.

4.2 The supersonic business jet design example

The supersonic business jet design problem is presented in Agte et al. (1999) and has been used to demonstrate the use of coordination algorithms such as the nonhierarchical analytical target cascading (ATC) formulation (Tosserams et al. 2010) for solving multidisciplinary design optimization (MDO) problems. The
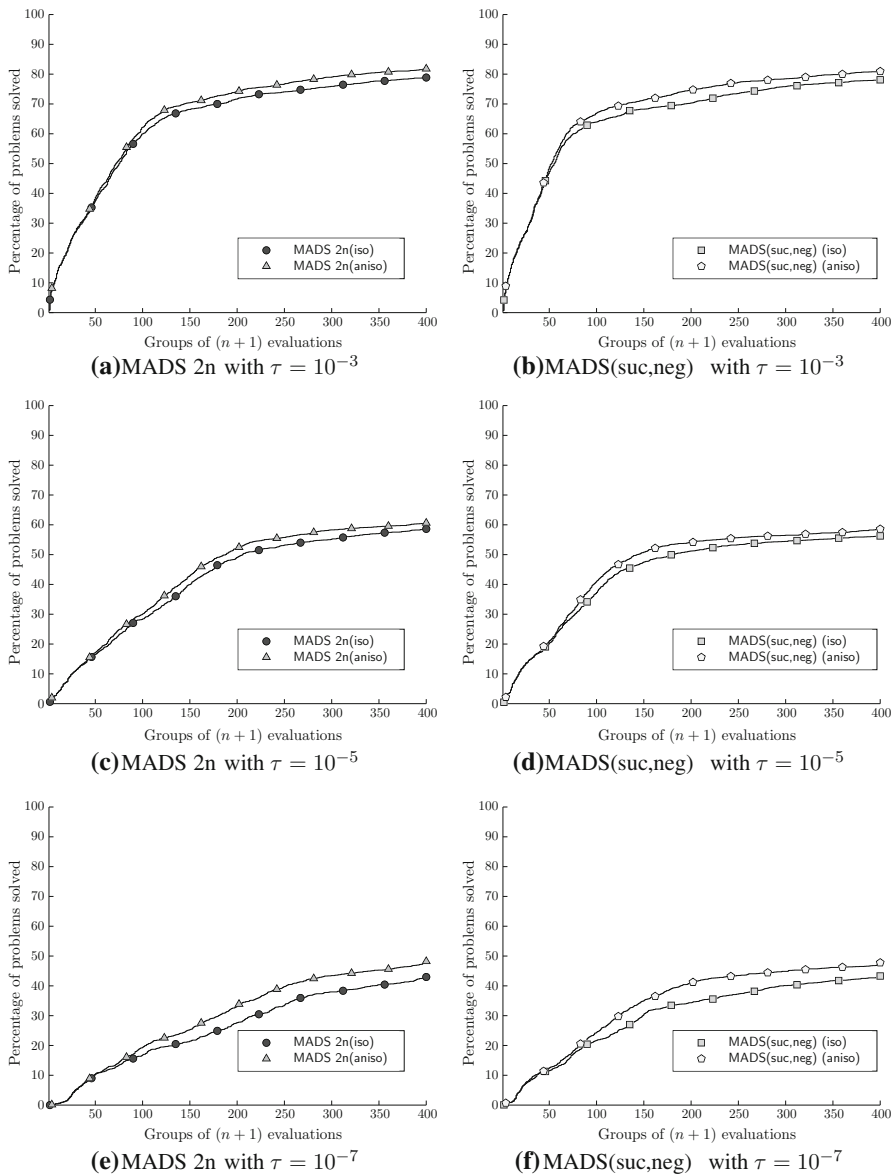
Fig. 3 Data profiles obtained with MADS 2n and MADS(suc,neg) and a convergence tolerance of $\tau = 10^{-3}$, 10^{-5} and 10^{-7} for 10 replications of 212 test problems from the literature. MADS 2n with $\tau = 10^{-3}$ MADS(suc,neg) with $\tau = 10^{-3}$ MADS 2n with $\tau = 10^{-5}$ MADS(suc,neg) with $\tau = 10^{-5}$ MADS 2n with $\tau = 10^{-7}$ MADS(suc,neg) with $\tau = 10^{-7}$

design problem involves the disciplines "Structure", "Aerodynamics", "Propulsion", and "Aircraft". In practice, each discipline analysis is calculated by a MATLAB function that takes some arguments as input and evaluates a subset of the
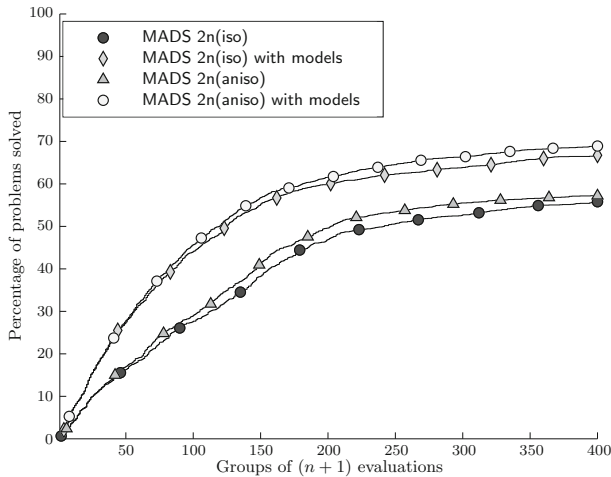
Fig. 4 Data profiles obtained with MADS 2n with and without quadratic models. A convergence tolerance of $\tau = 10^{-5}$ is considered for ten replications of 212 test problems from the literature

aircraft characteristics. The discipline analyses are interdependent: some discipline outputs are inputs of other disciplines. Only the multidisciplinary feasible optimization (MDF) approach (Cramer et al. 1994) for solving the design problem is considered here. The multidisciplinary feasibility is achieved by a fixed-point iterative approach calling the MATLAB functions in sequence until the maximum relative change of all outputs between two consecutive iterations is lower than a given precision of 10^{-5}. If this value is not reached after 50 fixed-point iterations, the multidisciplinary analysis is considered as failed.

The design objective is to minimize the weight of an aircraft while satisfying constraints from the disciplines. A description of the original optimization problem is available in Tosserams et al. (2010). Several optimal solutions can be obtained for this problem depending on the initial point considered. To better compare the algorithmic performances, a simpler instance of the problem has been considered. The original MDF optimization problem has 29 independent design variables and 78 design constraints. The objective, the design variables and constraints, the reference design values and the bound constraints considered in this work are the same as in Tosserams et al. (2010) except that the design variables associated to the structural elements thicknesses (18 variables) are fixed to the optimal values reported and the reference value for the wing surface area design variable is changed from 500 sq. ft. in the original problem to 650 sq. ft. in our modified version compared with the reported optimal value of 667 sq. ft. Among the design constraints, 12 are invariant and feasible by fixing the structural elements thicknesses. With these changes from the original problem, NOMAD is able to obtain more consistently solutions with objective in a narrower range. All of our numerical tests are conducted on this simpler instance.

Similarly to what is done in Tosserams et al. (2010), the optimization process is launched from 100 randomly generated starting points, selected within ± 10 % of

the reference values. The initial points are infeasible with objective values ranging from 34,020 lbs to 60,272 lbs. The design constraints are treated with the Extreme Barrier strategy, and the infeasible solutions are simply rejected during optimization. Because the reference value is infeasible, an initial phase to find a feasible solution is conducted by **NOMAD** by minimizing the constraint violation function Audet et al. (2010). A maximal budget of 20,000 multidisciplinary analyses function calls is imposed.

Under these conditions, we observe that all solutions produced by **NOMAD** are strictly feasible, and the final objective function values range from 34,271 lbs (best obtained objective value) to 35,876 lbs. In most cases, **NOMAD** terminates because the maximum number of evaluations is reached. The optimal weights reported in Tosserams et al. (2010) range from 33,600 lbs to 38,100 lbs with a constraint violation tolerance of 10^{-3}.

The data profiles displayed in Fig. 5 are obtained using tolerances $\tau \in \{2 \times 10^{-2}, 5 \times 10^{-2}\}$, which corresponds to the objective ranges of [34,271; 34,610] lbs and [34,271; 35,118] lbs, reached by 50 and 84 % of the runs, respectively. As for the analytic cases, the data profiles illustrate that the anisotropic mesh strategy outperforms the isotropic mesh strategy in reaching a solution.

The histogram plot given in Fig. 6 complements the data profile. It reveals that with a budget of 20,000 evaluations, for 36 of the 100 starting points, **MADS 2n(iso)** and **MADS 2n(aniso)** obtain solutions for which $|f_{iso} - f_{aniso}| < 125$ lbs, where f_{iso} and f_{aniso} denote the values of the solutions obtained by **MADS 2n(iso)** and **MADS 2n(aniso)**, respectively. In other words, from these 36 starting points both algorithms have obtained comparable solution values. However, for 43 of the 100 starting points **MADS 2n(aniso)** obtained a better solution than **MADS 2n(iso)** and for the remaining 21 starting points **MADS 2n(iso)** obtained a better solution than **MADS 2n(aniso)**.
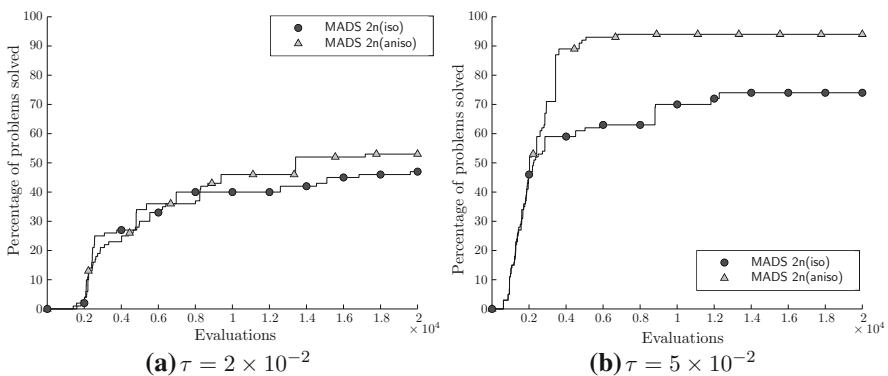

Fig. 5 Data profiles for the supersonic business jet design problem from 100 different starting points (one replication). $\tau = 2 \times 10^{-2}$ $\tau = 5 \times 10^{-2}$
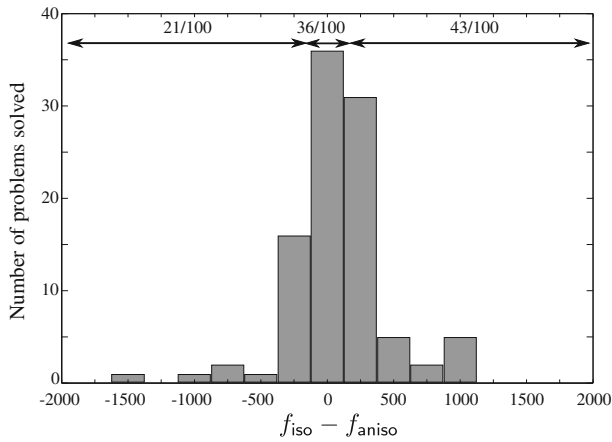
Fig. 6 Histogram plot of the difference in objective functions obtained by MADS 2n(iso) and MADS 2n(aniso) after 20,000 evaluations for 100 starting points
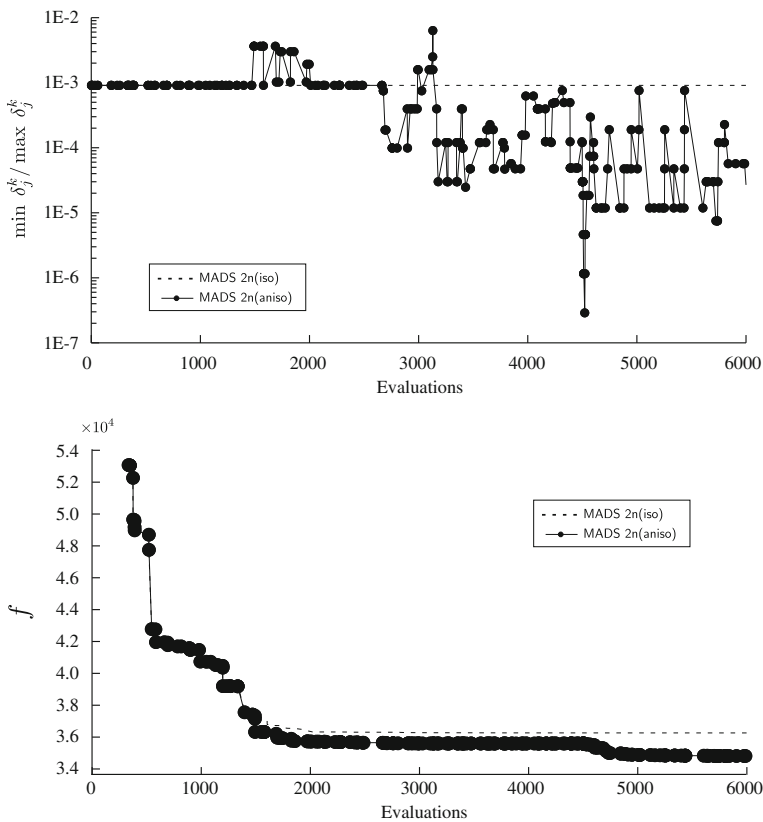

Fig. 7 Variation of the mesh size min/max ratio for a single run of the supersonic business jet design problem and objective function value versus number of function evaluations

In addition, the maximum of the histogram plot is skewed to the right of the central bar which shows that the anisotropic strategy described in the paper is beneficial.

The above analyses pertain to 100 runs. Let us now focus on a single run. Figure 7 shows the evolution of the anisotropy as the algorithm is deployed during the first 6,000 evaluations, by plotting the ratio of the minimum and maximum mesh size parameter values over all variables versus the number of blackbox evaluations on the runs MADS 2n(iso) and MADS 2n(aniso). Of course, the ratio is constant for the isotropic case and corresponds to the initial ratio for the anisotropic case. The value of this constant ratio is fixed by the poll size initialization, as described in Sect 3.1. Here, there is a scaling difference of the order of 10^{-3} between all the variables.

The figure shows that quite rapidly, MADS 2n(aniso) adapts the mesh size parameters to widely different magnitudes as the algorithm progresses. This adaptation occurs after the mesh index decreases below -2 as described in Sect. 3.2. The bottom plot of the same figure shows the evolution of the best feasible objective function value for MADS 2n(aniso) and MADS 2n(iso). Inspection of the log file reveals that the downward spike near evaluation 4,600 corresponds to significant decreases in the objective function values after a sequence of marginal improvements.

More precisely, between evaluations 4,507 and 4,522, a series of successful incumbents are obtained, and the objective function value reduced from 35,597.09 lbs to 35,556.02 lbs. The largest mesh size parameter increased from 7.1×10^{-7} to 4.5×10^{-5} while the smallest one remained constant at 1.3×10^{-11}. These successes are obtained while some of the other design variables remained constant and others changed values.

5 Discussion

This work introduces a new way of defining and updating the mesh, the spatial discretization on which the MADS algorithm for blackbox optimization is based. It allows a simpler, more intuitive, and more general description of the algorithm, including a simplified convergence analysis. In practice, this updated version of ORTHOMADS enables the use of an automatic and dynamic way of scaling the variables. In addition, more general types of directions can be considered as the trial points are no more restrained by the quasi-random Halton sequence.

We have compared the previous isotropic MADS approach with the proposed one. On both a collection of standard test problems and on an engineering application, the data profiles showed a clear domination of the anisotropic approach. Further investigation on the engineering problem showed that the anisotropy is exploited and allows to find better solutions more rapidly.

The proposed new algorithmic variant will be available in the release 3.7 of the NOMAD software package available at https://www.gerad.ca/nomad.

Priority in future research will be given to adapt the Progressive Barrier (Audet and Dennis 2009) for constraints to the new framework. Another project will

analyze links between the anisotropic mesh size parameters and the variable sensitivities, together with the exploitation of this information for the variable-based decomposition techniques of (Adjengue et al. 2014; Dennis et al. 2008).

Appendix: Convergence analysis of MADS with dynamic scaling

The convergence analysis of MADS with dynamic scaling follows the same two steps as in the previous isotropic case (Audet and Dennis 2006). Under appropriate conditions, the first step proves the existence of a subsequence of iterations for which the norm of the mesh and poll size vectors converges to zero. The second step studies unsuccessful iterations, and considers the limit of a convergent subsequence of incumbent solutions x^k for which the mesh size parameters goes to zero.

The key element of this analysis is that $f_\Omega(x^k)$ is less than or equal to the extreme barrier objective function value evaluated at all neighbouring poll points of x^k. Now, as the iteration k goes to infinity, the distance from the poll points to the incumbent x^k gets arbitrarily small, which leads to the satisfaction of necessary optimality conditions at the limit points.

The convergence analysis below relies on the standard assumptions that there exists a starting point $x^0 \in V^0 \cap \Omega$ such that $f(x^0)$ is finite, and that all iterates $\{x^k\}$ produced by the MADS algorithm lie in a bounded set. Another algorithmic requirement stated in Sect. 2.3 is that the β_j parameters are such that $(\beta_j)^2$ are rational numbers. This requirement imposed on the MADS algorithm is not an artifice of the proof since it cannot be relaxed as demonstrated in Audet (2004) on an example.

Proposition 6.1 The mesh and poll size vectors produced by an instance of Algorithm 1 satisfy

$$\liminf_{k \to +\infty} \|\delta^k\| = \liminf_{k \to +\infty} \|\Delta^k\| = 0.$$

Proof Let $j \in \{1, 2, \ldots, n\}$ be the index of a variable. Suppose by way of contradiction that there exists a positive integer ρ for which the mesh index satisfies $r_j^k > -\rho$ for all $k \geq 0$. It follows that the mesh size parameter $\delta_j^k = \delta_j^0(\beta_j)^{r_j^k - |r_j^k|}$ is bounded above by δ_j^0 and bounded below by $\delta_j^0(\beta_j)^{-2\rho}$. Consequently, δ_j^k takes its value from the finite set $\{\delta_j^0(\beta_j)^{-2\rho}, \delta_j^0(\beta_j)^{2-2\rho}, \ldots, \delta_j^0\}$ regardless of the iteration number $k \geq 0$.

But since $(\beta_j)^2$ is a rational number, there exists two integers p_j and q_j such that $(\beta_j)^{-2} = \frac{p_j}{q_j}$. It follows that for every $k \geq 0$,

$$\delta_j^k \in \left\{ \delta_j^0 \left(\frac{p_j}{q_j}\right)^\rho, \delta_j^0 \left(\frac{p_j}{q_j}\right)^{\rho-1}, \ldots, \delta_j^0 \right\} = \left\{ \frac{\delta_j^0}{(q_j)^\rho} (p_j)^{\rho-l} (q_j)^l : l \in \{0, 1, \ldots, \rho\} \right\}$$

and therefore, there exists some finite integer ζ_j^k in $\{(p_j)^{\rho-l}(q_j)^l : l \in \{0, 1, \ldots, \rho\}\}$ such that $\delta_j^k = \frac{\delta_j^0}{(q_j)^\rho} \zeta_j^k$. Any trial point $t \in M^k$ satisfies Eq. (3) and thus

$$t_j = x_j^0 + \sum_{\ell=0}^{k} \delta_j^\ell z_j^\ell = x^0 + \frac{\delta_j^0}{(q_j)^\rho} \sum_{\ell=0}^{k} \zeta_j^\ell z_j^\ell$$

for some initial point $x^0 \in V^0$. In other words, the possible values taken by t_j belong to the unbounded enumerable set $\left\{ x_j^0, x_j^0 \pm \frac{\delta_j^0}{(q_j)^\rho}, x_j^0 \pm \frac{2\delta_j^0}{(q_j)^\rho}, \ldots \right\}$ regardless of the iteration number k. But since it is assumed that all iterates produced by the MADS instance belong to a bounded set, then the trial points can take only finitely many different values in $\mathbb{R}^n$.

This contradicts the hypothesis that the mesh size parameter is bounded away from zero, implying, from the MADS PARAMETER UPDATE rule in Algorithm 1, that there are infinitely many successful iterations, and infinitely many different incumbent solutions. □

The second step of the convergence analysis studies optimality conditions at some accumulation points of the sequences of mesh local optimizers x^k. The following definition introduces refining sequences, refined points and refining directions.

Definition 6.2 Let U be the indices of the unsuccessful iterations. If K is an infinite subset of U such that $\lim_{k \in K} \|\delta^k\| = 0$ and the sequence $\{x^k\}_{k \in K}$ converges to some $x^* \in \mathbb{R}^n$, then the set of poll centers $\{x^k\}_{k \in K}$ is said to be a refining sequence, and x^* a refined point. Furthermore, if for each $k \in K$ there is a poll point $x^k + \mathrm{diag}(\delta^k)d^k$ for some direction $d^k \in D^k$ such that the sequence of normalized directions $\left\{ \frac{v^k}{\|v^k\|} \right\}_{k \in K}$ (with $v^k = \mathrm{diag}(\delta^k)d^k$) converges to some $v^* \in \mathbb{S}^n$, then v^* is called a refining direction.

Refined points are limits of mesh local optimizers on meshes that get infinitely fine.

Proposition 6.3 There exist a refining sequence and a refined point.

Proposition 6.1 ensures that there exists an infinite subset of indices K for which $\lim_{k \in K} \|\delta^k\| = 0$. The mesh size parameters are reduced only at the unsuccessful iterations, and therefore, it can be assumed without any loss of generality that $K \subseteq U$. Furthermore, all trial points are assumed to lie in a bounded set, and consequently we can also assume that the sequence $\{x^k\}_{k \in K}$ is convergent. □

The next result shows that the Clarke generalized derivative at a refined point x^* is nonnegative in a refining direction v^* belonging to the hypertangent cone to the

domain Ω. The definition of the Clarke derivative $f^{\circ}(x^*; v^*)$ and of the hypertangent $T_{\Omega}^{H}(x^*)$ and Clarke tangent $T_{\Omega}^{Cl}(x^*)$ cones can be found in Audet and Dennis (2006).

Theorem 6.4 Let $x^* \in \Omega$ be a refined point, with $v^* \in T_{\Omega}^{H}(x^*)$ a refining direction. If f is Lipschitz near x^* then $f^{\circ}(x^*; v^*) \geq 0$.

Proof Let K be a subset of indices of unsuccessful iterations such that $\{x^k\}$ is a refining sequence converging to $x^* \in \Omega$ and $v^* \in T_{\Omega}^{H}(x^*)$ a refining direction.

By definition of the hypertangent cone, and since $\|v^k\| \leq \|\Delta^k\| \to 0$ for $k \in K$, then $x^k + v^k \in \Omega$ for all k sufficiently large. Therefore, since polling at $x^k + v^k$ was unsuccessful it follows that

$$
\begin{aligned}
f^{\circ}(x^*; v^*) &= \limsup_{\substack{y \to x^*, \ y \in \Omega \\ t \downarrow 0, \ y + tv^* \in \Omega}} \frac{f(y + tv^*) - f(y)}{t} \\[2mm]
&= \limsup_{\substack{y \to x^*, \ y \in \Omega, w \to v^* \\ t \downarrow 0, \ y + tw \in \Omega}} \frac{f(y + tw) - f(y)}{t} \\[2mm]
&\qquad \text{by Prop 3.9 of Audet and Dennis (2006)} \\[2mm]
&\geq \limsup_{\substack{k \in K, y = x^k, \\ w = \frac{v^k}{\|v^k\|}, t = \|v^k\|}} \frac{f(y + tw) - f(y)}{t} \\[2mm]
&= \limsup_{k \in K} \frac{f(x^k + v^k) - f(x^k)}{\|v^k\|} \geq 0.
\end{aligned}
$$

$\square$

The next result constitutes the main convergence result. It states that if the mechanism used to produce the poll directions is sufficiently rich to generate a dense set of refining directions, then, under the constraint qualification that the hypertangent cone to Ω at a refined point x^* is non-empty, then x^* is a Clarke stationary point of the optimization problem.

Theorem 6.5 If the set of refining directions is dense in $\mathbb{S}$, and if f is Lipschitz near the refined point $x^* \in \Omega$ where $T_{\Omega}^{H}(x^*) \neq \emptyset$, then x^* is a Clarke stationary point, i.e., $f^{\circ}(x^*; v) \geq 0$ for every direction $v \in T_{\Omega}^{Cl}(x^*)$.

If the set of refining directions is dense in the sphere $\mathbb{S}$, then any normalize Clarke-tangent direction to Ω at a refined point x^* is the limit of refining directions. The result follows from Theorem 6.4. $\square$

References

Abramson MA, Audet C, Dennis JE Jr, Le Digabel S (2009) OrthoMADS: a deterministic MADS instance with orthogonal directions. SIAM J Optim 20(2):948–966

Adjengue L, Audet C, Ben I, Yahia IB (2014) A variance-based method to rank input variables of the mesh adaptive direct search algorithm. Optim Lett 8(5):1599–1610

Agte JS, Sobieszczanski-Sobieski J, Sandusky RRJ (1999) Supersonic business jet design through bilevel integrated system synthesis. In: Proceedings of the World Aviation Conference, volume SAE Paper No. 1999–01-5622, San Francisco, CA, 1999. MCB University Press, Bradford

Audet C (2004) Convergence results for generalized pattern search algorithms are tight. Optim Eng 5(2):101–122

Audet C, Béchard V, Le Digabel S (2008) Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. J Glob Optim 41(2):299–318

Audet C, Dennis JE Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. SIAM J Optim 17(1):188–217

Audet C, Dennis JE Jr (2009) A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM J Optim 20(1):445–472

Audet C, Dennis JE Jr, Le Digabel S (2008) Parallel space decomposition of the mesh adaptive direct search algorithm. SIAM J Optim 19(3):1150–1170

Audet C, Dennis JE Jr, Le Digabel S (2010) Globalization strategies for mesh adaptive direct search. Comput Optim Appl 46(2):193–215

Audet C, Ianni A, Le Digabel S, Tribes C (2014) Reducing the number of function evaluations in mesh adaptive direct search algorithms. SIAM J Optim 24(2):621–642

Box GEP, Muller ME (1958) A note on the generation of random normal deviates. Ann Math Stat 29(2):610–611

Conn AR, Le Digabel S (2013) Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optim Methods Softw 28(1):139–158

Conn AR, Scheinberg K, Vicente LN (2009) Introduction to derivative-free optimization. MOS-SIAM series on optimization. SIAM, Philadelphia

Coope ID, Price CJ (2000) Frame-based methods for unconstrained optimization. J Optim Theory Appl 107(2):261–274

Cramer EJ, Dennis JE Jr, Frank PD, Lewis RM, Shubin GR (1994) Problem formulation for multidisciplinary optimization. SIAM J Optim 4(4):754–776

Dennis Jr JE, Schnabel RB (1996) Numerical methods for unconstrained optimization and nonlinear equations. Prentice-Hall, Englewood Cliffs, NJ, 1983. Reissued in 1996 by SIAM Publications, Philadelphia, as, Vol. 16 in the series Classics in Applied Mathematics

Fletcher R (1980) Practical methods of optimization, volume 1: unconstrained optimization. Wiley, Chichester

García-Palomares UM, Rodríguez JF (2002) New sequential and parallel derivative-free algorithms for unconstrained optimization. SIAM J Optim 13(1):79–96

Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic Press, London

Gould N, Orban D, Toint P (2015) CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput Optim Appl 60(3):545–557. Code available at http://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki

Halton JH (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer Math 2(1):84–90

Hough PD, Kolda TG, Torczon V (2001) Asynchronous parallel pattern search for nonlinear optimization. SIAM J Sci Comput 23(1):134–156

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev 45(3):385–482

Kolda TG, Lewis RM, Torczon V (2006) Stationarity results for generating set search for linearly constrained optimization. SIAM J Optim 17(4):943–968

Le Digabel S (2011) Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. ACM Trans Math Softw 37(4):44:1–44:15

Liseikin VD (2009) Grid generation methods. Springer, Berlin

Lucidi S, Sciandrone M (2002) On the global convergence of derivative-free methods for unconstrained optimization. SIAM J Optim 13(1):97–116

Marsaglia G (1972) Choosing a point from the surface of a sphere. Ann Math Stat 43(2):645–646

Martínez JM, Sobral FNC (2013) Constrained derivative-free optimization on thin domains. J Glob Optim 56(3):1217–1232

McKay MD, Beckman RJ, Conover WJ (1979) 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

Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100

Moré JJ, Wild SM (2009) Benchmarking derivative-free optimization algorithms. SIAM J Optim 20(1):172–191

Nocedal J, Wright SJ (1999) Numerical optimization. Springer Series in Operations Research. Springer, New York

Papalambros PY, Wilde DJ (2000) Principles of optimal design: modeling and computation, 2nd edn. Cambridge University Press, Cambridge

Torczon V (1997) On the convergence of pattern search algorithms. SIAM J Optim 7(1):1–25

Tosserams S, Kokkolaras M, Etman LFP, Rooda JE (2010) A non-hierarchical formulation of analytical target cascading. J Mech Des 132(5):051002-1–051002-13

Van Dyke B, Asaki TJ (2013) Using QR decomposition to obtain a new instance of mesh adaptive direct search with uniformly distributed polling directions. J Optim Theory Appl 159(3):805–821

Van Loan CF (1985) How near is a stable matrix to an unstable matrix. Contemp Math 47:465–478

Vicente LN, Custódio AL (2012) Analysis of direct searches for discontinuous functions. Math Programm 133(1–2):299–325