Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 12, No. 4, pp. 813–832, October 2016

GENERATING POLL DIRECTIONS FOR MESH ADAPTIVE DIRECT SEARCH WITH REALIZATIONS OF A UNIFORMLY DISTRIBUTED RANDOM ORTHOGONAL MATRIX*

Árpád Bűrmen† and Tadej Tuma

Abstract: Mesh Adaptive Direct Search (MADS) is a family of derivative-free black-box optimization algorithms that rely on an asymptotically dense set of directions for exploring the neighborhood of the incumbent solution. The convergence properties of the algorithm are guaranteed by the poll step where the exploration of the neighborhood takes place. Because nonsmooth and constrained problems have points where the set of descent directions can be significantly smaller than an open half-space we expect to improve the algorithm's performance by using uniformly distributed directions in the poll step. We propose a simple approach based on transforming a prototype set of directions with a sequence of realizations of a uniformly distributed random orthogonal matrix. Numerical results on test functions and two real-world integrated circuit optimization problems demonstrate the advantage of the proposed approach.

Key words: *mesh adaptive direct search, derivative-free algorithms, random matrices, uniform distribution.*

Mathematics Subject Classification: *90C56, 90C30, 65K05, 15B52.*

1 Introduction

The derivatives of black-box constrained optimization problems of the form

$$\min_{\mathbf{x}\in\Omega\subseteq\mathbb{R}^n} f(\mathbf{x})$$

are often unavailable or unreliable. Several optimization algorithms were devised for solving such problems. One of them is the Mesh Adaptive Direct Search class of algorithms (MADS) proposed in [3]. The algorithm relies on poll steps that probe the function subject to optimization in the neighborhood of the best-yet solution (also referred to as the incumbent). The directions along which MADS explores the search space form a dense sequence of points on the surface of the unit sphere. Due to this the algorithm possesses some favorable convergence properties for nonsmooth functions [1, 3, 26] in the Clarke sense [11] and in the Rockafellar sense [23]. MADS can take into account constraints by using the extreme barrier approach where $f(\mathbf{x})$ is replaced with

$$f_\Omega(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \mathbf{x}\in\Omega \\ \infty, & \text{otherwise.} \end{cases} \tag{1.1}$$

*The research was co-funded by the Ministry of Education, Science, and Sport (Ministrstvo za Šolstvo, Znanost in Šport) of the Republic of Slovenia through the programme P2-0246 Algorithms and optimization methods in telecommunications.

†Corresponding author.

In the proposed algorithm this approach is used for handling the constraints. Consequently the initial point must be feasible (i.e. $\mathbf{x}_0 \in \Omega$).

Every poll step explores the search space along a finite number of directions. If it fails to find descent the incumbent solution is also referred to as a minimal frame center. The set of normalized poll directions across at least one convergent subsequence of minimal frame centers must be dense on the unit sphere. This requirement, however, does not prevent normalized poll directions from concentrating in certain regions of the unit sphere. Nonsmooth and constrained problems often have points where the cone of descent directions is significantly smaller than an open half-space. Probability of guessing a poll step that improves the incumbent solution in the neighborhood of such points can be low. Consequently the use of unevenly distributed normalized poll directions can result in slower convergence.

In MADS the set of poll directions applied in the neighborhood of the incumbent must positively span [14] the search space [3]. This imposes an additional requirement on the poll step generation algorithm. The first published instance of MADS [3] generated the poll directions in a random manner. The positive spanning property was enforced by obtaining the directions from a lower-triangular matrix (LTMADS). In [2] it was pointed out that the distribution of normalized poll directions in LTMADS is far from uniform. As a remedy a new algorithm was proposed deemed OrthoMADS. The set of poll directions in OrthoMADS is a maximal positive basis that comprises an orthogonal basis for the search space and its negative. The search directions are generated from the output of a quasi-random generator (Halton sequence [18]) which makes the algorithm deterministic. A scaled Householder reflection matrix is built from a random direction and its columns together with their negatives are used as scaled poll directions.

Although OrthoMADS distributes the normalized poll directions more uniformly than LTMADS, the distribution rapidly deteriorates with increasing problem dimension. This was first pointed out in [25]. There are two reasons for this. First of all the normalized random direction in OrthoMADS is not uniformly distributed on the unit sphere. The uniformity of the distribution deteriorates as the problem dimension increases. Secondly, the columns of the Householder matrix exhibit a nonuniform distribution that progressively peaks around coordinate directions as the problem dimension increases. In [25] a remedy was proposed that involves generating the search directions by means of QR decomposition applied to a random matrix (thus the name QRMADS). The columns of the resulting orthogonal matrix and their negatives form a maximal positive basis for the search space. The title of the paper [25] suggests the obtained distribution of normalized poll directions is uniform. No proof of this claim was given. The distribution was visualized with 2-dimensional scatter plots of projected normalized poll directions. Although the plots give the impression of uniformity we show (section 6) the generated directions are not uniformly distributed on the unit sphere. Nevertheless the distribution of normalized poll directions in QRMADS is significantly more uniform than in OrthoMADS.

With the above mentioned shortcomings in mind we propose a different approach based on a sequence of uniformly distributed orthogonal matrices [24] and a prototype set of search directions that positively spans the search space. It guarantees that the members of the set of normalized poll directions are uniformly distributed on the unit sphere. Unlike QRMADS, where every positive basis comprises an orthogonal basis and its negative, the proposed approach is capable of generating positive bases with arbitrary cardinality.

Because MADS algorithms base their convergence properties on selecting the evaluated points from progressively finer meshes, the poll steps must be forced to lie on a grid. In OrthoMADS this was achieved by scaling and rounding the generated random direction so that the elements of the obtained Householder matrix were integers. In [25] and in the proposed algorithm every poll step is rounded to the nearest mesh point. Rounding deteriorates the poll directions so that they are no longer orthogonal. It can even cause the resulting set to no longer positively span the search space. To avoid this situation a bound must be imposed on the length of the poll step with respect to the mesh density. This bound was estimated in [25] using singular value decomposition of the matrix representing the unrounded poll steps. We propose a more simple approach that is based on [8] and yields the same result. An additional advantage of the proposed approach is that it can easily be applied to positive spanning sets of arbitrary cardinality.

In section 2 an overview of the MADS algorithmic framework is given. Section 3 gives an introduction to uniformly distributed orthogonal matrices. The algorithm for generating scaled poll directions and its properties are the subject of section 4. Section 5 introduces the proposed algorithm deemed UniMADS and shows it is a valid MADS instance. Numerical results are given in Section 6.

Notation. We denote the coordinate vectors of the standard orthogonal basis by $\mathbf{e}_i$. The element of matrix $\mathbf{A}$ from the i-th row of the j-th column is denoted by a_{ij}. $\|\mathbf{a}\|$ denotes the Euclidean norm of $\mathbf{a}$. The smallest integer not smaller than a is denoted by $\lceil a \rceil$. Sequences are denoted by $\{a_i\}_{i=0}^{\infty}$. $N(a, \sigma^2)$ is the normal distribution with mean value a and variance σ^2. $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{N}$ denote the set of real, integer, rational, and natural numbers, respectively. The rounding operator $\mathcal{R}$ choses one of the points from set $\mathcal{G}$ that are closest to $\mathbf{x}$.

$$\mathcal{R}(\mathbf{x}, \mathcal{G}) \in \arg\min_{\mathbf{v}\in\mathcal{G}} \|\mathbf{x} - \mathbf{v}\|. \tag{1.2}$$

When arg min results in multiple points (with the same distance from $\mathbf{x}$) a tie breaking rule is used that chooses a unique point for every $\mathbf{x}$ and every $\mathcal{G}$.

2 Overview of MADS

Let $\mathcal{S}_k$ be the set of points visited by the algorithm at the start of iteration k including the initial point $\mathbf{x}_0 \in \Omega$. Points visited at the k-th iteration lie on a mesh defined as

$$\mathcal{M}_k = \cup_{x\in\mathcal{S}_k} \{\mathbf{x} + \mathbf{p} : \mathbf{p} \in \mathcal{G}_k\}. \tag{2.1}$$

$\mathbf{x}_k$ is the incumbent solution in k-th iteration (i.e. the feasible point with the lowest value of f found in iterations $1, ..., k-1$). The grid $\mathcal{G}_k$ comprises all steps available to the algorithm in k-th iteration. It is defined as

$$\mathcal{G}_k = \{\Delta_k^m \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D}\}, \tag{2.2}$$

where $\Delta_k^m > 0$ is the mesh size parameter. The n_D columns of $\mathbf{D}$ must positively span $\mathbb{R}^n$ [14] and $\mathbf{D}$ must be a product of the form $\mathbf{D} = \mathbf{G}\mathbf{Z}$ where $\mathbf{G}$ is a real $n \times n$ matrix and $\mathbf{Z}$ is an integer $n \times n_\mathrm{D}$ matrix. Convergence of the algorithm is enforced by means of poll steps. Let $\mathcal{D}_k$ denote the set of scaled poll directions at k-th iteration. Every $\mathcal{D}_k$ is a finite subset of $\mathcal{G}_k$. Let $\Omega \subseteq \mathbb{R}^n$ denote the feasible region defined by the constraints.

The MADS algorithmic framework can now be written as

Algorithm 2.1. MADS framework

1. Choose $\mathbf{x}_0 \in \Omega$, $\Delta_0^m \leq \Delta_0^p$, $\mathbf{D}$, $1 < \tau \in \mathbb{Q}$, $-1 \geq w^- \in \mathbb{Z}$, and $0 \leq w^+ \in \mathbb{Z}$.
   Set $k := 0$ and $f_0 := f_\Omega(\mathbf{x}_0)$.

2. Evaluate f_Ω on a finite subset of $\mathcal{M}_k$.
   If a point $\mathbf{x}'$ is found satisfying $f_\Omega(\mathbf{x}') < f_k$, go to step 4.

3. Evaluate f at points $\mathbf{x}' = \mathbf{x}_k + \mathbf{d}$ for $\mathbf{d} \in \mathcal{D}_k$ until $\mathbf{x}'$ is found satisfying $f_\Omega(\mathbf{x}') < f_k$ or the set $\mathcal{D}_k$ is exhausted.
4. Let $\mathbf{x}'$ denote the point with lowest value of f_Ω obtained in steps 2-3.
If $f_\Omega(\mathbf{x}') \le f_k$ let $\mathbf{x}_{k+1} := \mathbf{x}'$, $f_{k+1} := f_\Omega(\mathbf{x}')$ and
choose $w_k \in \mathbb{Z}$ satisfying $0 \le w_k \le w^+$.
Otherwise let $\mathbf{x}_{k+1} := \mathbf{x}_k$, $f_{k+1} := f_k$, and
choose $w_k \in \mathbb{Z}$ satisfying $w^- \le w_k < 0$.
5. Let $\Delta_{k+1}^m := \tau^{w_k}\Delta_k^m$ and choose $\Delta_{k+1}^p \ge \Delta_{k+1}^m$.
6. Set $k := k+1$ and go back to step 2.

Steps 2 and 3 are also referred to as the search and the poll step. Δ_k^p is the poll size parameter. Let $\mathcal{D}'_k = \{\mathbf{d}/\|\mathbf{d}\| : \mathbf{d} \in \mathcal{D}_k\}$ be the set of normalized poll directions corresponding to $\mathcal{D}_k$. A refining subsequence $\{\mathbf{x}_k\}_{k\in\mathcal{K}}$ is every subsequence for which Δ_k^p converges to zero. For Lipschitz continuous functions the subgradient of f at the limit points of a refining subsequence includes $\mathbf{0}$ if the following requirements are satisfied:

(A) there exists $C > 0$ such that for all $\mathbf{d} \in \mathcal{D}_k$ we have $\|\mathbf{d}\| \le C\Delta_k^p$,

(B) $\lim_{k\to\infty} \Delta_k^p = 0$ if and only if $\lim_{k\to\infty} \Delta_k^m = 0$,

(C) limit points of $\mathcal{D}'_k$ in the sense of [13] are positive spanning sets,

(D) the set $\cup_{k\in\mathcal{K}}\mathcal{D}'_k$ is dense on the unit sphere.

If f is continuously differentiable the generalized gradient is a set with a single element ∇f and the usual result (convergence to a stationary point) is obtained in the unconstrained case. Details of the convergence analysis can be found in [3].

3 Uniformly Distributed Random Orthogonal Matrices

The notion of a random distribution over sets can be defined by means of a measure [10]. A measure μ maps $\mathcal{A} \subseteq \mathcal{A}_0$ to $\mathbb{R}$ and satisfies 3 properties. It is non-negative ($\mu(\mathcal{A}) \ge 0$), it is zero for an empty set ($\mu(\emptyset) = 0$), and is countably additive ($\mu(\cup_{i\in\mathcal{I}}\mathcal{A}_i) = \sum_{i\in\mathcal{I}} \mu(\mathcal{A}_i)$ for any countable $\mathcal{I}$ and pairwise disjoint $\mathcal{A}_i$). Let X be a random variable with sample space $\mathcal{A}$ and let $\mathcal{B} \subseteq \mathcal{A}$. We can define the probability that X takes a value from $\mathcal{B}$ as

$$P(X \in \mathcal{B}) = \mu(\mathcal{B})/\mu(\mathcal{A}). \tag{3.1}$$

Thus the measure μ defines the probability distribution of X over $\mathcal{A}$. It is common to say that X is a random variable from measure μ on $\mathcal{A}$ or shortly X is distributed according to μ.

In $\mathbb{R}^n$ the most commonly used measure is the Lebesgue measure [10]. The Lebesgue measure (λ) of a box in $\mathbb{R}^n$ (i.e. the Cartesian product of n intervals $[a_i, b_i]$) is defined as $\prod_{i=1}^n (b_i - a_i)$. Random variables from λ are uniformly distributed over $\mathcal{A}$. The Lebesgue measure is invariant to translations and orthogonal linear transformations.

$$\begin{aligned} \lambda(\mathcal{A}) &= \lambda(\{\mathbf{x} + \mathbf{c} : \mathbf{x} \in \mathcal{A}\}), \quad \mathbf{c} \in \mathbb{R}^n, \\ \lambda(\mathcal{A}) &= \lambda(\{\mathbf{Q}\mathbf{x} : \mathbf{x} \in \mathcal{A}\}), \quad \mathbf{Q} \in \mathcal{O}_n. \end{aligned}$$

Orthogonal transformations can be represented as orthogonal matrices. An orthogonal matrix $\mathbf{Q}$ has the property $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$. The set of orthogonal matrices ($\mathcal{O}_n$) is a subset of $\mathbb{R}^{n^2}$.

The uniform distribution over $\mathcal{O}_n$ is defined by the Haar measure (μ_{H}) [10] and is invariant to orthogonal transformations. Let $\mathcal{A}$ and $\mathbf{Q}$ denote any subset and any element of $\mathcal{O}_n$, respectively. If $\mu_{\mathrm{H}}(\mathcal{A}) = \mu_{\mathrm{H}}\left(\{\mathbf{QA} : \mathbf{A} \in \mathcal{A}\}\right)$ holds then μ_{H} is left invariant. A right-invariant Haar measure satisfies $\mu_{\mathrm{H}}(\mathcal{A}) = \mu_{\mathrm{H}}\left(\{\mathbf{AQ} : \mathbf{A} \in \mathcal{A}\}\right)$. $\mathcal{O}_n$ is a compact group and therefore unimodular. Unimodular groups carry a Haar measure that is both left- and right-invariant. A direct consequence of the left- and right-invariance is the following corollary.

Corollary 3.1. *Let $\mathbf{Q} \in \mathcal{O}_n$ be a given orthogonal matrix and let $\mathbf{O} \in \mathcal{O}_n$ be distributed according to μ_{H}. Then $\mathbf{QO}$ and $\mathbf{OQ}$ are also distributed according to μ_{H}.*

QR decomposition factors matrix $\mathbf{A}$ into a product of an orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$. It is, however, not unique in the sense that any pair $(\mathbf{QD}, \mathbf{DR})$ is also a valid QR decomposition of $\mathbf{A}$ if $\mathbf{D}$ is a diagonal matrix with diagonal elements coming from $\{-1, 1\}$.

Definition 3.2. Orthogonal matrix $\overline{\mathbf{Q}}$ and upper triangular matrix $\overline{\mathbf{R}}$ are the unique QR decomposition of matrix $\mathbf{A}$ iff $\mathbf{A} = \overline{\mathbf{Q}}\,\overline{\mathbf{R}}$ and the diagonal elements of $\overline{\mathbf{R}}$ are nonnegative.

Let $\mathbf{O}$ be an arbitrary orthogonal matrix. Then the unique QR decomposition of $\mathbf{OA}$ is given by $\mathbf{O}\overline{\mathbf{Q}}$ and $\overline{\mathbf{R}}$. The unique QR decomposition can be obtained with the Gram-Schmidt orthogonalization algorithm. Because this algorithm is numerically unstable the decomposition is usually obtained by means of Householder reflections. The sign of the diagonal elements of the resulting $\mathbf{R}$ varies and the decomposition is no longer unique. Uniqueness can, however, be restored by choosing $\mathbf{D}$ in such manner that $d_{ii} = \mathrm{sign}(r_{ii})$.

Using Corollary 3.1 along with QR decomposition one can devise an algorithm for generating a sequence of random orthogonal matrices from the Haar measure on $\mathcal{O}_n$ (see [24]). Let $\mathbf{N}$ denote an $n \times n$ random matrix whose elements are independent random variables distributed according to $N(0, 1)$. The joint probability density for such a matrix is $(2\pi)^{-n^2/2} \exp(\mathrm{tr}(\mathbf{N}^{\mathrm{T}}\mathbf{N})/2)$. Let $\mathbf{Q}$ denote an arbitrary orthogonal matrix. Because $(\mathbf{QN})^{\mathrm{T}}(\mathbf{QN}) = \mathbf{N}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{QN} = \mathbf{N}^{\mathrm{T}}\mathbf{N}$ the distribution is invariant to left multiplication by $\mathbf{Q}$. The invariance to right multiplication by $\mathbf{Q}$ follows from $\mathrm{tr}(\mathbf{N}^{\mathrm{T}}\mathbf{N}) = \mathrm{tr}(\mathbf{N}\mathbf{N}^{\mathrm{T}})$. Let $\mathbf{N} = \overline{\mathbf{Q}}\,\overline{\mathbf{R}}$ denote the unique QR decomposition of $\mathbf{N}$. The unique QR decomposition of $\mathbf{QN}$ is then $(\mathbf{Q}\overline{\mathbf{Q}})\overline{\mathbf{R}}$. Because $\mathbf{QN}$ has the same distribution as $\mathbf{N}$, $\mathbf{Q}\overline{\mathbf{Q}}$ must have the same distribution as $\overline{\mathbf{Q}}$. Consequently $\overline{\mathbf{Q}}$ must be distributed according to μ_{H}. The following Lemma summarizes some of the properties of such a matrix.

Lemma 3.3. *Let $\mathbf{O} \in \mathcal{O}_n$ be a random matrix distributed according to μ_{H}, $\mathbf{a}$ an arbitrary unit vector, $\mathbf{Q}_1$ a random orthogonal matrix, and $\mathbf{a}_1$ a random unit vector. Suppose $\mathbf{Q}_1$ and $\mathbf{a}_1$ are independent from $\mathbf{O}$. Then*

1. *$\mathbf{O}^{\mathrm{T}}$ is distributed according to μ_{H},*

2. *$\mathbf{Q}_1\mathbf{O}$ and $\mathbf{O}\mathbf{Q}_1$ are distributed according to μ_{H},*

3. *$\mathbf{Oa}$ and $\mathbf{a}^{\mathrm{T}}\mathbf{O}$ are uniformly distributed on the unit sphere,*

4. *$\mathbf{O}\mathbf{a}_1$ and $\mathbf{a}_1^{\mathrm{T}}\mathbf{O}$ are uniformly distributed on the unit sphere, and*

5. *columns and rows of $\mathbf{O}$ are uniformly distributed on the unit sphere.*

Proof. 1: Let $\mathbf{Q}$ be an arbitrary orthogonal matrix. We can express $\mathbf{Q}\mathbf{O}^{\mathrm{T}} = (\mathbf{O}\mathbf{Q}^{\mathrm{T}})^{\mathrm{T}}$. Matrix $\mathbf{O}\mathbf{Q}^{\mathrm{T}}$ has the same distribution as $\mathbf{O}$. Therefore $(\mathbf{O}\mathbf{Q}^{\mathrm{T}})^{\mathrm{T}}$ must have the same distribution as $\mathbf{O}^{\mathrm{T}}$. Consequently $\mathbf{O}^{\mathrm{T}}$ is distributed according to μ_{H}.

2: Due to independence the conditional distribution $p(\mathbf{O}|\mathbf{Q}_1)$ is identical to $p(\mathbf{O})$. Corollary 3.1 implies the distributions $p(\mathbf{Q}_1\mathbf{O}|\mathbf{Q}_1)$ and $p(\mathbf{O}\mathbf{Q}_1|\mathbf{Q}_1)$ are identical to the distribution of $\mathbf{O}$. Matrix $\mathbf{O}$ is distributed according to μ_{H}. The conditional distribution $p(\mathbf{Q}_1\mathbf{O}|\mathbf{Q}_1) = p(\mathbf{O}\mathbf{Q}_1|\mathbf{Q}_1) = p(\mathbf{O})$ does not depend on $\mathbf{Q}_1$ so we can conclude $p(\mathbf{Q}_1\mathbf{O}) = p(\mathbf{O}\mathbf{Q}_1) = p(\mathbf{O})$. Note that the result holds for any distribution of $\mathbf{Q}_1$.

3: From $\mathbf{Q}(\mathbf{O}\mathbf{a}) = (\mathbf{Q}\mathbf{O})\mathbf{a}$ and the fact the $\mathbf{Q}\mathbf{O}$ is distributed according to μ_{H} we conclude that the distribution of vector $\mathbf{O}\mathbf{a}$ is identical to the distribution of $\mathbf{Q}(\mathbf{O}\mathbf{a})$. Therefore $\mathbf{O}\mathbf{a}$ must be uniformly distributed on the unit sphere. The uniform distribution of $\mathbf{a}^{\mathrm{T}}\mathbf{O}$ follows from the fact that $\mathbf{O}^{\mathrm{T}}$ is distributed according to μ_{H}.

4: The proof relies on the uniform distribution of $\mathbf{O}\mathbf{a}$, $\mathbf{a}^{\mathrm{T}}\mathbf{O}$, and the independence of $\mathbf{a}_1$. The argument is similar to the one used to prove 2.

5: Substituting $\mathbf{e}_i$ for $\mathbf{a}$ in property 3 yields the desired result. □

Lemma 3.3 provides the theoretical background for generating poll sets with uniformly distributed poll directions.

4 Generating the Set of Scaled Poll Directions

The set of scaled poll directions $\mathcal{D}_k$ must positively span $\mathbb{R}^n$. There are many different ways to construct a poll set, but most commonly it consists of n vectors ($\mathbf{d}_i$, $i = 1, .., n$) that span $\mathbb{R}^n$ and

1. their negatives $\mathbf{d}_{n+i} = -\mathbf{d}_i$ for $i = 1, .., n$ or
2. a nonzero vector $\mathbf{d}_{n+1}$ from the open cone -$\sum_{i=1}^{n} \alpha_i \mathbf{d}_i$, $\alpha_i > 0$.

In the former case $\mathcal{D}_k$ is a maximal positive basis while in the latter case it is a minimal positive basis. In order to spread the directions as uniformly as possible the maximal positive basis is often chosen in such manner that vectors $\mathbf{d}_i$, $i = 1, ..., n$ are mutually orthogonal. The angle between two vectors in such a poll set is either $\pi/2$ or π. Minimal positive bases with the analogous properties are referred to as regular n-simplices if all basis vectors have unit length. The angle between any pair of vectors $\mathbf{d}_i \neq \mathbf{d}_j$ in a regular n-simplex is equal to $\arccos(-n^{-1})$. See [12] on how to generate a regular n-simplex. The idea of using a regular simplex as the poll direction set was used in [15].

Let $\{\mathbf{N}_i\}_{i=0}^{\infty}$ denote a sequence of realizations of a random matrix whose elements are independent identically distributed random variables distributed according to $N(0, 1)$. Let $\mathcal{P} = \{\mathbf{b}_1, ..., \mathbf{b}_m\}$ positively span $\mathbb{R}^n$. This set is referred to as the prototype set of poll directions. The following algorithm generates a sequence of sets of poll directions for which the corresponding normalized unrounded poll directions are uniformly distributed on the unit sphere.

Algorithm 4.1. Generating the k-th member of a sequence of sets of scaled poll directions.

1. QR decompose $\mathbf{N}_{t_k}$ into $\mathbf{Q}_{t_k}$ and $\mathbf{R}_{t_k}$.
2. Construct a diagonal matrix $\mathbf{D}_{t_k}$ with with elements $d_{ii} = \mathrm{sign}(r_{ii})$.
3. $\mathbf{O}_{t_k} := \mathbf{Q}_{t_k}\mathbf{D}_{t_k}$,
4. $\mathcal{U}_k := \{\Delta_k^p \mathbf{O}_{t_k}\mathbf{p} : \mathbf{p} \in \mathcal{P}\}$.
5. $\mathcal{D}_k := \{\mathcal{R}(\mathbf{u}, \mathcal{G}_k) : \mathbf{u} \in \mathcal{U}_k\}$.

Steps 1-3 generate a random orthogonal matrix $\mathbf{O}_{t_k}$ distributed according to μ_{H} (see section 3). This matrix is used for transforming the members of the prototype set of poll directions in step 4. Every normalized member of the resulting set $\mathcal{U}_k$ is a realization of a random vector with uniform distribution on the unit sphere (Lemma 3.3). Finally in step 5 the members of $\mathcal{U}_k$ are rounded to the nearest points on grid $\mathcal{G}_k$ to obtain the set of scaled poll directions $\mathcal{D}_k$. Due to rounding the subset of n orthogonal vectors from $\mathcal{U}$ is transformed into a set of almost orthogonal vectors. Index t_k assigns a member of the sequence $\{\mathbf{N}_i\}_{i=0}^{\infty}$ to k-th iteration of the MADS framework. Its role will become more clear in section 5.

Although the MADS algorithmic framework is very flexible when it comes to defining the mesh points, most implementations use a simple orthogonal grid generated by $\mathbf{D} = [\mathbf{I}, -\mathbf{I}]$. In the remainder of the paper we assume this type of grid is used. Finding the point in $\mathcal{G}_k$ closest to some $\mathbf{d}$ is equivalent to rounding the components of $\mathbf{d}/\Delta_k^m$ to the nearest integers. The built-in rounding operation of the mathematical library includes a tie-breaking rule for (1.2). Rounding the n components of a vector to nearest integers is equivalent to adding an error vector that is bounded in norm by $n^{1/2}/2$. Rounding $\mathbf{d}$ to the nearest member of $\mathcal{G}_k$ is therefore equivalent to adding an error vector δ for which the following bound holds.

$$\|\delta\| \leq \delta_0(\Delta_k^m) = \Delta_k^m n^{1/2}/2. \tag{4.1}$$

Rounding can deteriorate a positive spanning set to such extent that it no longer positively spans $\mathbb{R}^n$. The cosine measure $\mathrm{cm}(\mathcal{U}_k)$ [21] indicates how good is set $\mathcal{U}_k$ at positively spanning $\mathbb{R}^n$. This measure is greater than zero if and only if $\mathcal{U}_k$ positively spans $\mathbb{R}^n$. Rounding the members of $\mathcal{U}_k$ to the nearest points from $\mathcal{G}_k$ can decrease its cosine measure. Let $\mathbf{u}_{\min}$ denote the member of $\mathcal{U}_k$ with the smallest Euclidean norm. In [8] the following relation was derived

$$\mathrm{cm}(\mathcal{D}_k) \geq \frac{\mathrm{cm}(\mathcal{U}_k) - \delta_0(\Delta_k^m)/\|\mathbf{u}_{\min}\|}{1 + \delta_0(\Delta_k^m)/\|\mathbf{u}_{\min}\|}. \tag{4.2}$$

Because $\mathbf{O}_{t_k}$ is an orthogonal matrix $\|\mathbf{u}_{\min}\| = \Delta_k^p \|\mathbf{p}_{\min}\|$ where $\mathbf{p}_{\min}$ denotes the member of $\mathcal{P}$ with the smallest Euclidean norm. The following Lemma establishes the condition under which all limit points of $\{\mathcal{D}'_k\}_{k=0}^{\infty}$ are positive spanning sets.

Lemma 4.2. *Assume a simple orthogonal grid is used and* $\mathrm{cm}(\mathcal{P}) > 0$. *If* $\Delta_k^m/\Delta_k^p \leq (1-\epsilon)/\gamma$, *with* $0 < \epsilon < 1$ *and* $\gamma = n^{1/2}/(2\mathrm{cm}(\mathcal{P})\|\mathbf{p}_{\min}\|)$ *then every limit point of* $\{\mathcal{D}'_k\}_{k=0}^{\infty}$ *is a positive spanning set.*

Proof. Because $\mathcal{U}_k$ is obtained from $\mathcal{P}$ by multiplication with an orthogonal matrix we have $\mathrm{cm}(\mathcal{U}_k) = \mathrm{cm}(\mathcal{P})$. From (4.2) it follows

$$\mathrm{cm}(\mathcal{D}'_k) = \mathrm{cm}(\mathcal{D}_k) \geq \frac{\mathrm{cm}(\mathcal{P}) - \Delta_k^m n^{1/2}/(2\Delta_k^p\|\mathbf{p}_{\min}\|)}{1 + \Delta_k^m n^{1/2}/(2\Delta_k^p\|\mathbf{p}_{\min}\|)} \geq \frac{\epsilon \mathrm{cm}(\mathcal{P})}{1 + (1-\epsilon)\mathrm{cm}(\mathcal{P})} > 0. \tag{4.3}$$

Because the cosine measure of $\mathcal{D}'_k$ is uniformly bounded away from 0 all limit points of $\{\mathcal{D}'_k\}_{k=0}^{\infty}$ are positive spanning sets. □

The cosine measure of a maximal positive basis comprising n orthogonal vectors and their negatives is $n^{-1/2}$. If the prototype set $\mathcal{P}$ is such a maximal positive basis we have $\|\mathbf{p}_{\min}\| = 1$ and $\gamma = n/2$. If, on the other hand, $\mathcal{P}$ is a regular simplex (with cosine measure n^{-1}, cf. [12]) we have $\gamma = n^{3/2}/2$.

The following Lemma is the basis for proving $\cup_{k=0}^{\infty}\mathcal{D}'_k$ is dense on the unit sphere.

Lemma 4.3. *Suppose* $\{\mathbf{d}_i\}_{i=0}^{\infty}$ *is a sequence of vectors such that* $\{\mathbf{d}_i/\|\mathbf{d}_i\|\}_{i=0}^{\infty}$ *is dense on the unit sphere. Let* $\mathbf{d}_i^{\mathrm{R}} = \mathcal{R}(\mathbf{d}_i, \mathcal{G}_i)$*. If* $\Delta_i^m/\|\mathbf{d}_i\| \to 0$ *then* $\{\mathbf{d}_i^{\mathrm{R}}/\|\mathbf{d}_i^{\mathrm{R}}\|\}_{i=0}^{\infty}$ *is also dense on the unit sphere.*

Proof. Given any $\epsilon > 0$ and any $\mathbf{x}$ satisfying $\|\mathbf{x}\| = 1$ there exists by assumption an infinite sequence of indices $\mathcal{K}$ such that that $\|\mathbf{x} - \mathbf{d}_k/\|\mathbf{d}_k\|\| < \epsilon/2$ for all $k \in \mathcal{K}$. Because the rounding error (δ_k) is proportional to Δ_k^m we have $\|\delta_k\|/\|\mathbf{d}_k\| \to 0$ and

$$\begin{aligned}
\left\|\frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} - \frac{\mathbf{d}_k^{\mathrm{R}}}{\|\mathbf{d}_k^{\mathrm{R}}\|}\right\| &= \left\|\frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} - \frac{\mathbf{d}_k + \delta_k}{\|\mathbf{d}_k + \delta_k\|}\right\| \\
&= \left\|\mathbf{d}_k\left(\frac{1}{\|\mathbf{d}_k\|} - \frac{1}{\|\mathbf{d}_k + \delta_k\|}\right) - \frac{\delta_k}{\|\mathbf{d}_k + \delta_k\|}\right\| \\
&\leq \|\mathbf{d}_k\|\left|\frac{\|\mathbf{d}_k + \delta_k\| - \|\mathbf{d}_k\|}{\|\mathbf{d}_k\|\|\mathbf{d}_k + \delta_k\|}\right| + \frac{\|\delta_k\|}{\|\mathbf{d}_k + \delta_k\|} \\
&\leq \frac{2\|\delta_k\|}{\|\mathbf{d}_k + \delta_k\|} \leq \epsilon/2
\end{aligned}$$

for sufficiently large k. We conclude

$$\left\|\mathbf{x} - \frac{\mathbf{d}_k^{\mathrm{R}}}{\|\mathbf{d}_k^{\mathrm{R}}\|}\right\| \leq \left\|\mathbf{x} - \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|}\right\| + \left\|\frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} - \frac{\mathbf{d}_k^{\mathrm{R}}}{\|\mathbf{d}_k^{\mathrm{R}}\|}\right\| \leq \epsilon.$$

□

Lemma 4.4. *Suppose* $\mathcal{K}$ *is such that* $\{t_k\}_{k\in\mathcal{K}} = \{0, 1, 2, ...\}$*. Let* $\mathcal{D}'_k$ *denote the set of normalized vectors corresponding to* $\mathcal{D}_k$*. Then* $\cup_{k\in\mathcal{K}}\mathcal{D}'_k$ *is dense on the unit sphere with probability 1.*

Proof. Because $\mathcal{U}_k$ is obtained by scaling vectors of the form $\mathbf{O}_i\mathbf{p}$ the normalized vectors from the sequence of sets $\{\mathcal{U}_k\}_{k\in\mathcal{K}}$ are uniformly distributed on the unit sphere (Lemma 3.3) and consequently the union of the normalized vectors from this sequence is dense on the unit sphere with probability 1. Due to Lemma 4.3 normalized rounded vectors from the sequence $\{\mathcal{D}_k\}_{k\in\mathcal{K}}$ are also dense on the unit sphere with probability 1 in the limit when $\Delta_k^m/\Delta_k^p \to 0$. □

Rounding affects the distribution of normalized poll directions. The following Lemma establishes the limiting distribution of normalized poll directions.

Lemma 4.5. *Let* $\mathbf{U}$ *be a random matrix distributed according to* μ_{H} *and let* $\mathbf{a}$ *be an arbitrary unit vector. Construct a random vector* $\mathbf{v} = \mathcal{R}(\Delta_k^p\mathbf{U}\mathbf{a}, \mathcal{G}_k)$*. As* Δ_k^m/Δ_k^p *approaches zero the distribution of* $\mathbf{v}$ *approaches the uniform distribution on the unit sphere.*

Proof. Rounding introduces an error denoted by δ.

$$\mathbf{v} = \Delta_k^p\mathbf{U}\mathbf{a} + \delta.$$

The normalized vector $\mathbf{v}$ can be expressed as

$$\frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{\Delta_k^p}{\|\Delta_k^p\mathbf{U}\mathbf{a} + \delta\|}\mathbf{U}\mathbf{a} + \frac{\delta}{\|\Delta_k^p\mathbf{U}\mathbf{a} + \delta\|}. \tag{4.4}$$

The error has an upper bound expressed by (4.1). By taking into account $\Delta_k^m/\Delta_k^p \to 0$ we have $\delta/\Delta_k^p \to \mathbf{0}$. By also taking into account $\|\mathbf{Ua}\| = 1$ we arrive at

$$\frac{\Delta_k^p}{\|\Delta_k^p \mathbf{Ua} + \delta\|} = \left\| \mathbf{Ua} + \frac{\delta}{\Delta_k^p} \right\|^{-1} \to 1. \tag{4.5}$$

Rewriting the second term in (4.4) and taking into account (4.5) yields

$$\frac{\delta}{\|\Delta_k^p \mathbf{Ua} + \delta\|} = \frac{\delta}{\Delta_k^p} \left\| \mathbf{Ua} + \frac{\delta}{\Delta_k^p} \right\|^{-1} \to \frac{\delta}{\Delta_k^p} \to \mathbf{0}. \tag{4.6}$$

From (4.4), (4.5), and (4.6) we conclude

$$\mathbf{v}/\|\mathbf{v}\| \to \mathbf{Ua}.$$

Vector **Ua** is uniformly distributed on the unit sphere (see Lemma 3.3). □

5 MADS with Uniformly Distributed Poll Directions

The algorithm for generating scaled poll directions (Algorithm 4.1) requires a sequence of realizations of a random matrix whose elements are independent identically distributed random variables with distribution $N(0, 1)$. Generating one such matrix involves the generation of n^2 independent samples from $N(0, 1)$. In order to make the algorithm repeatable one can use a low discrepancy sequence for generating the samples. Similarly as in [15] we used a Sobol sequence described in [20]. A Sobol sequence of dimension m uniformly covers $[0, 1)^m$. The algorithm and data from [20] are capable of generating Sobol sequences of dimension up to $m = 21201$. We skip the first two members of the sequence to avoid the all-zero and all-0.5 member. For transforming the uniformly distributed samples into normally distributed samples one can use the Box-Muller transformation [7] that takes a pair of independent random variables U_1 and U_2 with uniform distribution on $(0, 1]$ and transforms them into two independent random variables Z_1 and Z_2 distributed according to $N(0, 1)$.

$$\begin{aligned} Z_1 &= (-2\ln(U_1))^{1/2} \cos(2\pi U_2), \\ Z_2 &= (-2\ln(U_1))^{1/2} \sin(2\pi U_2). \end{aligned}$$

Generating an $n \times n$ matrix with independent identically distributed samples from $N(0, 1)$ requires a Sobol sequence of dimension $m = 2\lceil n^2/2 \rceil$. The Box-Muller transformation is then used pairwise on the components of a Sobol sequence member to produce pairs of samples from $N(0, 1)$ and construct matrix $\mathbf{N}_{t_k}$.

The initial mesh size is set to $\Delta_0^m = \lceil 1 + \gamma \rceil^{-1}$ and $\tau = 4$. When a better point is found (i.e. $f(\mathbf{x}') < f_k$ in step 4 of algorithm 2.1) the mesh size is increased ($w_k = 1$), otherwise it is decreased ($w_k = -1$). The mesh size has an upper bound $\lceil 1 + \gamma \rceil^{-1}$. When this bound is reached the mesh size parameter is no longer increased (i.e. $w_k = 0$ when a better point is found). The step size parameter is initially set to $\Delta_0^p = 1$. It is increased by a factor of two whenever a better point is found and decreased by a factor of two otherwise. Similarly to [2, 25] we assign a mesh index (l_k) to every iteration of the algorithm.

$$l_{k+1} = \begin{cases} l_k - 1 & f(\mathbf{x}') < f_k \text{ in step 4 of algorithm 2.1} \\ l_k + 1 & \text{otherwise.} \end{cases} \tag{5.1}$$

The mesh and the step size update rules can be expressed with l_k as

$$\begin{aligned} \Delta_k^m &= \frac{\min\{1, 4^{-l_k}\}}{\lceil 1+\gamma \rceil}, & (5.2) \\ \Delta_k^p &= 2^{-l_k}. & (5.3) \end{aligned}$$

In [2] t_k was used for denoting the index of the set of scaled poll directions used in k-th iteration of the algorithm. In our implementation it is initially set to $t_0 = 0$. The following rule is used for determining the value of t_k for $k > 0$.

$$t_k = \begin{cases} l_k & l_k \geq \max_{i<k} l_i \\ 1 + \max_{i<k} t_i & \text{otherwise.} \end{cases} \tag{5.4}$$

By setting $t_k = l_k$ in (5.4) whenever the mesh index is the largest so far we ensure the existence of at least one refining subsequence of points $\{\mathbf{x}_k\}_{k\in\mathcal{K}}$ for which the sequence of normalized poll directions $\{\mathcal{D}_k'\}_{k\in\mathcal{K}}$ corresponds to the complete sequence $\{\mathbf{N}_i\}_{i=0}^{\infty}$ and is thus dense on the unit sphere.

We refer to the MADS instance that uses Algorithm 4.1 along with (5.4) for generating the scaled poll directions and equations (5.1)-(5.3) for updating the mesh and the poll size as UniMADS.

Theorem 5.1. *UniMADS is a valid MADS instance with convergence properties given in [3].*

Proof. From (5.2) and (5.3) we have

$$\frac{\Delta_k^m}{\Delta_k^p} = \frac{2^{-|l_k|}}{\lceil 1+\gamma \rceil} \leq \frac{1}{\lceil 1+\gamma \rceil}. \tag{5.5}$$

In the remainder of the proof we show requirements A-D are satisfied. Unrounded scaled poll directions from the set $\mathbf{u} \in \mathcal{U}_k$ satisfy $\|\mathbf{u}\| = \Delta_k^p$. Rounding introduces an error δ. The norm of a rounded scaled poll direction $\mathbf{d} \in \mathcal{D}_k$ satisfies

$$\|\mathbf{d}\| \leq \Delta_k^p + \max \|\delta\| = \Delta_k^p + \Delta_k^m n^{1/2}/2.$$

By choosing

$$C = 1 + \frac{n^{1/2}}{2\lceil 1+\gamma \rceil} \geq 1 + \frac{2^{-|l_k|}}{\lceil 1+\gamma \rceil} \cdot \frac{n^{1/2}}{2} = 1 + \frac{\Delta_k^m}{\Delta_k^p}\frac{n^{1/2}}{2} \geq \frac{\|\mathbf{d}\|}{\Delta_k^p}$$

requirement A is satisfied. Requirement B is satisfied by construction (see (5.2) and (5.3)). From (5.5) we have

$$\frac{\Delta_k^m}{\Delta_k^p} = \frac{2^{-|l_k|}}{\lceil 1+\gamma \rceil} \leq \frac{1}{1+\gamma} = \frac{1-\epsilon}{\gamma}.$$

By choosing $\epsilon = (1+\gamma)^{-1}$ and taking into account Lemma 4.2 we see all limit points of sequence $\{\mathcal{D}_k'\}_{k=0^\infty}$ are positive spanning sets (requirement C). Finally due to Lemma 4.4 requirement D is also satisfied with probability 1. □

Due to Theorem 5.1 the convergence properties established in [3] also apply to UniMADS with probability 1. The use of a low discrepancy sequence instead of a random number generator has the advantage that "with probability 1" can be removed from Lemma 4.4 and from the proof of Theorem 5.1.

6 Numerical Results

In [25] the set of scaled poll directions is generated by forming a $n \times (n+1)$ matrix $\mathbf{A}$ where the first column is a random point chosen from the uniform distribution on the unit sphere and the remaining n columns form an orthogonal augmentation matrix $\mathbf{Q_B}$. Matrix $\mathbf{Q}_\mathrm{B}$ is obtained by applying QR decomposition to a realization of a random $n \times n$ matrix with independent random elements chosen from the uniform distribution over $[-1, 1]$. Next QR decomposition is applied to $\mathbf{A}$ resulting in an orthogonal matrix $\mathbf{Q}$ and an upper trapezoidal matrix $\mathbf{R}$. The set of $2n$ scaled poll directions in QRMADS comprises the columns of $\mathbf{Q}$ and their negatives. In [25] no proof was given for the uniform distribution of the resulting normalized poll directions. The uniformity of the distribution was demonstrated by scatter plots of 2-dimensional projections of the generated directions.

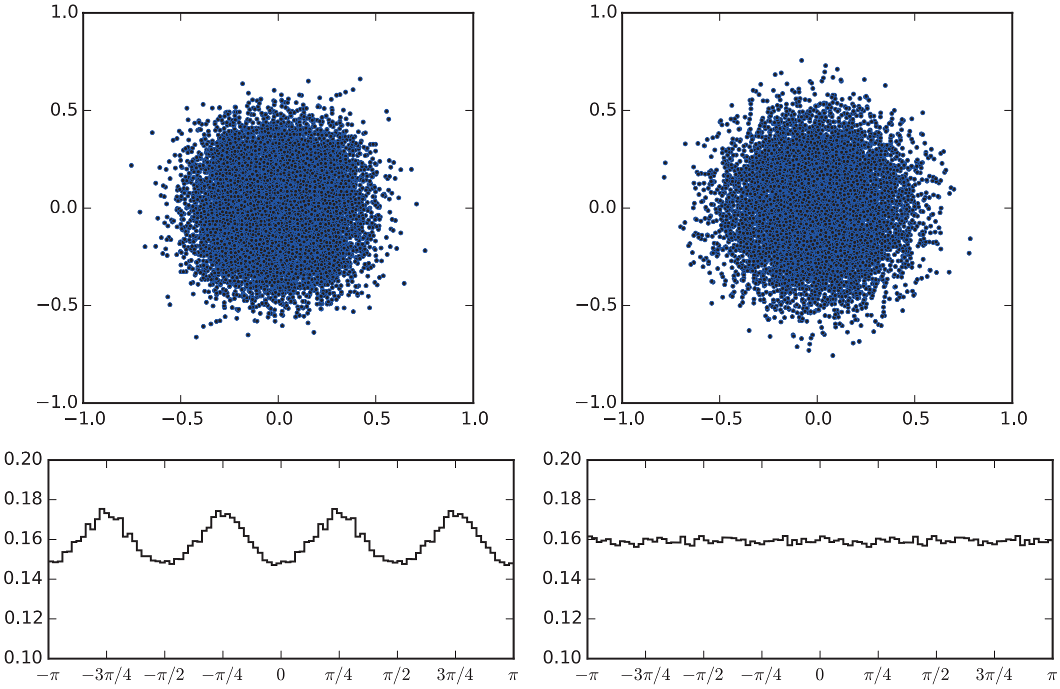

Figure 1: Scatter plot of the projected normalized poll directions (top) and the distribution of the angle between the projected normalized directions and $\mathbf{e}_1$ (bottom) for QRMADS (left) and UniMADS (right).

To compare the uniformity of the scaled poll directions we generated 10^6 scaled poll directions with n set to 20. Figure 1 depicts the two-dimensional projections of the generated directions and the distribution of the corresponding polar angle for QRMADS and UniMADS with n orthogonal vectors and their negatives in the role of the prototype set. It is quite obvious from the angular distributions that the normalized poll directions in QRMADS are not uniformly distributed (Figure 1, left, extrema at $k\pi/4$, $k \in \mathbb{Z}$). The distribution obtained with UniMADS is uniform (Figure 1, right). Both scatter plots in Figure 1 look similar suggesting that plots of the projections of generated directions are not the best tool for observing the uniformity of a distribution. Only a close look at the scatter plot in Figure

1 (a) reveals that its shape resembles a square (directions close to $\pi/4 + k\pi/2$, $k \in \mathbb{Z}$ are more common than directions close to $k\pi/2$).

The performance of UniMADS was compared to the performance of QRMADS and the state of the art MADS implementation NOMAD [5]. For the purpose of comparison four sets of test problems were used. The first three sets comprise 60 smooth problems, 62 nonsmooth problems, and 28 constrained problems from [25]. The fourth set (referred to as the Moré-Wild set) comprises a mixture of 159 smooth, nonsmooth and deterministically noisy problems [22] based on the CUTEst collection [16]. The dimension of the problems ranged from 3 to 40 for the first three sets and from 2 to 12 for the fourth set. A random orthogonal transformation was applied to all problems to eliminate any unfair advantage of algorithms that generate scaled poll directions aligned with the descent directions of test problems.

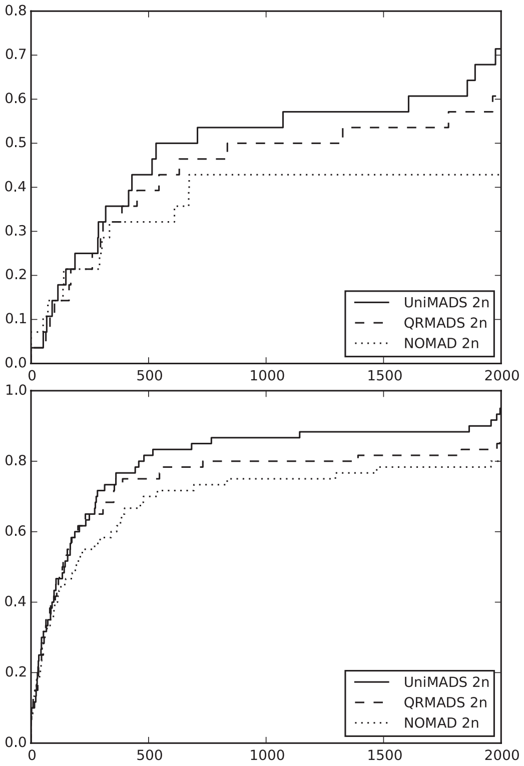

Figure 2: Data profiles for the constrained (top left), the nonsmooth (top right), the smooth (bottom left), and the Moré-Wild (bottom right) set of test problems obtained with poll direction sets comprising $2n$ members.

The constraints were handled with the extreme barrier approach which requires the initial point to be feasible (i.e. $\mathbf{x}_0 \in \Omega$). The results were expressed in terms of data profiles [22]. A data profile visualizes the share of the solved problems with respect to the computational budget expressed in terms of simplex gradient evaluations. Every simplex gradient evaluation comprises $n + 1$ function evaluations. A problem is considered as solved when

$$f_\Omega < f_L + \varepsilon(f_0 - f_L),$$

where f_L denotes the lowest value of f_Ω observed across all compared algorithms and f_0 denotes the value of f_Ω at the initial point. The comparison tolerance was set to $\varepsilon = 10^{-3}$. We limit our comparisons to a computational budget of $2000(n+1)$ function evaluations. In order to evaluate the efficiency of the poll step we omit the search step completely. For all tested algorithms the set of scaled poll directions is ordered before polling according to the angle formed with the last scaled poll direction that decreased the value of f_Ω. Polling was interrupted as soon as a point with a lower value of f_Ω was found. Optimization was stopped when the poll size parameter satisfied $\Delta_k^p < 10^{-6}$ or the computational budget was exhausted.

Our goal was to compare the performance of different poll direction generators. Therefore QRMADS used a different rounding than in [25]. A poll direction $\mathbf{u}$ was rounded to the closest grid point without first normalizing it so that $\|\mathbf{u}\|_\infty = \Delta_k^p$. This way the length of all rounded poll directions was approximately equal to Δ_k^p (like in OrthoMADS). The original QRMADS rounding scheme results in poll directions with different lengths (i.e. the length ratio between the shortest and the longest poll direction in a set can be up to $1 : n^{1/2}$).

In the first comparison we considered scaled poll direction sets with $2n$ members. We compared UniMADS with NOMAD and QRMADS. The results are depicted in Figure 2. On all but the Moré-Wild set of test problems UniMADS outperformed NOMAD. The performance of UniMADS was slightly better than the performance of QRMADS on smooth and constrained problems. This is consistent with the observed nonuniformity of directions generated by QRMADS (Figure 1). On nonsmooth problems UniMADS and QRMADS exhibited similar performance. On the Moré-Wild set of test problems the performance of all three algorithms was similar.

Table 1: The number of problems solved by UniMADS/QRMADS/NOMAD listed by problem type and problem dimension. The set of poll directions comprised $2n$ directions. The number of problems in every group is listed in parentheses.

n	Constrained	Nonsmooth	Smooth	Moré-Wild
1..5	4/3/3 (5)	9/7/8 (11)	15/14/14 (15)	50/50/53 (60)
6..10	7/5/3 (9)	12/10/6 (16)	11/8/8 (11)	62/63/62 (72)
11..20	3/3/3 (6)	9/11/9 (15)	10/11/10 (12)	22/26/22 (27)
21..30	3/3/1 (4)	9/9/5 (10)	9/9/7 (10)	-/-/- (0)
31..40	3/3/2 (4)	6/9/6 (10)	12/10/9 (12)	-/-/- (0)
All dimensions	20/17/12 (28)	45/46/34 (62)	57/52/48 (60)	134/139/137 (159)

Table 1 compares the performance of UniMADS, QRMADS, and NOMAD using $2n$ poll directions for various problem types and problem dimensions. For almost all problems and all dimensions QRMADS outperforms NOMAD. On constrained and smooth problems UniMADS exhibits similar or better performance than QRMADS. The difference between both algorithms is bigger for lower dimensional problems. This also holds for nonsmooth problems with the exception of problem dimensions $n = 11..20$ and $n = 31..40$. For the former UniMADS performs slightly worse than QRMADS, while for the latter the difference is bigger. On the Moré-Wild set of problems the performance difference between UniMADS and QRMADS is small up to $n = 10$. For $n > 10$ QRMADS performs better than UniMADS.

In our second comparison we considered algorithms for which the set of scaled poll directions comprised $n + 1$ vectors. We compared the performance of UniMADS to the

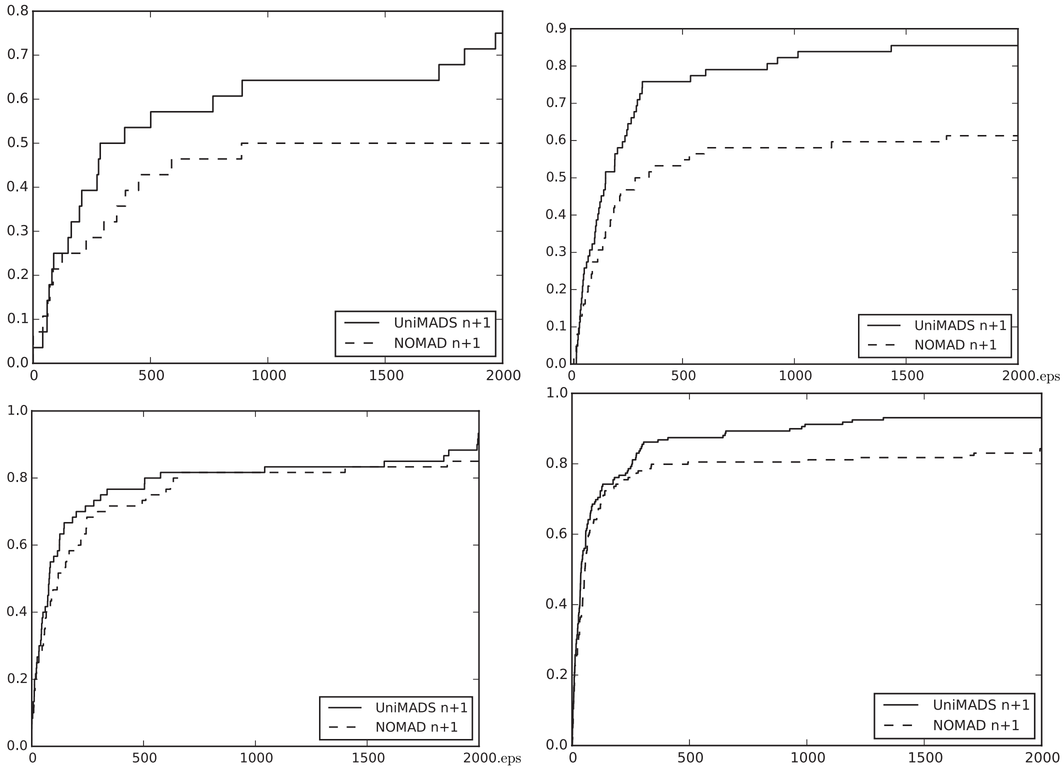

Figure 3: Data profiles for the constrained (top left), the nonsmooth (top right), the smooth (bottom left), and the Moré-Wild (bottom right) set of test problems obtained with poll direction sets comprising $n + 1$ members.

performance of NOMAD instance from [4]. This instance choses the first n scaled poll directions from the set of $2n$ poll directions used by NOMAD $2n$. The selected directions are mutually orthogonal and the angle between any of the first n directions and the last poll direction that reduced the value of f does not exceed $\pi/2$. The $(n + 1)$-th direction is obtained as the negative sum of the first n directions. The results of the comparison are depicted in Figure 3. On all sets of test problems UniMADS outperformed NOMAD, most notably on nonsmooth problems.

We also tested the performance of UniMADS on two nominal simulation-based integrated circuit sizing problems. Nominal circuit sizing is the initial step of automated circuit sizing [17]. The goal of circuit sizing is to find the values of circuit's design parameters for which the circuit's performance satisfies the design requirements. Two circuits were sized. The first one was a Miller transconductance amplifier (MTA) in Figure 4 and the second one was a folded cascode transconductance amplifier (FCTA) in Figure 5 [6].

The function subject to optimization reflects how much the circuit corresponding to design parameters given by $\mathbf{x}$ violates the m design requirements. A systematic approach for constructing f is given in [9]. It can be formulated as

$$f(\mathbf{x}) = \sum_{i=1}^{m} p_i(g_i, y_i(\mathbf{x})),$$

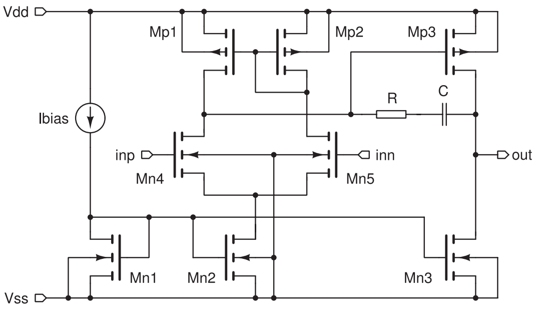

Figure 4: Schematic of the Miller transconductance amplifier (MTA).

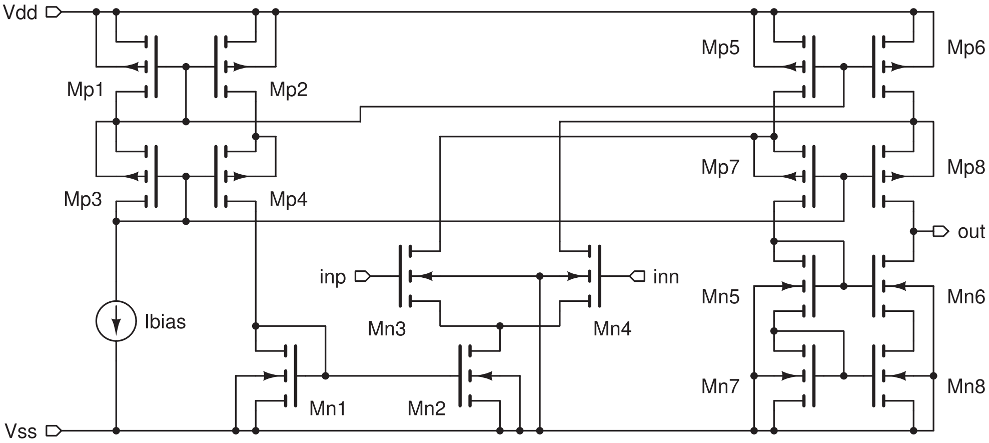

Figure 5: Schematic of the folded cascode transconductance amplifier (FCTA).

Table 2: Design requirements for MTA and FCTA.

Property	Type	MTA g_i	FCTA g_i
V_{GS} overdrive [mV]	$>$	1	1
V_{DS} overdrive [mV]	$>$	1	1
Swing [V]	$>$	1.5	0.5
Gain [dB]	$>$	65	70
Unity gain bandwidth [MHz]	$>$	40	7
Phase margin [°]	$>$	90	60
Overshoot [%]	$<$	0.1	0.1
Undershoot [%]	$<$	0.1	0.1
Rising slew rate [V/μs]	$>$	20	1
Falling slew rate [V/μs]	$>$	20	1
Area [μm^2]	$<$	1000	1000

where $y_i(\mathbf{x})$ is the value of i-th property computed by a circuit simulator and g_i is the corresponding minimal requirement. Function p_i is defined depending on the type of the design requirement listed in Table 2 as

$$p_i(g_i, y_i(\mathbf{x})) = \begin{cases} \max(g_i - y_i(\mathbf{x}), 0)/n_i & \text{type}(i) \text{ is} > \\ \max(y_i(\mathbf{x}) - g_i, 0)/n_i & \text{type}(i) \text{ is} < . \end{cases}$$

The norm n_i was set to 1 for all properties, except for the circuit area where $n_i = 10^{-10}$ was used. Function f is nonnegative and nonsmooth with $f(\mathbf{x}) = 0$ corresponding to design parameters for which the circuit satisfies all design requirements. The MTA (FCTA) circuit comprises 8 (16) transistors with two design parameters per transistor (i.e. channel width and length). The first two design requirements (V_{GS} and V_{DS} overdrive) apply to the operating point of every transistor. Therefore every overdrive requirement is composed of 8 sub-requirements in the MTA case and 16 sub-requirements in the FCTA case.

Certain design parameters must by construction be identical for a group of transistors. In the MTA case such groups are (Mn1, Mn2, Mn3), (Mn4, Mn5), (Mp1, Mp2), and Mp3. Together with the resistance of resistor R and the capacitance of capacitor C MTA has 10 design parameters. In the FCTA case the groups of transistors with identical dimensions are (Mn1, Mn2), (Mn3, Mn4), (Mn5, Mn6), and (Mn7, Mn8). Transistors in groups (Mp1, Mp3), (Mp2, Mp4), (Mp5, Mp6), and (Mp7, Mp8) share the same channel width, while transistors in groups (Mp1, Mp2, Mp5, Mp6) and (Mp3, Mp4, Mp7, Mp8) share the same channel length. This results in 14 design parameters for the FCTA case. Transistor channel widths and lengths were limited to intervals $[1\mu\text{m}, 95\mu\text{m}]$ and $[0.18\mu\text{m}, 4\mu\text{m}]$, respectively. Resistances and capacitances were limited to $[1\Omega, 200\text{k}\Omega]$ and $[10\text{fF}, 10\text{pF}]$, respectively. The lower bounds of the respective parameters were chosen as the components of the initial point $\mathbf{x}_0$.

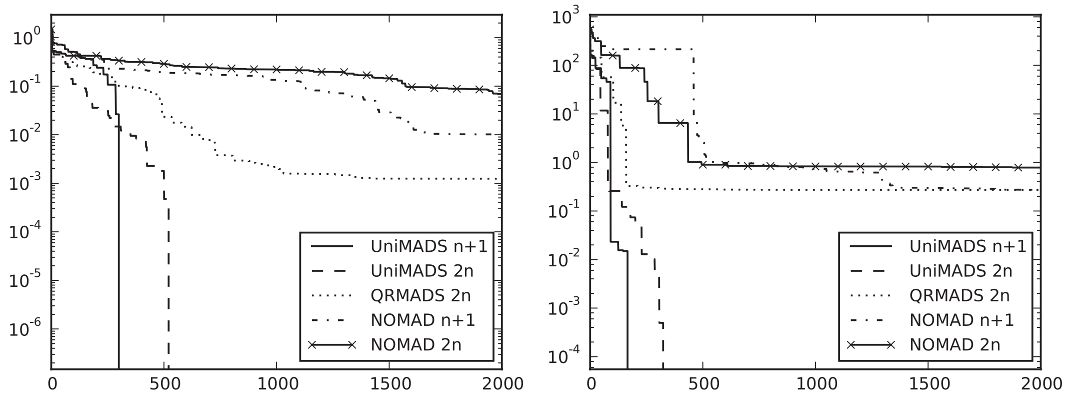

Figure 6: Progression of the objective function value versus the number of evaluations for the MTA (left) and the FCTA (right).

All simulations were performed using the HSPICE® circuit simulator [19] for temperature 25°C, supply voltage 1.8V, and bias current (Ibias current source) 100μA for the MTA and 2μA for the FCTA case. The progress of UniMADS, QRMADS, and NOMAD is depicted in Figure 6. Of both NOMAD instances the one using $n+1$ poll directions performed best. The performance of QRMADS was better than the performance of NOMAD and worse than that of UniMADS. On both cases the fastest progress was exhibited by UniMADS using $n + 1$ poll directions followed by UniMADS using $2n$ poll directions. Both algorithms found a point where $f(\mathbf{x}) = 0$. The remaining algorithms were only able to decrease the constraint violation. QRMADS was better than NOMAD $n+1$, which in turn was better than NOMAD $2n$. Most algorithms managed to reduce the value of f for at least 3 orders of magnitude (the only exceptions were the NOMAD algorithms in the MTA case and the NOMAD $2n$ algorithm in the FCTA case). This reduction is significant from a design perspective.

Table 3: Scores of the tested algorithms on 100 randomly rotated versions of the MTA and FCTA problem.

	UniMADS $n+1$	UniMADS $2n$	QRMADS	NOMAD $n+1$	NOMAD $2n$
MTA	252	266	263	122	97
FCTA	276	252	268	106	98

Finally, we optimized 100 randomly rotated versions of the MTA and the FCTA problem. For every run the algorithms were ranked using the best found value of f as the primary criterion and the number of evaluations required for finding that value as the secondary criterion. Points were assigned to algorithms depending on the rank. Algorithms were assigned scores from 4 to 0 depending on the achieved rank (the highest ranking algorithm received a score of 4). Table 3 lists the sums of scores across 100 runs. The best performing algorithm was in both cases UniMADS, albeit it was not always the version using $n + 1$ poll directions. Using a smaller number of poll directions decreases the number of function evaluations required for one iteration of the MADS algorithm. On the other hand, fewer poll directions mean that the neighborhood of the incumbent solution is explored less thoroughly. QRMADS outperformed both versions of NOMAD. The worst performing algorithm was NOMAD using $2n$ poll directions.

7 Conclusion

Instances of MADS rely on sequence of sets of normalized poll directions that is dense on the unit sphere to find descent and converge to points with favorable properties. One does not know in advance the scaled poll directions that will yield descent while at the same time result in a feasible point. Therefore a reasonable strategy is to examine the function subject to optimization in directions that are uniformly distributed on the unit sphere. In the past several MADS algorithms were proposed that gradually approached this goal: LTMADS, OrthoMADS, and QRMADS. The last one yielded most promising results although the uniformity of the distribution of the normalized poll directions was never proven mathematically. Another drawback of QRMADS is the fact that constructing poll sets with cardinality different from $2n$ is not trivial (see [15] for poll sets with $n + 1$ members).

The proposed algorithm (UniMADS) relies on a sequence of realizations of a random orthogonal matrix from the Haar measure on $\mathcal{O}_n$. A given prototype set of poll directions is transformed using the members of this sequence to produce a sequence of sets of directions that are realizations of a random unit vector with uniform distribution on the unit sphere. By scaling the members of these sets and rounding them to the grid a sequence of sets of scaled poll directions is obtained. The cardinality of the resulting sets is equal to the cardinality of the prototype set and is not restricted to $2n$.

Because MADS requires the generated points to lie on a mesh, the scaled poll directions must be rounded to a grid. The effect of rounding can deteriorate the quality of the set of poll directions, in some cases even resulting in a set that no longer positively spans the search space. To avoid such situations the poll step length must be sufficiently large compared to the mesh size. The lower bound on the ratio between the minimal poll step length and the mesh size was established for arbitrary prototype sets of poll directions. Rounding also affects the distribution of normalized poll directions. We show that in the limit this distribution approaches the uniform distribution on the unit sphere.

The proposed approach for generating the sets of scaled poll directions was used as the basis for a new instance of MADS deemed UniMADS. The distribution of the normalized poll directions in UniMADS was compared to that of QRMADS. The comparison shows the normalized poll directions generated by QRMADS are not uniformly distributed on the unit sphere. UniMADS, QRMADS, and the state of the art MADS implementation (NOMAD) were compared on four sets of test problems. Two variants of UniMADS were tested that differed in the number of directions constituting the prototype set ($2n$ and $n+1$). Both variants outperformed the corresponding MADS algorithms in NOMAD on all sets of test problems. UniMADS $2n$ exhibited slightly better performance than QRMADS. All mentioned algorithms were also tested on two nominal circuit sizing problems. Both variants of UniMADS outperformed QRMADS and the algorithms in NOMAD in terms of both speed and final solution quality. The best performing algorithm was UniMADS with a prototype set comprising $n+1$ directions. Running 100 rotated versions of both problem revealed that UniMADS $n+1$ is not always better than UniMADS $2n$. We attribute this to the fact that albeit a smaller number of poll directions results in faster search, such search is less thorough and can therefore miss a descent direction.

Acknowledgements

The authors would like to thank two anonymous reviewers for their comments which helped to improve the paper.

References

[1] M.A. Abramson and C. Audet, Convergence of mesh adaptive direct search to second-order stationary points, *SIAM J. Optim.* 17 (2006) 606–619.

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

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

[4] C. Audet, A. Ianni, S. Le Digabel and C. Tribes, Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms, *SIAM J. Optim.* 24 (2014) 621–642.

[5] S. Le Digabel, Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm, *ACM T. Math. Software* 37 (2011 1–15.

[6] R.J. Baker, *CMOS Circuit Design, Layout, and Simulation*, Wiley-IEEE Press, Hoboken (NJ), 2007.

[7] G.E.P. Box and M.E. Muller, A note on the generation of random normal deviates, *Ann. Math. Stat.* 29 (1958) 610–611.

[8] Á. Bűrmen, J. Puhan and T. Tuma. Grid restrained Nelder-Mead algorithm, *Comput. Optim. Appl.* 34 (2006) 359–375.

[9] Á Bűrmen, D. Strle, F. Bratkovič, J. Puhan, I. Fajfar and T. Tuma, Automated robust design and optimization of integrated circuits by means of penalty functions, *AEU-Int. J. Electron.* C. 57 (2003) 47–56.

[10] V.I. Bogachev, *Measure Theory*, Springer, Berlin, 2006.

[11] F. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia (PA), 1983.

[12] A.R. Conn, K. Scheinberg and L.N. Vincente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia (PA), 2009.

[13] I.D. Coope and C.J. Price, Frame-based methods for unconstrained optimization, *J. Optim. Theory Appl.* 107 (2000) 261–274.

[14] C. Davis, Theory of positive linear dependence, *Amer. J. Math.* 76 (1954) 733–746.

[15] B. Van Dyke, *Directional Direct-Search Optimization Methods with Polling Directions Based on Equal Angle Distributions*, PhD thesis, Washington State University (WA), 2012.

[16] N.I.M. Gould, D. Orban and Ph. Toint, CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Comput. Optim. Appl.* (2014), DOI 10.1007/s10589-014-9687-3. Code available at `http://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki`.

[17] H.E. Graeb, *Analog Design Centering and Sizing*, Springer, Berlin, 2007.

[18] J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numer. Math.* 2 (1960) 84–90.

[19] *HSPICE® User Guide*, Synopsys, Mountain View (CA), 2008.

[20] S. Joe and F.Y. Kuo, Constructing Sobol sequences with better two-dimensional projections, *SIAM J. Sci. Comput.* 30 (2008) 2635–2654.

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

[22] J. Moré and S. Wild, Benchmarking Derivative-Free Optimization Algorithms, *SIAM. J. Optim.* 20 (2009) 172–191.

[23] R. Rockafellar, Generalized directional derivatives and subgradients of nonconvex functions, *Can. J. Math.* 32 (1980) 257–280.

[24] G.W. Stewart, The efficient generation of random orthogonal matrices with an application to condition estimators, *SIAM. J. Numer. Anal.* 17 (1980) 403–409.

[25] B. Van Dyke and T.J. Asaki, Using QR decomposition to obtain a new instance of mesh adaptive direct search with uniformly distributed polling directions, *J. Optim. Theory Appl.* 159 (2013) 805–821.

[26] L.N. Vincente and A. Custódio, Analysis of direct searches for discontinuous functions, *Math. Program.* 133 (2012) 299–325.

Manuscript received 26 November 2014
revised 26 March 2015
accepted for publication 9 April 2015

Árpád Bűrmen
University of Ljubljana, Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail address: arpadb@fides.fe.uni-lj.si

Tadej Tuma
University of Ljubljana, Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail address: tadej.tuma@fe.uni-lj.si