

Research Article

Equal Angle Distribution of Polling Directions in Direct-Search Methods

Benjamin Van Dyke

Department of Mathematics, Walla Walla Community College, 500 Tausick Way, Walla Walla, WA 99362, USA

Correspondence should be addressed to Benjamin Van Dyke; benjamin.vandyke@wwcc.edu

Received 25 February 2014; Revised 18 June 2014; Accepted 20 June 2014; Published 20 July 2014

Academic Editor: Efrén Mezura-Montes

Copyright © 2014 Benjamin Van Dyke. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The purpose of this paper is twofold: first, to introduce deterministic strategies for directional direct-search methods, including new instances of the mesh adaptive direct-search (MADS) and the generating set search (Gss) class of algorithms, which utilize a nice distribution of POLL directions when compared to other strategies, and second, to introduce variants of each algorithm which utilize a minimal positive basis at each step. The strategies base their POLL directions on the use of the QR decomposition to obtain an orthogonal set of directions or on using the equal angular directions from a regular simplex centered at the origin with vertices on the unit sphere. Test results are presented on a set of smooth, nonsmooth, unconstrained, and constrained problems that give comparisons between the various implementations of these directional direct-search methods.

1. Introduction

In [1], Vicente and Custódio introduced a framework for directional direct-search methods (DSM) encompassing both the MADS [2] class of algorithms (using an underlying mesh and simple decrease) and the Gss [3] class of algorithms (using sufficient decrease under a forcing function ρ) for black-box optimization problems of the form

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

where the objective function $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is nonsmooth, possibly discontinuous and Ω is the set of feasible points. The algorithms under this framework are shown to have first order convergence results in the Rockafellar sense [4] for discontinuous functions. The framework is shown in Algorithm 1.

While both algorithms fall under this framework and are shown to have the same convergence results, there are tradeoffs for either choice. The MADS class of algorithms not only requires a bit more complexity in choosing polling directions, but also requires simple decrease in order to accept an iteration as successful. Under the given framework, any instance of Gss requires sufficient decrease using a forcing function that must be chosen under unknown criteria.

The sufficient decrease condition, however, can be implemented in such a way that it is mostly negligible and the method for choosing polling directions is simpler because of the absence of a mesh.

Each iteration of this DSM framework is characterized by two steps, a SEARCH and a POLL. The POLL step may be opportunistic (the POLL stops if a successful point has been found) and is governed by a POLL size parameter α_k (MADS denotes this Δ_k^{ρ}) such that $\liminf_{k \rightarrow \infty} \alpha_k = 0$ under certain assumptions. These assumptions include the requirement that at each iteration either the set of SEARCH and POLL points lie on an integer mesh (MADS) or in order for the iteration to be a success the algorithm must find a point that satisfies sufficient decrease for some forcing function $\rho : (0, \infty) \rightarrow (0, \infty)$ that is continuous, nondecreasing with $\rho(t)/t \rightarrow 0$, when $t \downarrow 0$ as shown in Algorithm 1. The use of $\bar{\rho}$ in the figure indicates the use of either a forcing function ρ (sufficient decrease) or the zero function (simple decrease).

In [2], Audet and Dennis introduced the mesh adaptive direct-search (MADS) class of algorithms for (1), where the objective function f either has no available derivative information or the derivative information is unreliable and Ω is the set of feasible points determined by black-box constraint functions. Under certain assumptions, MADS is

(0) **Initialization**
 $x_0 \in \Omega, \alpha_0 > 0, 0 < \beta_1 \leq \beta_2 < 1, \gamma \geq 1$
let \mathcal{D} be a (possibly infinite) set of positive spanning sets

(1) **Iteration k**
SEARCH
evaluate f on a finite set of points S_k (on a mesh if $\bar{\rho} = 0$)
POLL (optional if **SEARCH** is successful)
Choose $D_k \in \mathcal{D}$, order the set of **POLL** points
 $P_k = \{x_k + \alpha_k d : d \in D_k\}$, and evaluate f on P_k

(2) **Updates**
if the iteration is successful (i.e. there exists $x_s \in S_k$ or $x_p \in P_k$
such that $f(x_s) < f(x_k) - \bar{\rho}(\alpha_k)$ or $f(x_p) < f(x_k) - \bar{\rho}(\alpha_k \|d_k\|)$)
 $x_{k+1} \leftarrow x_s$ or x_p
maintain or increase the step size: $\alpha_k \in [\alpha_k, \gamma\alpha_k]$
else (the iteration fails)
 $x_{k+1} \leftarrow x_k$
decrease the step size: $\alpha_k \in [\beta_1\alpha_k, \beta_2\alpha_k]$
 $k \leftarrow k + 1$
goto (1) if stopping criteria are not met

ALGORITHM 1: The directional direct-search method of [1].

shown to converge to both first-order [2] and second-order [5] stationary points in the Clarke sense [6], as well as the previously mentioned first-order stationary points in the Rockafellar sense.

MADS was introduced as a means of extending the Gps [7] class of algorithms by allowing for an infinite number of polling directions when exploring the space of variables. The first instance of MADS called LTMADS [2] relied on a random lower triangular matrix to generate the set of **POLL** directions and relied on probabilistic arguments to obtain convergence results. This instance was later shown to have undesirably large angles between **POLL** directions at some iterations. To correct these deficiencies, a second instance of MADS called ORTHOMADS [8] was introduced that is deterministic and uses a maximal orthogonal basis at each iteration for the set of polling directions.

Each iteration of MADS is characterized by two steps, a **SEARCH** and a local **POLL**, performed on a mesh whose fineness approaches zero in the limit. At each iteration the mesh M_k is defined by

$$M_k = \{x + \Delta_k^m D z : x \in U_k, z \in \mathbb{N}^{n_D}\}, \quad (2)$$

where U_k is the set of all evaluated points at the start of iteration k , Δ_k^m is the mesh size parameter at iteration k , and $D \in \mathbb{R}^{n \times n_D}$ is a matrix with n_D directions in \mathbb{R}^n . The **POLL** step is executed if the **SEARCH** step fails to find a lower objective function value than the current best solution. The **POLL** step evaluates mesh points adjacent to the current best solution in directions that form a positive spanning set for \mathbb{R}^n . The **POLL** step can be characterized by the following definition taken from [2].

Definition 1. At iteration k , the set of **POLL** trial points for the MADS algorithm is defined to be

$$P_k = \{x_k + \Delta_k^m d : d \in D_k\} \subset M_k, \quad (3)$$

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

- (i) d can be written as a nonnegative integer combination of the directions of D ,
- (ii) the distance in ℓ_∞ -norm from the **POLL** center x_k to a **POLL** trial point is bounded above by the **POLL** size parameter Δ_k^p , and
- (iii) limits (as defined by Coope and Price [9]) of the normalized sets D_k are positive spanning sets.

In [2] it is observed that the **POLL** and mesh size parameters produced by a MADS instance satisfy

$$\liminf_{k \rightarrow +\infty} \Delta_k^p = \liminf_{k \rightarrow +\infty} \Delta_k^m = 0. \quad (4)$$

ORTHOMADS [8] uses the Halton sequence [10] and the scaled Householder transformation, $H = \|q\|^2(I_N - 2vv^T)$ with $q \in \mathbb{R}^n$ and $v = q/\|q\|$, to construct an orthogonal basis for \mathbb{R}^n at each iteration. It defines

$$\Delta_k^p = 2^{-\ell_k}, \quad \Delta_k^m = \min\{1, 4^{-\ell_k}\}, \quad (5)$$

where ℓ_k is the mesh index, with $\ell_0 = 0$ at the initial iteration. ORTHOMADS has been shown to behave well in practice, but recently a new variant of MADS, called QRMADS [11], was introduced to counter the tendency of ORTHOMADS to generate a nonuniform set of polling directions.

QRMADS augments a single direction from a dense sequence on \mathbb{S}^{n-1} with a matrix of full rank and then uses the QR decomposition to construct an orthogonal basis that is then rounded to the current mesh. It defines

$$\Delta_k^p = 2^{-\ell_k}, \quad \Delta_k^m = \min\{4^{-\ell_k - \ell_n}, 4^{-\ell_n}\}, \quad (6)$$

where ℓ_n satisfies

$$\sqrt{n^2 - n} < 2^{2\ell_n + 1} \quad (7)$$

and is chosen so that after rounding the basis does not become degenerate.

Although QRMADS was shown to have a more uniform set of polling directions than ORTHOMADS, the version tested in [11] is not deterministic. This is due to the fact that the augmented matrix with full rank was generated at random. Consequently, one outcome of this paper is to introduce a deterministic version of QRMADS by deterministically generating a matrix with full rank.

It has been observed, in some cases, that reducing the number of function evaluations at every iteration from a maximal positive spanning set to a minimal positive spanning set can improve the performance of these types of algorithms (see, e.g., [12–14]). For this reason, this paper introduces instances of MADS and Gss, called EADMADS and EADGss, that have versions using both a maximal positive basis ($2n$ polling directions) and a minimal positive basis ($n+1$ polling directions). This is accomplished by generating either an orthogonal basis or a regular simplex at each iteration. In the case of the maximal variant, the algorithmic framework for EADMADS is exactly the same as QRMADS and the new name supersedes the old. The new name was chosen to represent the use of a set of directions, for each POLL step, that have an “equal angle distribution.” By this it is meant that within the positive basis, the angle between any two directions (except a direction and its opposite, when applicable) is constant.

Section 2 gives the methods for constructing maximal and minimal polling sets by finding either an orthogonal set of directions or the vertices of a regular simplex centered at the origin on the unit sphere. Section 3 gives the method used by EADMADS for constructing a polling set on the current mesh consisting of “nearly orthogonal” directions or a “nearly regular” simplex. Section 4 outlines the EADMADS algorithm and shows that it is a valid MADS instance that shares the theoretical convergence results of this class of algorithms. Section 5 outlines the EADGss instance of Gss which is essentially the same as EADMADS except that it requires sufficient decrease instead of restricting the SEARCH and POLL points to a mesh. Finally, in Section 6, a method for making both EADMADS and EADGss deterministic is detailed and numerical results comparing each against ORTHOMADS, QRMADS, and one another on a set of test problems taken from the optimization literature are given.

2. Constructing a Basis

At each iteration of the MADS algorithm (and DSM framework) a positive spanning set D_k is required. In this section, we discuss a method for taking a single direction from a sequence that is dense on the unit sphere and forming either an orthogonal basis or the set of vertices of a regular simplex on the unit sphere centered at the origin that includes the given direction. We can then take the negatives of all the directions from the orthogonal basis to form a maximal positive spanning set or take the vertices of the regular

simplex to form a minimal positive spanning set. Since one of the directions from the dense set is included in each positive spanning set, we will then have $\bigcup_{k=1}^{\infty} D_k$ as dense in the unit sphere.

2.1. Orthogonal Basis. To generate an orthogonal basis from a given direction c_k from a dense set, we employ the QR decomposition of an $n \times n$ matrix A_k of full rank. This matrix A_k can be created by augmenting c_k with any matrix of full rank (e.g., the identity or a rotation matrix) and taking the first n columns that form a basis. The QR decomposition then generates unique orthogonal matrix Q_k and upper triangular matrix R_k such that $Q_k R_k = A_k$. As this method will generate the same basis as the Gram-Schmidt process, the first direction will correspond to c_k .

2.2. Regular Simplex. To generate a regular simplex, V_k , centered at the origin with vertices on the unit sphere from a given direction c_k from a dense set, we again start from an $n \times n$ matrix A_k of full rank, created in the same manner as above, and use the following algorithm.

Let $A_k \in \mathbb{R}^{n \times n}$, with $a_i \in \mathbb{S}^{n-1}$, $i = 1, \dots, n$, where a_i is the i th column of A_k and $\{a_i\}_{i=1}^n$ is linearly independent. We construct a regular simplex, V_k , with vertices v_1, \dots, v_{n+1} on the unit sphere by using the fact that $v_i \cdot v_j = -(1/n)$ for any two vertices of the simplex. This can be done as follows.

(i) Set $v_1 = a_1$.

(ii) For $j = 1, \dots, n-1$ set

$$\lambda = \left((n-j+1)^2 - \frac{j}{n} (n-j+1) \right)^{-1/2}, \quad (8)$$

$$v_{j+1} = \frac{a_{j+1} - \lambda \sum_{i=1}^j v_i}{(n-j+1)\lambda}.$$

(iii) Set

$$v_{n+1} = -\sum_{i=1}^n v_i. \quad (9)$$

Proposition 2. v_{j+1} is the unique solution to the following equation:

$$v_{j+1} \in \arg \max_{w \in \mathbb{R}^n} w \cdot a_{j+1} \quad (10)$$

$$\text{subject to } \|w\|^2 = 1, \quad w \cdot v_i = -\frac{1}{n}, \quad i = 1, \dots, j.$$

Due to rounding issues, this algorithm has the same unfortunate computational problems as the Gram-Schmidt process and may give an answer that does not closely approximate a regular simplex, possibly even containing complex directions. To ensure that a valid regular simplex is generated, an orthogonal matrix can be used in place of an arbitrary matrix of full rank. This can be accomplished by finding the QR decomposition of A_k and then applying the algorithm to the orthogonal matrix Q_k . In the following proof, we will simply assume that A_k is already orthogonal.

Proof. To find the solution to (10) for each j , we employ the method of Lagrange multipliers. First we define the Lagrangian function as follows:

$$\begin{aligned}\mathcal{L}(w, \lambda) &= f(w) - \sum_{i=1}^{j+1} \lambda_i c_i(w) \\ &= w \cdot a_{j+1} - \sum_{i=1}^j \lambda_i \left(w \cdot v_i + \frac{1}{n} \right) - \lambda_{j+1} (\|w\|^2 - 1).\end{aligned}\quad (11)$$

We then solve the system of the following equations:

$$\nabla_w \mathcal{L}(w, \lambda) = a_{j+1} - \sum_{i=1}^j \lambda_i v_i - 2\lambda_{j+1} w = 0, \quad (12)$$

$$\|w\|^2 - 1 = 0, \quad (13)$$

$$w \cdot v_k + \frac{1}{n} = 0. \quad (14)$$

First, we multiply (12) by v_k and use (14) to obtain

$$a_{j+1} \cdot v_k - \left(\sum_{i=1}^j \lambda_i v_i \right) \cdot v_k - \frac{2}{n} \lambda_{j+1} = 0. \quad (15)$$

We can then solve for λ_{j+1} to obtain

$$\lambda_{j+1} = -\frac{n}{2} \left(a_{j+1} \cdot v_k - \left(\sum_{i=1}^j \lambda_i v_i \right) \cdot v_k \right), \quad k = 1, \dots, j. \quad (16)$$

From (12) and by induction, we can see that for each i , $v_i \in \text{span}\{a_1, \dots, a_i\}$. Consequently, $a_{j+1} \cdot v_i = 0$, $i = 1, \dots, j$. We can then reduce (16) to

$$\lambda_{j+1} = \frac{n}{2} \left(\sum_{i=1}^j \lambda_i v_i \right) \cdot v_k, \quad k = 1, \dots, j. \quad (17)$$

Using $v_k \cdot v_k = 1$, $v_i \cdot v_k = -1/n$, when $i \neq k$, and (17), we solve $(n/2)(\sum_{i=1}^j \lambda_i v_i) \cdot v_1 = (n/2)(\sum_{i=1}^j \lambda_i v_i) \cdot v_k$ to obtain

$$\lambda_k = \lambda_1, \quad k = 2, \dots, j. \quad (18)$$

Now we can substitute (18) into (17) and simplify to obtain

$$\lambda_{j+1} = \left(\frac{n-j+1}{2} \right) \lambda_1. \quad (19)$$

Next, we substitute (18) and (19) into (12) to obtain

$$a_{j+1} - \sum_{i=1}^j \lambda_1 v_i = (n-j+1) \lambda_1 w. \quad (20)$$

Then we take the norm of each side and use $v_k \cdot v_k = 1$, $v_i \cdot v_k = -1/n$, when $i \neq k$, (13), and the fact that $\|a_{j+1}\| = 1$ to obtain the quadratic equation

$$\left[(n-j+1)^2 - \frac{j(n-j+1)}{n} \right] \lambda_1^2 - 1 = 0. \quad (21)$$

We then get $\lambda_1 = \pm 1/\sqrt{(n-j+1)^2 - j(n-j+1)/n}$.

Finally, we observe

$$\nabla_w^2 \mathcal{L} = \begin{bmatrix} -2\lambda_{j+1} & 0 & \dots & 0 \\ 0 & -2\lambda_{j+1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -2\lambda_{j+1} \end{bmatrix}. \quad (22)$$

Thus the maximum occurs when this is negative definite or when $\lambda_{j+1} > 0$. We can then take $\lambda_{j+1} = (n-j+1)\lambda_1/2$ and the corresponding value of $\lambda_1 = 1/\sqrt{(n-j+1)^2 - j(n-j+1)/n}$. We then substitute λ_1, λ_{j+1} , and (18) into (12) and solve for w to obtain v_{j+1} . \square

Note, the above computation shows that when orthogonal directions are used, problem (10) has a unique solution. This is reflected by the fact that the above quadratic equation has two real solutions, one corresponding to the closest point to a_{j+1} within the constrained region and the other corresponding to the farthest point. In order to ensure that the process for generating a regular simplex is deterministic, it is important to note that this unique closest point exists and is found. We can also observe that each new vertex v_{j+1} of the regular simplex is linearly independent to the set $\{v_i\}_{i=1}^j$, showing that the first n vertices of the simplex form a basis for \mathbb{R}^n .

3. Rounding a Basis to the Mesh

Since each iteration of the MADS class of algorithms requires a positive spanning set to lie on a mesh, it is necessary to take any basis from Section 2 and obtain a close approximation to that basis lying on the given mesh. To obtain a ‘‘nearly orthogonal’’ integer basis or a ‘‘nearly regular’’ simplex with integer vertices we take each column vector q_i of Q_k or the first n columns v_i of V_k , then project onto the hypercube centered at the origin with integer side length $2 \cdot 2^{|\ell_k|+2\ell_n}$, and then round each component to the nearest integer as follows:

$$d_i = \text{round} \left(2^{|\ell_k|+2\ell_n} \frac{q_i}{\|q_i\|_\infty} \right), \quad (23)$$

where $\text{round}(\cdot)$ refers to the usual rounding operation applied to each component of the vector. We then generate the new basis H_k with column vectors d_i . When H_k is formed from Q_k we define $D_k = [H_k \ -H_k]$ to obtain a maximal positive spanning set. When H_k is formed from V_k we set $D_k = [H_k \ -\sum_{i=1}^n d_i]$ to get a minimal positive spanning set.

Note, for small values of $2^{|\ell_k|}$, it is possible that the new set of vectors $\{d_i : 1 \leq i \leq n\}$ will not form a basis. Consequently, we add the constant $2\ell_n$ to the exponent in order to ensure that H_k will be nonsingular. In [11], it was shown that when H_k is formed from Q_k and ℓ_n satisfies (7), the resulting set of directions will form a basis.

To find the constant ℓ_n when H_k is obtained from V_k , we use the fact that the distance from a matrix A to the nearest singular matrix B (in Frobenius norm) is bounded below by the smallest singular value of A ; that is, if B is singular, then $\|B - A\| \geq \sigma_n(A)$. Thus we need the singular values of the

matrix comprised of the first n columns of V_k . Proposition 3 gives this information. (Note: in [13] the matrices V and $V^T V$ shown below are studied in a similar context.)

Proposition 3. *Let $V \in \mathbb{R}^{n \times n}$ be a matrix with columns corresponding to any n vertices of a regular simplex centered at the origin with vertices on \mathbb{S}^{n-1} . Then the singular values of V are $\sigma_1 = \sqrt{(n+1)/n}$ and $\sigma_2 = \sqrt{(n+1)/n}, \dots, \sigma_n = 1/\sqrt{n}$.*

Proof. We need to find the eigenvalues of $V^T V$. We use the fact that the vertices, v_1, \dots, v_{n+1} , of a regular simplex centered at the origin on \mathbb{S}^{n-1} have the property

$$v_i \cdot v_j = \begin{cases} -\left(\frac{1}{n}\right) & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad (24)$$

to obtain

$$V^T V = \begin{bmatrix} 1 & -\frac{1}{n} & \dots & -\frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & 1 & \dots & -\frac{1}{n} & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1 & -\frac{1}{n} \\ -\frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} & 1 \end{bmatrix}. \quad (25)$$

Then observe that

$$\begin{aligned} & \begin{bmatrix} 1 & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & 1 & \dots & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 - \frac{n-1}{n} \\ 1 - \frac{n-1}{n} \\ \vdots \\ 1 - \frac{n-1}{n} \end{bmatrix} = \begin{bmatrix} \frac{1}{n} \\ \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \end{aligned} \quad (26)$$

showing that $V^T V$ has eigenvalue $1/n$. Next, observe that

$$\begin{aligned} & \begin{bmatrix} 1 & -\frac{1}{n} & \dots & -\frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & 1 & \dots & -\frac{1}{n} & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1 & -\frac{1}{n} \\ -\frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 + \frac{1}{n} \\ -1 - \frac{1}{n} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{n+1}{n} \\ -\frac{n-1}{n} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{n+1}{n} \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \end{aligned} \quad (27)$$

showing that $(n+1)/n$ is an eigenvalue with eigenvector $[1 \ -1 \ 0 \ \dots \ 0]^T$.

Similarly, $[1 \ 0 \ -1 \ 0 \ \dots \ 0]^T, \dots, [1 \ 0 \ 0 \ \dots \ 0 \ -1]^T$ are also eigenvectors with eigenvalue $(n+1)/n$. Since these vectors are linearly independent, they form a basis for the corresponding eigenspace, thus accounting for all n eigenvalues. We finish by taking the square roots to obtain the singular values of V . \square

Next, we need the fact that for any two $n \times n$ matrices B and C , $\sigma_n(BC) \geq \sigma_n(B)\sigma_n(C)$, where σ_n denotes the smallest singular value (follows from Theorem 3.1 of [21]). Then letting A be the matrix with columns $2^{|\ell_k|+2\ell_n} v_i / \|v_i\|_\infty$, that is,

$$\begin{aligned} A = VT &= 2^{|\ell_k|+2\ell_n} \begin{bmatrix} \frac{v_{1_1}}{\|v_1\|_\infty} & \frac{v_{2_1}}{\|v_2\|_\infty} & \dots & \frac{v_{n_1}}{\|v_n\|_\infty} \\ \frac{v_{1_2}}{\|v_1\|_\infty} & \frac{v_{2_2}}{\|v_2\|_\infty} & \dots & \frac{v_{n_2}}{\|v_n\|_\infty} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{v_{1_n}}{\|v_1\|_\infty} & \frac{v_{2_n}}{\|v_2\|_\infty} & \dots & \frac{v_{n_n}}{\|v_n\|_\infty} \end{bmatrix} \\ &= \begin{bmatrix} v_{1_1} & v_{2_1} & \dots & v_{n_1} \\ v_{1_2} & v_{2_2} & \dots & v_{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1_n} & v_{2_n} & \dots & v_{n_n} \end{bmatrix} \begin{bmatrix} \frac{2^{|\ell_k|+2\ell_n}}{\|v_1\|_\infty} & 0 & \dots & 0 \\ 0 & \frac{2^{|\ell_k|+2\ell_n}}{\|v_2\|_\infty} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{2^{|\ell_k|+2\ell_n}}{\|v_n\|_\infty} \end{bmatrix}, \end{aligned} \quad (28)$$

we obtain $\sigma_n(A) = \sigma_n(VT) \geq \sigma_n(V)\sigma_n(T) \geq 2^{|\ell_k|+2\ell_n}(1/\sqrt{n})$. So for $A + E$ to be nonsingular, we need

$$\|A + E - A\| = \|E\| < \frac{2^{|\ell_k|+2\ell_n}}{\sqrt{n}}, \quad (29)$$

```

(0) Initialization
     $x_0 \in \Omega, \ell_0 \leftarrow 0, k \leftarrow 0, t_0 \leftarrow 0$ 
(1) Iteration  $k$ 
    SEARCH (optional)
        evaluate  $f$  on a finite subset  $S_k \subset M_k$ 
    POLL (optional if SEARCH is successful)
        if the POLL size is the smallest one so far
             $t_k \leftarrow \ell_k + t_0$ 
        else
             $t_k \leftarrow 1 + \max\{t_j : j = 0, 1, \dots, k\}$ 
        find  $c_k$ , compute  $H_k$  and  $D_k = [H_k - H_k]$  or  $D_k = [H_k - \sum_{i=1}^n d_i]$ 
        evaluate  $f$  on  $P_k \subset M_k$ 
(2) Updates
    if the iteration is successful (there exists  $x_s \in S_k$  or  $x_p \in P_k$ 
        such that  $f(x_s) < f(x_k)$  or  $f(x_p) < f(x_k)$ )
         $x_{k+1} \leftarrow x_s$  or  $x_p$ 
         $\ell_{k+1} \leftarrow \ell_k - 1$ 
    else (the iteration fails)
         $x_{k+1} \leftarrow x_k$ 
         $\ell_{k+1} \leftarrow \ell_k + 1$ 
     $k \leftarrow k + 1$ 
    goto (1) if stopping criteria are not met

```

ALGORITHM 2: The EADMADS algorithm.

where E represents the perturbation matrix resulting from rounding. Thus we require

$$\frac{\sqrt{n^2 - n}}{2} < \frac{2^{|\ell_k|+2\ell_n}}{\sqrt{n}}, \quad (30)$$

for all values of ℓ_k . Then letting $\ell_k = 0$, we obtain the following requirement for the constant ℓ_n determined only by the dimension n :

$$\sqrt{n-1} < 2^{2\ell_n+1}. \quad (31)$$

Finally, we end with the following proposition from [11] which will be useful for showing EADMADS as a valid instance of MADS and for establishing convergence results.

Proposition 4. *If $w \in \mathbb{S}^{n-1}$ is arbitrary, $\varepsilon > 0$, and $d = \text{round}(\alpha w / \|w\|_\infty)$, then for large $\alpha > 0$ we have $\|d/\|d\| - w\| < \varepsilon/2$.*

4. The EADMADS Instance of MADS

The EADMADS algorithm differs from the LTMADS and ORTHOMADS algorithms only in the construction of the POLL directions D_k and the POLL set P_k . For each of the three instances, the mesh M_k is defined by the set of directions $D = [I_n \ -I_n]$. For the EADMADS implementation with $2n$ polling directions, the choices of the mesh and POLL size parameters are

$$\Delta_k^p = 2^{-\ell_k}, \quad \Delta_k^m = \min \{4^{-\ell_k - \ell_n}, 4^{-\ell_n}\}, \quad (32)$$

with ℓ_n defined as in (7), so that $\Delta_k^p / \Delta_k^m = 2^{|\ell_k|+2\ell_n}$, for all k , as in (23). For the EADMADS implementation with $n+1$ polling

directions, the choices of the mesh and POLL size parameters are

$$\Delta_k^p = n2^{-\ell_k}, \quad \Delta_k^m = \min \{4^{-\ell_k - \ell_n}, 4^{-\ell_n}\}, \quad (33)$$

with ℓ_n defined as in (31), so that $\Delta_k^p / \Delta_k^m = n2^{|\ell_k|+2\ell_n}$, for all k . The use of the factor n reflects the possibility that $-\sum_{i=1}^n d_i$ may be up to n times as long as each d_i in ℓ_∞ -norm and is the same strategy used for the $n+1$ implementation of LTMADS [2]. Since H_k approximates a regular simplex, however, we can expect $-\sum_{i=1}^n d_i$ to remain close to the length of each d_i . The update rules for the mesh index ℓ_k are as in Algorithm 2, which describes the EADMADS algorithm.

This algorithm is very similar to the algorithm found in [8], with t_k and c_{t_k} in place of the Halton directions u_{t_k} and q_{t_k, ℓ_k} . The integer t_k and the update rules are chosen so that the algorithm will generate a subsequence of unsuccessful iterations with $\Delta_k^m \rightarrow 0$ and such that the directions used in this subsequence will correspond to the entire sequence $\{c_k\}_{k=1}^\infty$, where $\{c_k\}_{k=1}^\infty$ represents a sequence that is dense on \mathbb{S}^{n-1} .

The POLL directions D_k are determined by using any sequence $\{c_i\}_{i=1}^\infty$ that is dense on the unit sphere (or the hypercube $[-1, 1]^n$), the mesh index ℓ_k , a sequence of $n \times n$ matrices of full rank, and the index t_k . At each iteration, a direction from the dense sequence c_{t_k} is augmented with a matrix to obtain a basis and then either QR decomposition or the algorithm from Section 2.2 is used to find a basis matrix with the direction from the first column corresponding to c_{t_k} . The columns of the resulting matrix are then projected onto the hypercube centered at zero with side length $2 \cdot 2^{|\ell_k|+2\ell_n}$ and rounded. The resulting matrix H_k will then be either “nearly orthogonal” or correspond to n vertices of a “nearly

```

(0) Initialization
     $x_0 \in \Omega, \ell_0 \leftarrow 0, k \leftarrow 0, t_0 \leftarrow 1$ 
(1) Iteration  $k$ 
    SEARCH (optional)
        evaluate  $f$  on a finite set of points  $S_k$ 
    POLL (optional if SEARCH is successful)
        if the POLL size is the smallest one so far
             $t_k \leftarrow \ell_k + t_0$ 
        else
             $t_k \leftarrow 1 + \max\{t_j : j = 0, 1, \dots, k\}$ 
        find  $c_{t_k}$ ,
        compute  $Q_k$  and  $D_k = [Q_k - Q_k]$  or  $V_k$  and  $D_k = [V_k - \sum_{i=1}^n v_i]$ ,
        evaluate  $f$  on  $P_k = \{x_k + 2^{-\ell_k} d : d \in D_k\}$ 
(2) Updates
    if the iteration is successful (i.e. there exists  $x_s \in S_k$  or  $x_p = x_k + 2^{\ell_k} d \in P_k$ 
        such that  $f(x_s) < f(x_k) - \bar{\rho}(2^{-\ell_k})$  or  $f(x_p) < f(x_k) - \bar{\rho}(2^{\ell_k} \|d\|)$ .)
         $x_{k+1} \leftarrow x_s$  or  $x_p$ 
         $\ell_{k+1} \leftarrow \ell_k - 1$ 
    else (the iteration fails)
         $x_{k+1} \leftarrow x_k$ 
         $\ell_{k+1} \leftarrow \ell_k + 1$ 
     $k \leftarrow k + 1$ 
    goto (1) if stopping criteria are not met

```

ALGORITHM 3: The EADGss algorithm.

regular” simplex with the length of each column in ℓ_∞ -norm bounded by Δ_k^p / Δ_k^m . In the case of the orthogonal basis, H_k is completed to a maximal positive basis composed of the $2n$ directions given by $D_k = [H_k \ -H_k]$. In the case of the regular simplex, H_k is completed to a minimal positive basis composed of the $n + 1$ directions given by $D_k = [H_k \ -\sum_{i=1}^n d_i]$. This then forces the trial point $x_k + \Delta_k^m D_k e_i$ to be exactly $2^{-\ell_k}$ from the POLL center in ℓ_∞ -norm (except possibly in the case of the direction $-\sum_{i=1}^n d_i$).

The following proposition from [11] will be useful to show that EADMADS has the same convergence properties as in [2].

Proposition 5. *The set of normalized directions $\{d_{i,k} / \|d_{i,k}\|_{i=1}^\infty\}$, with $d_{i,k} = \text{round}(2^{|\ell_k|+2\ell_n} c_i / \|c_i\|_\infty)$ as in (23), is dense on \mathbb{S}^{n-1} .*

Proposition 6 was proven in [11] when EADMADS uses $2n$ polling directions (H_k is generated from Q_k) and so only the proof for the $n + 1$ version (H_k is generated from V_k) is included.

Proposition 6. *EADMADS is a valid MADS instance with the convergence properties from [2] (and [1]).*

Proof. We need to show that the POLL directions for EADMADS satisfy the following five properties from [2, 22].

- (i) Each D_k is a positive spanning set: this follows from the use of the constant ℓ_n to ensure that the round procedure on the first n columns of V_k results in a basis.

- (ii) Any direction $D_k e_i$ ($1 \leq i \leq n + 1$) can be written as a nonnegative integer combination of the directions of D : this is true by construction since $D = [I_n \ -I_n]$ and $D_k e_i$ is an integer vector on the hypercube centered at the origin with side length $2 \cdot 2^{|\ell_k|+2\ell_n}$.
- (iii) The distance from the POLL center x_k to a POLL trial point in ℓ_∞ -norm is bounded above by Δ_k^p : this is true by construction since we ensured $\|D_k e_i\|_\infty < n2^{|\ell_k|+2\ell_n}$ and so $\|\Delta_k^m D_k e_i\|_\infty < n2^{|\ell_k|+2\ell_n} \Delta_k^m = \Delta_k^p$.
- (iv) The set of normalized directions used over all failed iterations is dense on the unit sphere: this follows from the strategy chosen for updating D_k so that the entire sequence $\{c_i\}_{i=1}^\infty$ corresponds to the sequence of failed iterates, by Proposition 5, and (4) which ensures the existence of large $|\ell_k|$.
- (v) Limits (as defined by [9]) of convergent subsequences of the normalized sets $\bar{D}_k = \{d/\|d\| : d \in D_k\}$ are positive spanning sets: for this property we need to show $\det(\bar{H}_k) > \tau$, for all k , for some constant $\tau > 0$. The result will then follow as in [22].

First, note that the set $\mathcal{S} = \{A \in \mathbb{R}^{n \times n} : \text{columns of } A \text{ correspond to } n \text{ vertices of a regular simplex centered at the origin on the unit sphere}\}$ is bounded since $\mathcal{S} \subset \{A \in \mathbb{R}^{n \times n} : \|A\| = \sqrt{n}\} \subset B(0, \sqrt{n} + \varepsilon)$, where $\|\cdot\|$ denotes the Frobenius norm on $\mathbb{R}^{n \times n}$ and $B(0, \sqrt{n} + \varepsilon)$ is the ball of radius $\sqrt{n} + \varepsilon$ centered around the matrix of zeros. Also, the set $W_\delta = \{B : \|A - B\| \leq n\delta \text{ for any } A \in \mathcal{S}\}$ is bounded since any such B would satisfy $\|B\| \leq \sqrt{n} + n\delta$ and so $W_\delta \subset B(0, \sqrt{n} + n\delta + \varepsilon)$. Now, since the determinant function is C^1 on $\mathbb{R}^{n \times n}$, it is

Lipschitz on the bounded set W_δ . That is, for any $B, C \in W_\delta$ there exists a constant $K > 0$ such that

$$|\det(C) - \det(B)| \leq K \|C - B\|. \quad (34)$$

This then yields

$$|\det(B)| \geq |\det(C)| - K \|C - B\|. \quad (35)$$

So, in particular, if we take $C \in \mathcal{S}$ (since $\mathcal{S} \subset W_\delta$) and B such that $\|C - B\| \leq n\delta$, then by applying Proposition 3 to compute $|\det(C)|$, we get

$$|\det(B)| \geq |\det(C)| - K \|C - B\| \geq \sqrt{\frac{(n+1)^{n-1}}{n^n}} - Kn\delta. \quad (36)$$

If A_k is the matrix corresponding to the first n columns of V_k , then note that, for any k , $A_k \in \mathcal{S}$ and $\bar{H}_k \in W_{\delta_k}$, for some $\delta_k > 0$, since $\bar{H}_k = Q_k + E_k$ for some perturbation matrix E_k resulting from rounding. We can then choose δ small enough so that $\sqrt{((n+1)^{n-1}/n^n)} - Kn\delta > 0$. Then by Proposition 4, we can choose $|\ell_k|$ large enough, say $|\ell_k| > \ell_\delta$, so that $\|\bar{H}_k - A_k\| \leq n\delta$ and conclude that

$$|\det(\bar{H}_k)| \geq \sqrt{\frac{(n+1)^{n-1}}{n^n}} - Kn\delta, \quad (37)$$

for all k such that $|\ell_k| > \ell_\delta$.

Next, let H_k have columns d_i and note that by construction $\det(H_k) \neq 0$. Since H_k has integer components we know that $|\det(H_k)| \geq 1$. Also, note that $\|d_i\| \leq \sqrt{n}2^{|\ell_k|+2\ell_n}$, $1 \leq i \leq n$, since each d_i is on the hypercube with side length $2 \cdot 2^{|\ell_k|+2\ell_n}$. We then get the following estimate:

$$\begin{aligned} |\det(\bar{H}_k)| &= \left| \det \begin{bmatrix} d_{1_1} & \cdots & d_{n_1} \\ \|d_{1_1}\| & \cdots & \|d_{n_1}\| \\ \vdots & \ddots & \vdots \\ d_{1_n} & \cdots & d_{n_n} \\ \|d_{1_n}\| & \cdots & \|d_{n_n}\| \end{bmatrix} \right| \\ &= \frac{|\det(H_k)|}{\prod_{i=1}^n \|d_i\|} \geq \frac{|\det(H_k)|}{n^{n/2} 2^{n(|\ell_k|+2\ell_n)}} \\ &\geq n^{-n/2} 2^{-n(|\ell_k|+2\ell_n)}. \end{aligned} \quad (38)$$

Finally, note that $n^{-n/2} 2^{-n(|\ell_k|+2\ell_n)}$ is decreasing as $|\ell_k|$ increases. So $|\det(\bar{H}_k)| \geq n^{-n/2} 2^{-n(\ell_\delta+2\ell_n)}$, if $|\ell_k| \leq \ell_\delta$, where ℓ_δ is defined above. We then let

$$\tau = \min \left\{ \sqrt{\frac{(n+1)^{n-1}}{n^n}} - Kn\delta, n^{-n/2} 2^{-n(\ell_\delta+2\ell_n)} \right\} \quad (39)$$

and conclude that

$$\det(\bar{H}_k) > \tau, \quad \text{for all } k. \quad (40)$$

□

TABLE 1: List of test problems where S: smooth, N: nonsmooth, and C: constrained.

Function	n	type
Brown almost-linear [15]	15, 40, 55	S
Discrete boundary value [15]	10, 30, 60	S
Discretized variational problem [16]	15, 55	S
Extended Rosenbrock [15]	20, 40, 60	S
Penalty 1 [15]	30, 55	S
Trigonometric [15]	35, 45	S
Watson [15]	10, 25	S
Chained CB3 I [17]	25, 40, 50	N
Chained Cragg and Levy [16]	20, 40, 56	N
Chained crescent II [17]	26, 34, 60	N
Chained Mifflin 2 [17]	15, 30, 55	N
Generalized Broyden tridiagonal [16]	25, 35, 50	N
MXHILB [17]	20, 40, 50	N
Number of active faces [17]	20, 35, 50	N
Polak 2 [18]	10	N
Schwefel [19]	15, 30, 45	N
Sparse signomial [16]	20, 40, 60	N
Watson [18]	15, 25, 30	N
Wong 3 [18]	20	N
Brown with tridiagonal II [17]	16, 34, 56	C
Crescent with tridiagonal II [17]	30, 44, 56	C
Hypersphere problem [2]	15, 40, 60	C
MAD 8 [18]	20	C
Mifflin 2 with P20 II [17]	15, 35, 50	C
Number of active faces with P20 I [17]	20, 35, 55	C
Problem 7.29 [20]	30, 45, 55	C
Wong 3 [18]	20	C
Broyden tridiagonal [15]	20, 50	S
Discrete integral equation [15]	20, 35, 50	S
Extended Powell singular [15]	16, 32, 44	S
Generalized Brown function I [16]	24, 40	S
Penalty 2 [15]	10, 35, 50	S
Variably dimensioned [15]	24, 45, 60	S
Ackley [19]	25, 45, 60	N
Chained CB3 II [17]	10, 45, 60	N
Chained crescent I [17]	30, 44	N
Chained LQ [17]	15, 35, 50	N
Generalized Broyden banded [16]	30, 45, 55	N
Generalized Maxq [17]	30, 46, 54	N
N generalized Brown [17]	26, 40, 60	N
Osborne 2 [18]	11	N
Polak 3 [18]	11	N
Shell dual [18]	15	N
Steiner 2 [18]	12	N
Wong 2 [18]	10	N
Yang [19]	34, 56, 60	N
CB3 with MAD1 I [17]	25, 40, 50	C
HS114 [18]	9	C
LQ with MAD1 I [17]	25, 40, 50	C
Maxq with MAD1 II [17]	10, 30, 50	C
MXHILB with P20 II [17]	20, 45, 60	C
Problem 7.26 [20]	16, 24, 60	C
Wong 2 [18]	10	C

TABLE 2: Distribution of test problems used for the numerical results.

Dimension	9–12	15–16	20	24–26	30–32	34–35	40	44–46	50	54–56	60	Total
Smooth functions	3	3	3	3	3	3	3	3	3	3	3	33
Nonsmooth functions	6	5	5	6	6	5	5	6	5	5	6	60
Constrained functions	3	4	4	3	3	3	3	3	4	4	3	37
Total	12	12	12	12	12	11	11	12	12	12	12	130

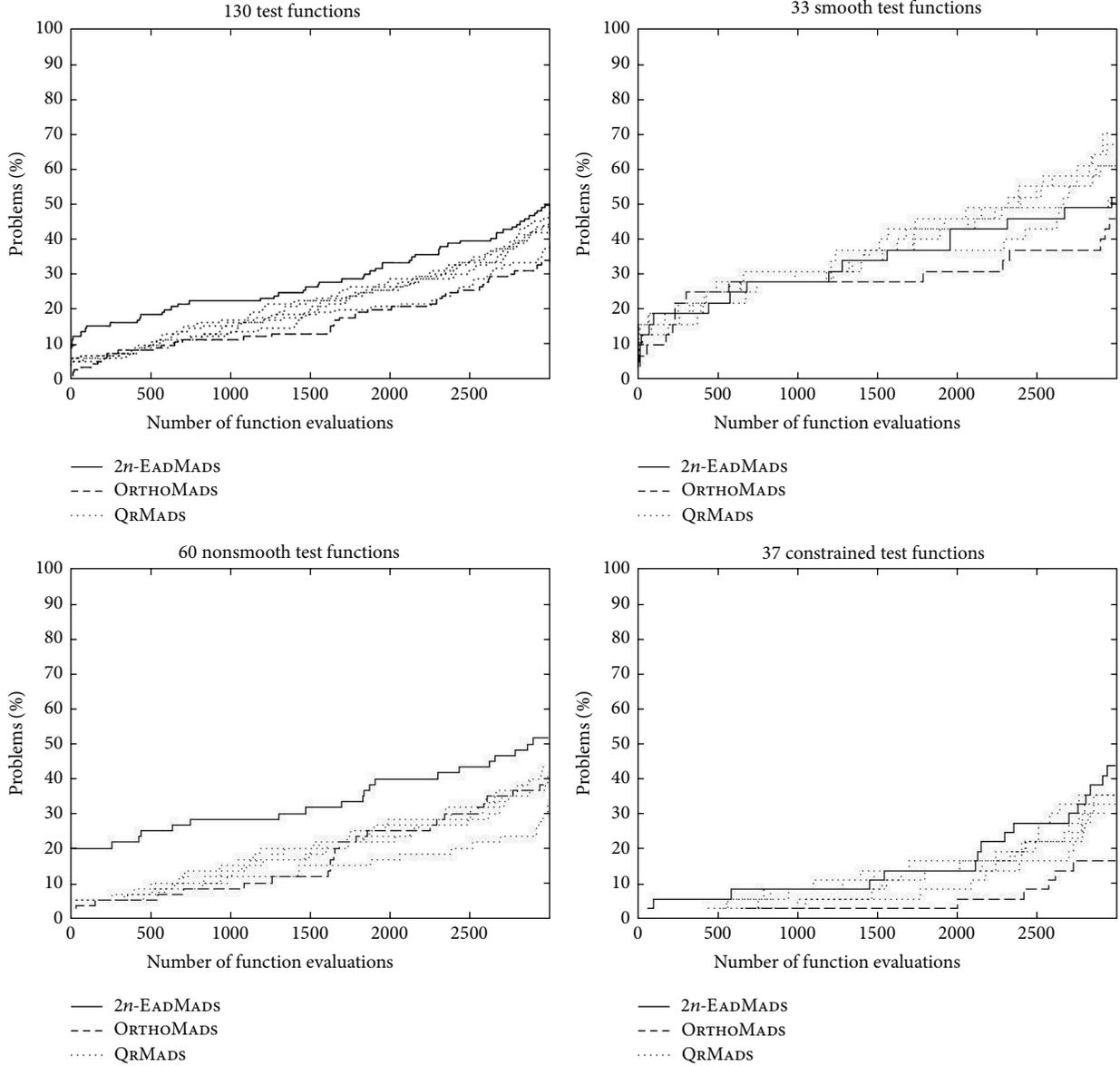


FIGURE 1: Data profiles for three instances of MADS, including five runs of QRMADS, on 130 test functions.

5. The EADGss Instance of a Directional Direct-Search Method

At each iteration, the EADGss instance of Gss uses a positive spanning set whose directions have an equal angle distribution either by using the QR decomposition to generate an orthogonal basis or by generating a regular simplex centered

at the origin. The integer t_k and the update rules are chosen so that there will be a subsequence of unsuccessful iterations with $\alpha_k \rightarrow 0$ and such that the directions used in this subsequence will correspond to the entire sequence $\{c_k\}_{k=1}^\infty$, where $\{c_k\}_{k=1}^\infty$ is dense on \mathbb{S}^{n-1} . The POLL size parameter is given by $\alpha_k = 2^{-\ell_k}$. The update rules for ℓ_k are as in Algorithm 3, which describes the EADGss algorithm.

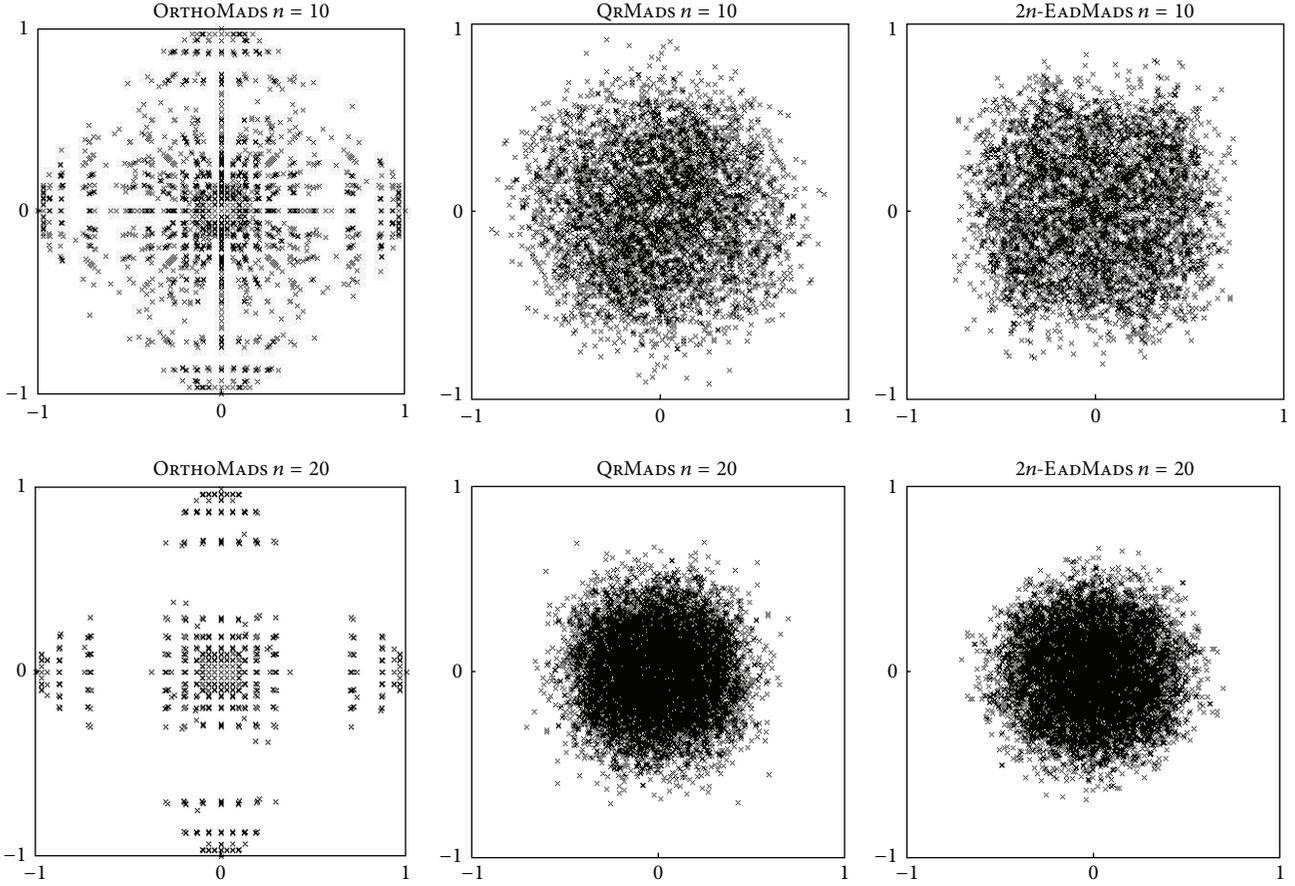


FIGURE 2: First two coordinates of the normalized POLL directions for three variants of the MADS algorithm on the n -dimensional Rosenbrock function with $n = 10$ and 20 .

The POLL directions D_k are determined by using any sequence $\{c_i\}_{i=1}^{\infty}$ that is dense on the unit sphere (or the hypercube $[-1, 1]^n$), the index ℓ_k , a sequence of $n \times n$ matrices of full rank, and the index t_k . At each iteration, a direction from the dense sequence c_{t_k} is augmented with a matrix to obtain a basis and then either QR decomposition or the algorithm from Section 2.2 (along with the QR decomposition) is used to find a basis matrix with the direction from the first column corresponding to c_{t_k} . The resulting matrix will then be either orthogonal, Q_k , or correspond to n vertices of a regular simplex, V_k , with the length of each column having ℓ^2 -norm one. In the case of the orthogonal basis, Q_k is completed to a maximal positive basis composed of $2n$ directions, $D_k = [Q_k \ -Q_k]$. In the case of the regular simplex, the first n columns of V_k are completed to a minimal positive basis composed of $n+1$ directions by finding the final vertex of the simplex, $D_k = [V_k \ -\sum_{i=1}^n v_i]$, where $\{v_i\}_{i=1}^n$ are the columns of V_k .

It should be noted that this algorithm is very similar to that for EADMADS (Algorithm 2), with the exception that EADGss uses sufficient decrease at each iteration instead of confining the SEARCH and POLL steps to a mesh. Consequently, there are only two main differences. The first is that in the final step of the construction of D_k , EADMADS rounds

and EADGss does not (hence no need for separate α_k for the $2n$ and $n+1$ variants). The second difference is in the determination of a successful iterate; EADGss uses a forcing function to determine sufficient decrease and EADMADS uses simple decrease.

6. Numerical Tests

In this section, numerical test results are given to compare various instances of MADS and Gss amongst one another. Each test was performed using a MATLAB implementation. The tests were performed on 130 problems taken from the optimization literature, with dimensions between nine and 60. The test problems were chosen to satisfy three categories. The first category consists of 33 smooth functions, the second category consists of 60 nonsmooth functions, and the third category consists of 37 constrained test problems, as listed in Table 1. Since many of the problems are variably dimensioned versions of one another, no generalization was used more than three times. The distribution of problems by dimension is given in Table 2.

The choices for Δ_k^m and Δ_k^p for ORTHOMADS correspond to (5), for the $2n$ versions of EADMADS they correspond to (6) and (7), and for the $n+1$ version of EADMADS they correspond

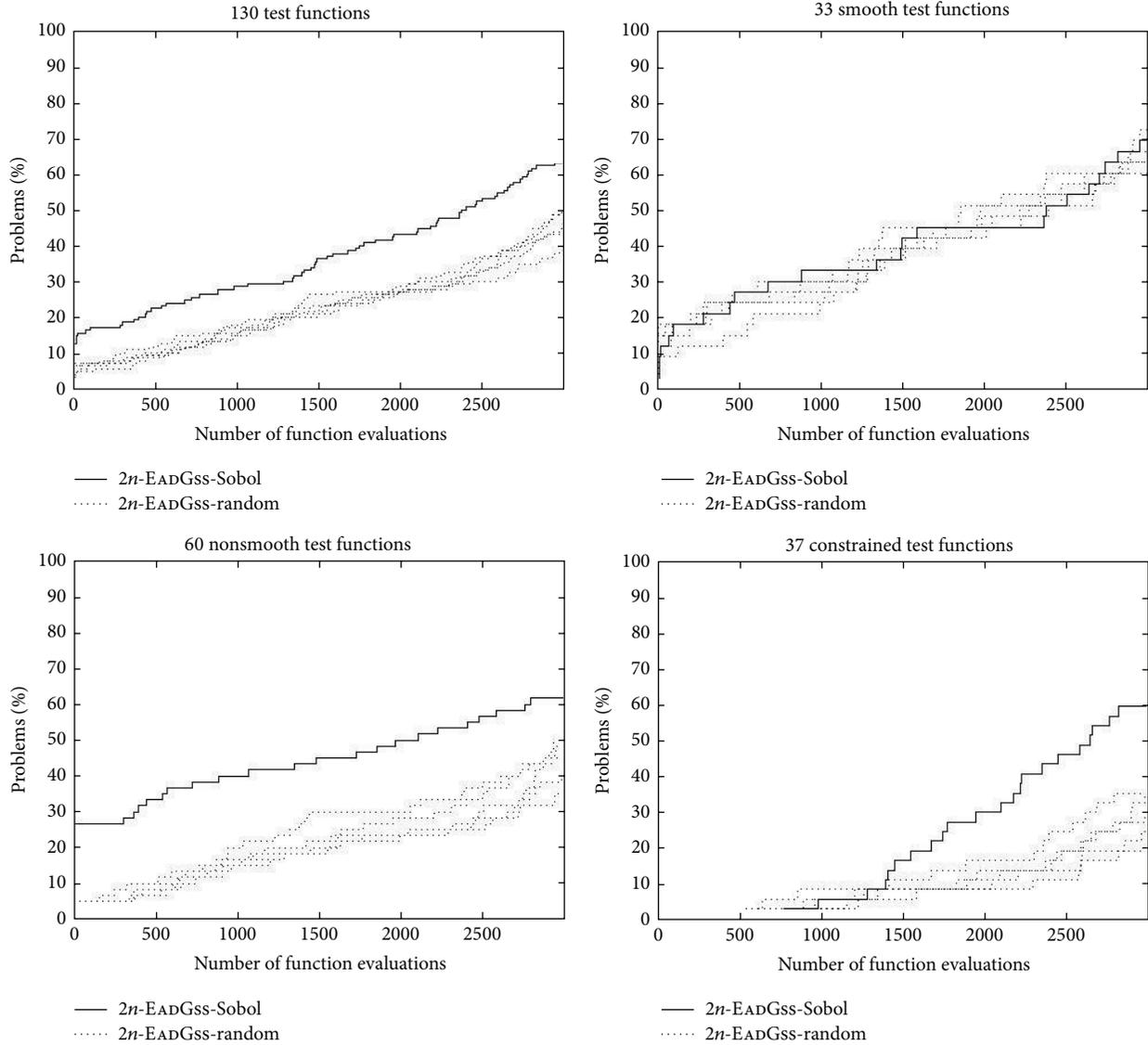


FIGURE 3: Data profiles for two variants of EADGss, including five runs of EADGss with random directions, on 130 test functions.

to (33) and (31). For each version of EADGss $\alpha_k = 2^{-\ell_k}$. The mesh index ℓ_k is allowed to be negative for all instances of both algorithms and for each instance of MADS the initial mesh size Δ_0^m is set to one.

The stopping criteria are satisfied when the number of function evaluations reach 3,000. For each implementation, a POLL step was required at every iteration and an opportunistic strategy was used (the POLL stops if a successful point has been found). For each test, no search was performed and constraints are handled using the extreme barrier method ($f(x) = +\infty$ if $x \notin \Omega$).

In order to create and test deterministic versions of both EADMADS and EADGss, the Sobol sequence [23] is used in place of either the Halton sequence or the sequence of partition centers used for QRMADS. This is done for two reasons. Although the Halton sequence is deterministic, the sequence of partition centers is not. In higher dimensions,

however, it can be observed that the Halton sequence exhibits poor uniformity. Sobol sequences, on the other hand, are both deterministic and remain uniform. For each test, the Sobol sequence was generated using the *sobolset* command in MATLAB and used in the given order.

At each iteration c_k is taken to be the t_k^{th} Sobol direction. This is then augmented with the next $2n - 1$ Sobol directions, taken in order, and the identity matrix. This is done in order to ensure that a matrix of full rank is created. From this matrix, a new matrix is formed from the first n columns that form a basis for \mathbb{R}^n and then the QR decomposition is applied to this new matrix. When a minimal positive basis is required, the resulting orthogonal matrix is then input into the simplex algorithm of Section 2.2. For EADMADS, these directions are then rounded to the current mesh. In place of the mesh, EADGss uses a sufficient decrease requirement implemented with the forcing function $\rho = t^2$.

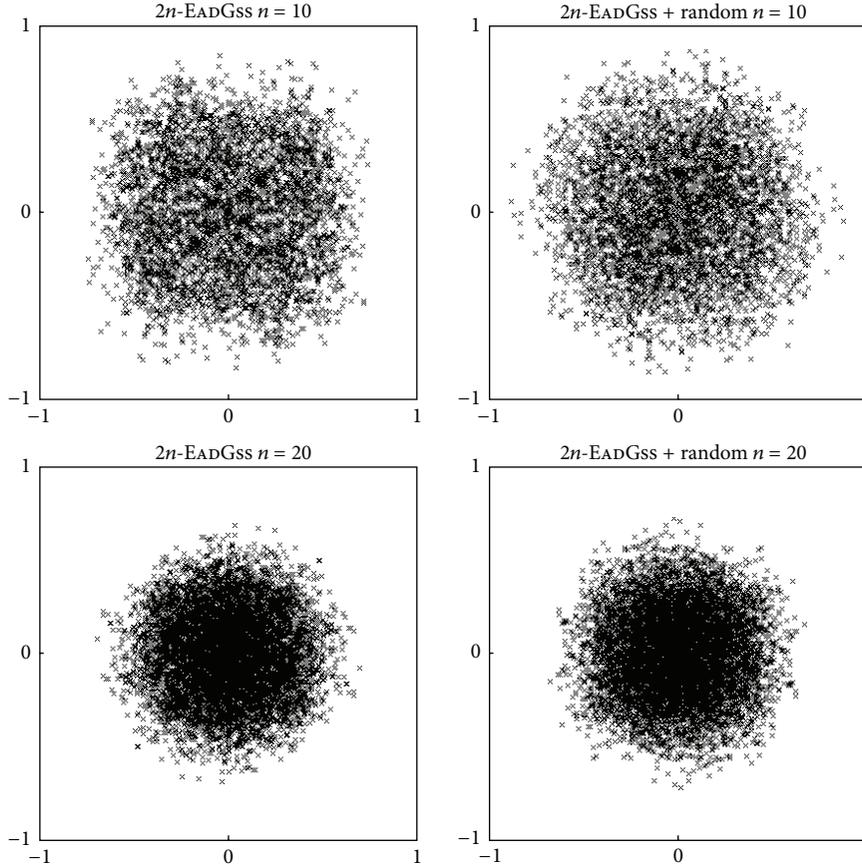


FIGURE 4: First two coordinates of the normalized POLL directions for two variants of EADGss on the n -dimensional Rosenbrock function with $n = 10$ and 20.

To compare the various instances of each algorithm, the results from the tests were used to generate data profiles, as outlined in [24]. For each test function, f_L is defined to be the smallest value found for f by any of the instances of either MADS or Gss after 3,000 function evaluations. Then after each evaluation, the algorithm is declared a success if the smallest known value for f thus far satisfies the test

$$f(x) \leq f_L + 0.001(f(x_0) - f_L). \quad (41)$$

The number of successes over all the test functions is then recorded as a percent of success. The results for each instance of MADS or Gss are then plotted as the number of function evaluations against the percent of problems to obtain the data profile.

The first test performed was to compare both the non-deterministic and deterministic instances of EADMADS with $2n$ polling directions against ORTHOMADS. The data profiles for the deterministic instance of EADMADS, five runs of the non-deterministic instance of EADMADS (formerly called QRMADS), and ORTHOMADS are given in Figure 1. A data profile is given for each instance using the set of all test functions, the set of smooth test functions only, the set of nonsmooth test functions only, and the set of constrained test functions only. In this figure, we can see that the deterministic instance of EADMADS performs very well compared to the

other two instances when considering the set of all test functions or just the set of nonsmooth functions. When just considering the set of smooth or constrained test functions, the deterministic instance of EADMADS performs roughly as well as the non-deterministic instance and better than ORTHOMADS.

Figure 2 illustrates the distribution and the density of directions for all three Mads instances. Each instance was run on the Rosenbrock function with dimensions $n = 10$ and 20. In each case, there are roughly $500n$ directions shown and each direction is projected onto the plane defined by the first two coordinates. The resulting plots are typical of what can be expected regardless of the choices of the two coordinates. In this figure, we can see that the directions for both the deterministic and non-deterministic instances of EADMADS have relatively uniform distributions with little clustering. The directions for ORTHOMADS, on the other hand tend to cluster within specific regions. While the directions for both instances of EADMADS are very similar, with the unit sphere being covered nicely, the test results from this section indicate that the deterministic instance is expected to perform as well as or better than the non-deterministic instance (QRMADS).

Note 1. In Figures 2, 4, and 6, the EADMADS and EADGss directions become more restricted towards the center as the

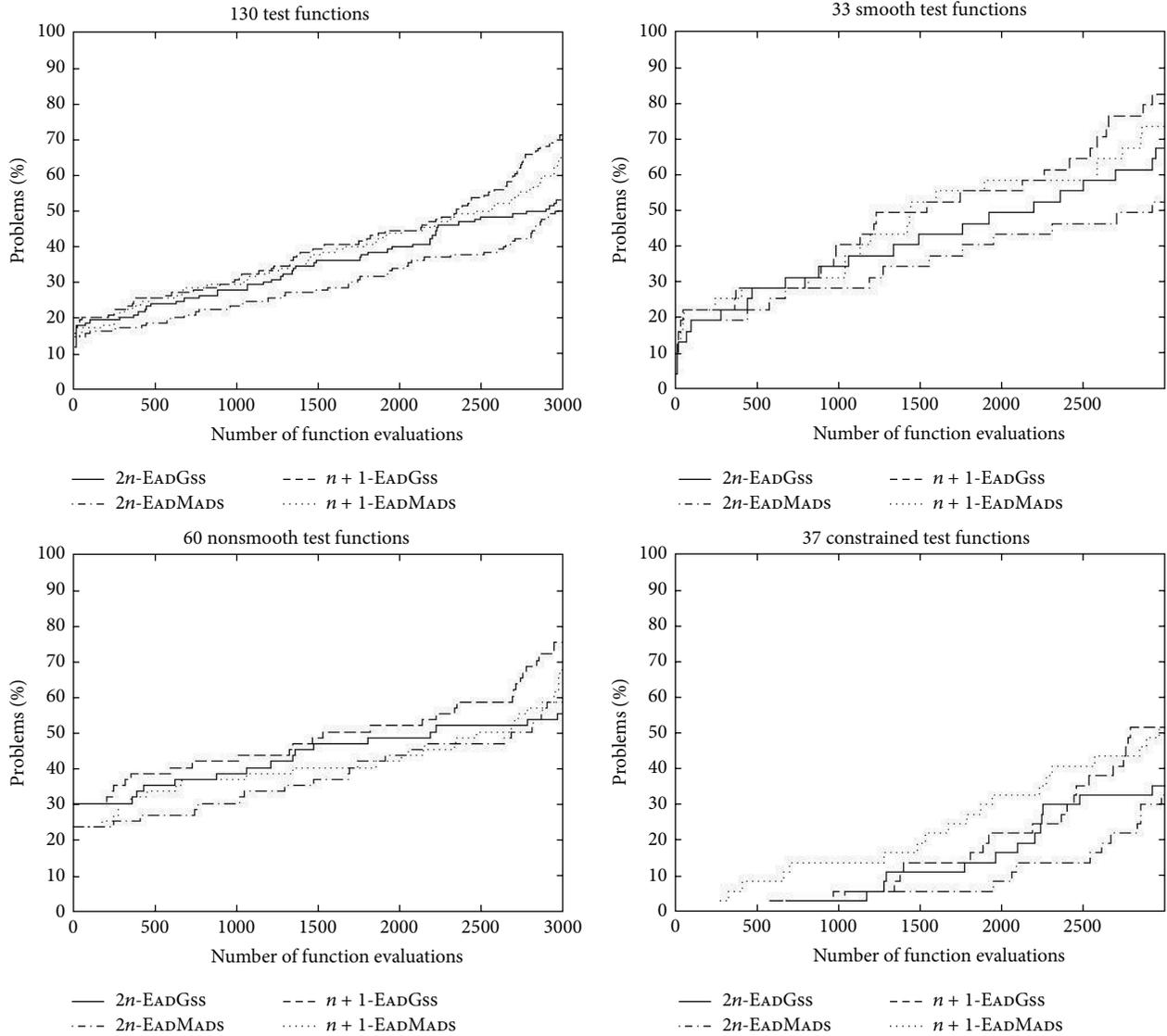


FIGURE 5: Data profiles for four instances of EADMADS on 130 test functions.

dimension increases. This behavior is a reflection of the concentration of measure phenomenon and the fact that the observable diameter of S^{n-1} converges to zero as $n \rightarrow \infty$ (for more on this topic see [25]). Intuitively, one can imagine that as n increases, the contribution to the norm of a vector from the first two elements gets smaller.

The next test performed was to compare the deterministic instance of EADGss with $2n$ polling directions to a nondeterministic instance of Gss with $2n$ polling directions using random directions in place of the Sobol sequence (this is essentially the algorithm tested in [1]). The data profiles for the deterministic instance of EADGss and five runs of the nondeterministic instance are given in Figure 3. A data profile is given for each instance using the set of all test functions, the set of smooth test functions only, the set of nonsmooth test functions only, and the set of constrained test functions only. From this figure, we can observe that the deterministic

instance performed better on the sets containing all the test functions, the nonsmooth test functions, and the constrained test functions. Both the deterministic and nondeterministic instances show similar performances on the set of smooth test functions.

Figure 4 illustrates the distribution and the density of directions for both instances of Gss. Each instance was run on the Rosenbrock function with dimensions $n = 10$ and 20 . Similar to what is observed in Figure 2, we can see that the directions from these two instances of EADGss have relatively uniform distributions with little clustering.

The last test performed was to compare the deterministic instances of EADMADS and EADGss with both $n + 1$ and $2n$ polling directions. The data profiles for each instance of the two algorithms are given in Figure 5. A data profile is given for each instance using the set of all test functions, the set of smooth test functions only, the set of nonsmooth test functions only, and the set of constrained test functions only.

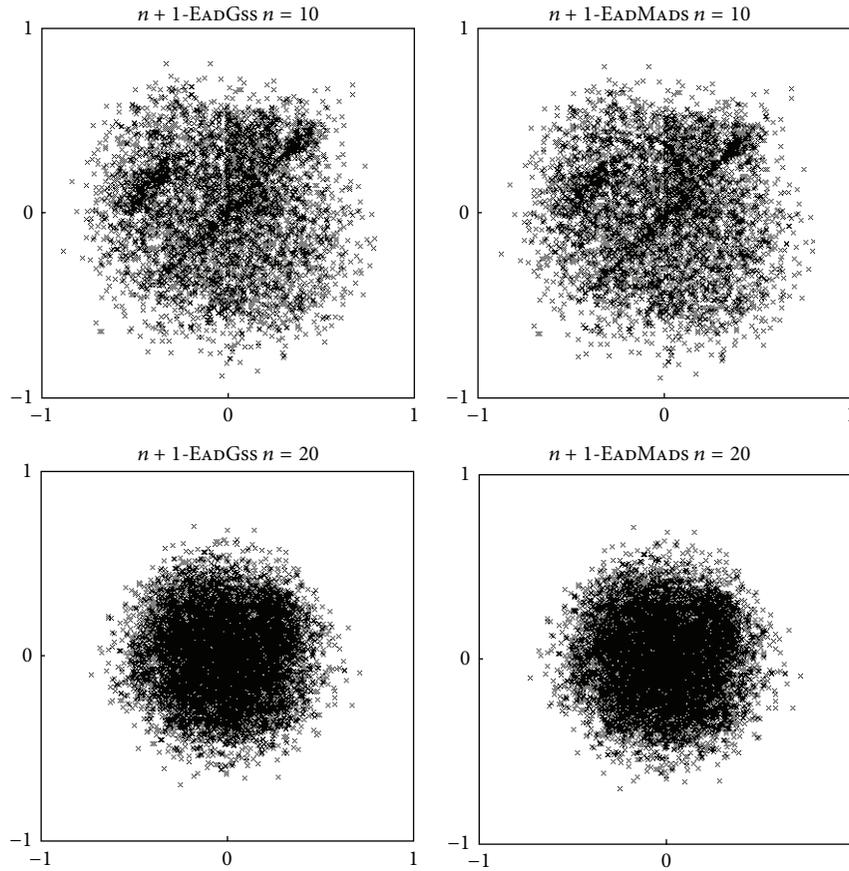


FIGURE 6: First two coordinates of the normalized POLL directions for the variants of EADGSS and EADMADS using the Sobol sequence and $n + 1$ polling directions on the n -dimensional Rosenbrock function with $n = 10$ and 20.

From this figure, we can observe that the $n + 1$ instances of both EADMADS and EADGSS perform better than the $2n$ instances on both the set of all test problems and the set of smooth test problems. We can see a lot of parity amongst the four instances on the set of nonsmooth functions with the two $n + 1$ instances doing a bit better as the number of function evaluations approaches 3,000. The results from the set of constrained test functions are a little more mixed with the $n + 1$ instance of EADMADS performing very well overall and the $n + 1$ instance of EADGSS performing well as the number of function evaluations approaches 3,000.

Figure 6 illustrates the distribution and the density of directions for the $n + 1$, deterministic instances of EADMADS and EADGSS. Each instance was run on the Rosenbrock function with dimensions $n = 10$ and 20. In this figure, we can observe that the overall distribution of directions for the $n + 1$ instances of EADMADS and EADGSS is very similar to one another and while they do exhibit some clustering, they still have relatively uniform distributions overall.

7. Conclusions

When choosing a direct-search method, often two desirable attributes have a deterministic algorithm and the ability to

use a minimal positive spanning set. This paper has outlined strategies for making both the MADS and GSS algorithms deterministic, ensuring that the end results are repeatable, and outlined a method for using a set of $n + 1$ directions generated from a regular simplex. These strategies were devised in a way that modifications are readily available without losing convergence results. For example, one could easily substitute any deterministic sequence that is dense on the unit sphere into the framework presented in this paper. Consequently, an area of future inquiry could include studying the behavior of these algorithms using a variety of different sequences.

It was not the direct intention of this paper to make a comparison of the MADS algorithm and GSS algorithm using sufficient decrease. Although these two algorithms are similar, they each have advantages and disadvantages. The advantage to the MADS class of algorithms is that it only requires simple decrease, with the disadvantage being that it has the added complexity of requiring POLL and SEARCH points to lie on a mesh at each iteration. Alternately, GSS has the advantage of eliminating the mesh requirement but is burdened by needing a sufficient decrease requirement before accepting a new best point. One can imagine that the advantages of the two algorithms may be more desirable for

different real-world applications. Consequently, an area of future study could include a more in-depth comparison of these two algorithms.

In this paper, constraints are handled using the extreme barrier method. Recently, however, there have been suggestions for better ways to handle constraints, such as the progressive barrier method [26]. Another area of future study would include analyzing the methods from this paper in conjunction with the progressive barrier method, as well as other extensions of the MADS algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] L. N. Vicente and A. L. Custódio, "Analysis of direct searches for discontinuous functions," *Mathematical Programming*, vol. 133, pp. 299–325, 2012.
- [2] C. Audet and J. E. Dennis, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006.
- [3] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: new perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, 2003.
- [4] R. T. Rockafellar, "Generalized directional derivatives and subgradients of nonconvex functions," *Canadian Journal of Mathematics*, vol. 32, no. 2, pp. 257–280, 1980.
- [5] M. A. Abramson and C. Audet, "Convergence of mesh adaptive direct search to second-order stationary points," *SIAM Journal on Optimization*, vol. 17, no. 2, pp. 606–619, 2006.
- [6] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, Pa, USA, 1983.
- [7] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 1–25, 1997.
- [8] M. A. Abramson, C. Audet, J. Dennis, and S. Le Digabel, "OrthoMADS: a deterministic MADS instance with orthogonal directions," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 948–966, 2009.
- [9] I. D. Coope and C. J. Price, "Frame based methods for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 107, no. 2, pp. 261–274, 2000.
- [10] J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numerische Mathematik*, vol. 2, no. 1, pp. 84–90, 1960.
- [11] 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," *Journal of Optimization Theory and Applications*, vol. 159, no. 3, pp. 805–821, 2013.
- [12] C. Audet, A. Ianni, S. le Digabel, and C. Tribes, "Reducing the number of function evaluations in mesh adaptive direct search algorithms," *SIAM Journal on Optimization*, vol. 24, no. 2, pp. 621–642, 2014.
- [13] P. Alberto, F. Nogueira, H. Rocha, and L. N. Vicente, "Pattern search methods for user-provided points: application to molecular geometry problems," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1216–1236, 2004.
- [14] K. R. Fowler, J. P. Reese, C. E. Kees et al., "Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems," *Advances in Water Resources*, vol. 31, no. 5, pp. 743–757, 2008.
- [15] J. J. Moré, B. S. Garbow, and K. E. Hillstom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software*, vol. 7, no. 1, pp. 17–41, 1981.
- [16] L. Luksan and J. Vlcek, "Test problems for unconstrained optimization," Tech. Rep. 897, Academy of Sciences of the Czech Republic, Institute of Computer Science, 2003.
- [17] N. Karmitsa, "Test problems for large-scale nonsmooth minimization," Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing. Technical Report B. 4/2007, University of Jyväskylä, 2007.
- [18] L. Luksan and J. Vlcek, "Test problems for nonsmooth unconstrained and linearly constrained optimization," Tech. Rep. 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, January 2000.
- [19] X.-S. Yang, "Appendix A: test problems in optimization," in *Engineering Optimization*, pp. 261–266, 2010.
- [20] L. Luksan, M. Tuma, J. Vlcek et al., "UFO 2011 interactive system for universal functional optimization," Tech. Rep. 1151, Academy of Sciences of the Czech Republic, Institute of Computer Science, December 2011.
- [21] L. Z. Lu and C. E. M. Pearce, "Some new bounds for singular values and eigenvalues of matrix products," *Annals of Operations Research*, vol. 98, pp. 141–148, 2000.
- [22] C. Audet, A. L. Custódio, and J. E. Dennis Jr., "Erratum: mesh adaptive direct search algorithms for constrained optimization," *The SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1501–1503, 2008.
- [23] I. M. Sobol, "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, 1967.
- [24] J. J. More and S. M. Wild, "Benchmarking derivative-free optimization algorithms," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 172–191, 2009.
- [25] M. Ledoux, *The Concentration of Measure Phenomenon*, American Mathematical Society, Providence, RI, USA, 2001.
- [26] C. Audet and J. E. Dennis Jr., "A progressive barrier for derivative-free nonlinear programming," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 445–472, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

