



Grid Restrained Nelder-Mead Algorithm

ÁRPÁD BÜRMEŃ
JANEZ PUHAN
TADEJ TUMA

arpadb@fides.fe.uni-lj.si
janez.puhan@fe.uni-lj.si
tadej.tuma@fe.uni-lj.si

Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, SI-1000 Ljubljana, Slovenia

Received June 28, 2004; Revised June 20, 2005; Accepted July 11, 2005

Published online: 23 March 2006

Abstract. Probably the most popular algorithm for unconstrained minimization for problems of moderate dimension is the Nelder-Mead algorithm published in 1965. Despite its age only limited convergence results exist. Several counterexamples can be found in the literature for which the algorithm performs badly or even fails. A convergent variant derived from the original Nelder-Mead algorithm is presented. The proposed algorithm's convergence is based on the principle of grid restraint and therefore does not require sufficient descent as the recent convergent variant proposed by Price, Coope, and Byatt. Convergence properties of the proposed grid-restrained algorithm are analysed. Results of numerical testing are also included and compared to the results of the algorithm proposed by Price et al. The results clearly demonstrate that the proposed grid-restrained algorithm is an efficient direct search method.

Keywords: unconstrained minimization, Nelder-Mead algorithm, direct search, simplex, grid

1. Introduction

We consider the local unconstrained optimization problem starting with initial point $x_0 \in \mathbb{R}^n$ where $x_l \in \mathbb{R}^n$ is sought, subject to

$$\begin{aligned} f : \mathbb{R}^n &\rightarrow \mathbb{R} \\ f(x_l) &\leq f(x_0) \\ \exists \delta > 0 : f(x_l) &\leq f(x_l + a) \quad \forall a \in \mathbb{R}^n, \|a\| \leq \delta \end{aligned} \quad (1)$$

$f(x)$ is often referred to as the cost function (CF). If $f(x)$ is continuous and continuously differentiable (C^1) all points (x_l) that satisfy (1) are stationary points of $f(x)$.

To solve problems of the form (1) many different methods have been devised. Lately there has been increased activity in the area of direct search optimization methods [8]. The main feature of direct search methods is that they don't use the gradient of the cost function in the process of search. Probably the most popular direct search method is the Nelder-Mead (NM) algorithm [11] devised in the 1960s. The NM algorithm tries to find

a stationary point of the CF by moving and reshaping a polytope with $n + 1$ vertices, also referred to as the simplex. Despite its popularity the theoretical results regarding its convergence properties are very limited [6]. Furthermore several examples for which the NM algorithm either converges very slowly or even fails to converge to a stationary point of the CF were provided by [14, 17], and [9].

Shortly after McKinnon's results were published Kelley [5] proposed oriented restart as means for improving convergence properties of the original NM algorithm. Nevertheless oriented restart does not guarantee convergence. Tseng [16] presented a more general, sufficient descent based approach (fortified descent simplicial search method), capable of guaranteeing convergence to a local minimizer. His paper includes an extensive overview of papers from English, Russian, and Chinese literature dealing with simplex-based algorithms. Recently a slightly different method, focusing more on simplex reshape than preventing simplex degeneration, was presented by Price et al. [12] and is based on Byatt's MSc thesis [1] (referred to also as sufficient descent Nelder-Mead algorithm or SDNMA for short). Their approach is derived from the class of frame-based methods established by Coope and Price [2]. All of these approaches for ensuring convergence require the consecutive simplices to fulfill a sufficient descent condition, which is not quite in the spirit of the original NM algorithm. The original NM algorithm requires only simple descent between consecutive iterates. Furthermore only the relative ordering of the simplex vertices according to the CF value is needed by the algorithm. Rykov [13] has also published a class of convergent NM methods where ideas like the pseudo-expansion of Coope, Price, and Byatt first appeared in more general form.

Lately convergence properties of direct search descent methods have received a lot of attention. Most of this attention went to the family of methods which accept a trial point if it decreases the best-yet CF value (descent methods). The original NM algorithm and SDNMA concentrate on improving the worst point of the simplex, which is not the best-yet point. With this peculiarity in mind we can still treat them as descent methods if we consider only the best point of the simplex. This point is not allowed to move to a position with a higher CF value in any of the above mentioned algorithms.

Currently two approaches are known for ensuring the convergence of such methods to a local minimizer of the CF. The first approach is based on a sufficient descent requirement imposed on CF values at consecutive iterates (e.g. SDNMA). The amount of sufficient descent is supposed to gradually become 0 as the method converges to a set of stationary points of the CF [2]. The second approach requires only simple descent from the accepted trial point, but on the other hand all trial points must lie on a grid [3]. The grid is then gradually refined as the method approaches a stationary point of the CF. To the authors' best knowledge there exists no variant of the NM algorithm that would take advantage of the grid-based approach at the moment.

The original NM algorithm manipulates a set of $n + 1$ vertices in \mathbb{R}^n . Let x^1, x^2, \dots, x^{n+1} denote these vertices. Simplex ordering relabels the vertices according to the CF value so that $f(x^1) \leq f(x^2) \leq \dots \leq f(x^{n+1})$. For the sake of simplicity of notation f^i is used for the CF value at x^i . The centroid of the n vertices with the lowest CF value is defined as $x^{cb} = \frac{1}{n} \sum_{i=1}^n x^i$. The centroid and x^{n+1} define the line along which candidates are examined for replacing the vertex with the highest CF value. The examined points can be

GRID RESTRAINED NELDER-MEAD ALGORITHM

expressed as

$$x(\gamma) = x^{cb} + \gamma(x^{cb} - x^{n+1}). \quad (1.2)$$

They are denoted by x^r , x^e , x^{oc} , and x^{ic} with the corresponding values of γ denoted by γ_r , γ_e , γ_{oc} , and γ_{ic} . They are often referred to as the reflection, expansion, outer contraction, and inner contraction points. Under certain circumstances the simplex is shrunk towards x^1 using the formula $x^1 + \gamma_s(x^i - x^1)$ for $i = 2, 3, \dots, n + 1$.

Values of γ satisfy the following requirements

$$0 < \gamma_r < \gamma_e, \quad \gamma_e > 1, \quad 0 < \gamma_{oc} < 1, \quad -1 < \gamma_{ic} < 0, \quad 0 < \gamma_s < 1. \quad (1.3)$$

Nelder and Mead proposed the following values: $\gamma_r = 1$, $\gamma_e = 2$, and $\gamma_{oc} = -\gamma_{ic} = \gamma_s = 0.5$. Algorithm 1 summarizes the steps taken in one iteration of the NM algorithm as stated by [6]. This algorithm differs slightly from the original version in [11] where several ambiguities are present. The initial simplex can be chosen randomly or using some predefined rules. The CF is evaluated at the simplex vertices upon which iterations of algorithm 1 are repeated until some stopping condition is satisfied.

Algorithm 1. One iteration of the Nelder-Mead simplex algorithm.

1. Order the simplex.
2. Evaluate $f^r = f(x^r)$. If $f^r < f^1$ evaluate $f^e = f(x^e)$.
If $f^e < f^r$ replace x^{n+1} with x^e , otherwise replace it with x^r .
3. If $f^1 \leq f^r < f^n$, replace x^{n+1} with x^r .
4. If $f^n \leq f^r < f^{n+1}$, evaluate $f^{oc} = f(x^{oc})$.
If $f^{oc} \leq f^{n+1}$ replace x^{n+1} with x^{oc} .
5. If $f^{n+1} \leq f^r$, evaluate $f^{ic} = f(x^{ic})$.
If $f^{ic} \leq f^{n+1}$, replace x^{n+1} with x^{ic} .
6. If x^{n+1} wasn't replaced, shrink the simplex towards x^1 .

The remainder of the paper is divided as follows. In the second section relevant properties of positive bases and grids are discussed, upon which the principle of grid restraint is introduced. The effects of grid restraint on the properties of positive bases are examined and the requirements for convergence to a stationary point of the CF are given. The third section introduces the grid-restrained Nelder-Mead algorithm (GRNMA). The convergence properties of GRNMA are analysed. The fourth section discusses implementation details of the algorithm. Special attention is paid to the effects of finite precision floating point arithmetic. In the final section the results of numerical trials on the Moré-Garbow-Hillstrom [10] set of test problems for unconstrained optimization are presented. The performance of GRNMA is compared to the performance of SDNMA. Finally the conclusions are given.

Notation. Vectors are denoted by lowercase letters and are assumed to be column vectors so that $x^T y$ denotes the scalar product of x and y , Matrices are denoted by uppercase

letters e.g. A . The corresponding lowercase letter with a superscript is reserved for matrix columns (e.g. a^i). Set members are also denoted with a superscript. Members of a sequence $\{x_k\}_{k=1}^{\infty}$ are denoted by a subscript (e.g. x_k). Calligraphic uppercase letters are reserved for transformations and sets. \mathbb{R} and \mathbb{Z} denote the set of real and integer numbers respectively. The remaining notation is introduced in the text as needed.

2. Positive bases, grids, and grid restraintment

A set of nonzero vectors $\mathcal{B}^+ = \{b^1, b^2, \dots, b^r\}$ positively spans \mathbb{R}^n if every $x \in \mathbb{R}^n$ can be expressed as a nonnegative linear combination of members of \mathcal{B}^+ ($x = \sum_{i=1}^r \alpha_i b^i$ with $\alpha_i \geq 0$). \mathcal{B}^+ is a positive basis [4] for \mathbb{R}^n if it positively spans \mathbb{R}^n and no $b \in \mathcal{B}^+$ can be expressed as a nonnegative linear combination of vectors from $\mathcal{B}^+ \setminus \{b\}$. A positive basis must have at least $n + 1$ members (minimal positive basis) and can have at most $2n$ members (maximal positive basis). For every positive basis \mathcal{B}^+ there exists a subset $\mathcal{B} \subset \mathcal{B}^+$ containing n vectors such that \mathcal{B} is a linear basis for \mathbb{R}^n .

Positive bases have a useful property pointed out by Torczon [15] for $r = 2n$ and later more generally by Lewis and Torczon [7] for $n + 1 \leq r \leq 2n$.

$$\exists \epsilon > 0 \text{ such that } \forall x \in \mathbb{R}^n \exists b \in \mathcal{B}^+ : x^T b \geq \epsilon \|x\| \|b\| \quad (2.1)$$

For a given positive basis \mathcal{B}^+ ϵ has an upper bound $\epsilon_0(\mathcal{B}^+)$

$$\epsilon \leq \epsilon_0(\mathcal{B}^+) = \min_{x \in \mathbb{R}^n, \|x\| \neq 0} \max_{b \in \mathcal{B}^+} \frac{x^T b}{\|x\| \|b\|} \quad (2.2)$$

In [15] and [7] an estimate of ϵ_0 is derived. Consider the case of an orthogonal maximal positive basis where $\mathcal{B}^+ = \{b^1, b^2, \dots, b^n, -b^1, -b^2, \dots, -b^n\}$ and $(b^i)^T b^j \neq 0$ only for $i = j$. For such a positive basis ϵ_0 is maximal ($\epsilon_0 = n^{-1/2}$) considering all possible positive bases for \mathbb{R}^n .

Lemma 2.1. *If (2.1) holds for a set of vectors \mathcal{B}^+ then that set positively spans \mathbb{R}^n .*

Proof: Let $x^0 = x$ be some vector from \mathbb{R}^n and let b^i denote the member of \mathcal{B}^+ for which $(x^{i-1})^T b^i / (\|b^i\| \|x^{i-1}\|)$ is maximal. Let $\alpha^i b^i$ denote the projection of x^{i-1} on the direction of b^i (with $\alpha^i = (x^{i-1})^T b^i / \|b^i\|^2 \geq 0$ due to (2.1)). According to (2.1) the norm of $\alpha^i b^i$ is bound from below by $\epsilon \|x^{i-1}\|$.

If we express the norm of $x^{i-1} - \alpha^i b^i$ and replace α^i with $(x^{i-1})^T b^i / \|b^i\|^2$, we get

$$\|x^{i-1} - \alpha^i b^i\|^2 = \|x^{i-1}\|^2 - ((x^{i-1})^T b^i / \|b^i\|)^2$$

By taking into account the lower bound on the norm of $\alpha^i b^i$ we get

$$\|x^{i-1} - \alpha^i b^i\|^2 \leq \|x^{i-1}\|^2 (1 - \epsilon^2) \quad (2.3)$$

From $x^i = x^{i-1} - \alpha^i b^i$ and (2.3) follows by induction $\|x^i\| \leq \|x^0\|(1 - \epsilon^2)^{i/2}$. $\|x^i\|$ approaches zero geometrically. Therefore $\sum_{i=1}^{\infty} \alpha^i b^i$ is convergent and due to the way the series $\{\alpha^i b^i\}_{i=1}^{\infty}$ was constructed, converges to x .

The set \mathcal{B}^+ is finite and $x = \sum_{i=1}^{\infty} \alpha^i b^i$ can be split into partial sums such that $x = \sum_{j=1}^{|\mathcal{B}^+|} b^j \sum_{k=1}^{\infty} \alpha^{i,j,k} = \sum_{j=1}^{|\mathcal{B}^+|} b^j \beta^j$. Since all α^i are nonnegative, all β^j must also be nonnegative. \square

Lemma 2.2. *If (2.1) holds for a finite set of vectors \mathcal{B}^+ then that set contains a subset which is a positive basis for \mathbb{R}^n .*

Proof: Due to Lemma 2 \mathcal{B}^+ positively spans \mathbb{R}^n . Now choose $b^* \in \mathcal{B}^+$ such that it can be expressed as a nonnegative linear combination of remaining members of \mathcal{B}^+ . If we remove b^* from \mathcal{B}^+ the resulting set still positively spans \mathbb{R}^n . We can repeat this until we can find no such b^* . When we are done the resulting \mathcal{B}^+ still positively spans \mathbb{R}^n and no member $b \in \mathcal{B}^+$ can be expressed as a nonnegative linear combination of members of $\mathcal{B}^+ \setminus \{b\}$. \square

(2.1) and (2.2) assure us that for every positive basis \mathcal{B}^+ and every direction x there exists a member of \mathcal{B}^+ for which the angle θ between x and b is bound ($0 < \theta \leq \arccos \epsilon_0(\mathcal{B}^+)$). In other words this means that for C^1 functions we can always find descent along some $b \in \mathcal{B}$ as long as the step in that direction is not too long and we are not at a stationary point of $f(x)$.

Lemma 2.3. *Let \mathcal{B}^+ be a positive basis for \mathbb{R}^n with r members and $a \in \mathbb{R}^n$. Then*

$$a^T b \geq 0 \quad \forall b \in \mathcal{B}^+ \implies a = 0$$

Proof: Since \mathcal{B}^+ is a positive basis, we can write $-a = \sum_{i=1}^r \alpha_i b_i$ where all $\alpha_i \geq 0$. Now $0 \geq -a^T a = \sum_{i=1}^r \alpha_i a^T b_i \geq 0$. This is true only if $a = 0$. \square

Definition 2.4. A limit point of a sequence of positive bases $\{\mathcal{B}_i^+\}_{i=1}^{\infty}$ is a set \mathcal{B}_{∞}^+ with cardinality r for which a subsequence $\{\mathcal{B}_{i_j}^+\}_{j=1}^{\infty}$ of sets with cardinality r exists such that

$$\lim_{j \rightarrow \infty} \min_{b \in \mathcal{B}_{i_j}^+} \|b - b_{\infty}\| = 0, \quad \forall b_{\infty} \in \mathcal{B}_{\infty}^+ \tag{2.4}$$

Lemma 2.4. *Suppose the sequence $\{\mathcal{B}_i^+\}_{i=1}^{\infty}$ of positive bases for \mathbb{R}^n fulfills the following requirements*

$$\limsup_{i \rightarrow \infty} \max_{b \in \mathcal{B}_i^+} \|b\| < \infty, \tag{2.5}$$

$$\liminf_{i \rightarrow \infty} \min_{b \in \mathcal{B}_i^+} \|b\| > 0, \tag{2.6}$$

$$\liminf_{i \rightarrow \infty} \epsilon_0(\mathcal{B}_i^+) > 0. \tag{2.7}$$

Then at least one \mathcal{B}_{∞}^+ exists and contains a subset which is a positive basis for \mathbb{R}^n .

Proof: Equation (2.5) causes the members of positive bases to be bound from above in norm. Therefore at least one limit point \mathcal{B}_∞^+ of the sequence must exist. Due to (2.6) this limit point can't have members which are zero vectors. Finally (2.7) ensures that (2.1) holds for \mathcal{B}_∞^+ and Lemma 2.2 guarantees the existence of a subset which is a positive basis for \mathbb{R}^n . \square

A grid $\mathcal{G}(z, \mathcal{B}, \Delta)$ is a set of points defined as

$$\mathcal{G}(z, \mathcal{B}, \Delta) = \left\{ x : x = z + \sum_{i=1}^n N^i b^i \Delta^i, N^i \in \mathbb{Z} \right\} \quad (2.8)$$

where z is the grid origin, \mathcal{B} is the generating set of linear basis vectors, and $\Delta = [\Delta^0, \Delta^1, \dots, \Delta^n]$ is the grid scaling vector. Without the loss of generality we can assume that $\|b\| = 1$ for all $b \in \mathcal{B}$. The intersection of a grid and any compact set \mathcal{C} is a finite set of points.

In [3] the grid is generated by a linear basis \mathcal{B} . But since the framework of [3] allows moves in the remaining $r - n$ directions that complement \mathcal{B} into a positive basis \mathcal{B}^+ , an additional restriction is introduced. The remaining $r - n$ directions must be integer linear combinations of the first n directions. This guarantees that the points produced by the algorithm remain on the grid.

In the remainder of the paper we assume $\mathcal{B} = \{e^1, e^2, \dots, e^n\}$, where e^i are unit vectors constituting the orthogonal basis for \mathbb{R}^n . The notation $\mathcal{G}(z, \Delta)$ is short for $\mathcal{G}(z, \{e^1, e^2, \dots, e^n\}, \Delta)$. Sometimes, when it is clear which z and Δ are meant, the notation \mathcal{G} will be used instead of $\mathcal{G}(z, \Delta)$.

Definition 2.6. A grid restraintment operator $\mathcal{R}(\mathcal{G}, x)$ is a transformation $\mathbb{R}^n \rightarrow \mathcal{G}$ with the following property

$$\begin{aligned} \forall x \in \mathbb{R}^n, x_G &= \mathcal{R}(\mathcal{G}, x) \\ \exists x' \in \mathcal{G} : \|x - x'\| &< \|x - x_G\| \end{aligned}$$

The restraintment error vector $\delta = x - x_G$ plays an important role in the convergence theorem of the grid-restrained algorithm. It is fairly easy to show that for an orthogonal grid $\mathcal{G}(z, \Delta)$ the restraintment error is bound from above by $\|\delta\| \leq \|\Delta\|/2 = \delta_0$.

The next lemma deals with the effect of grid restraintment on $\epsilon_0(\mathcal{D})$.

Lemma 2.7. *Suppose that \mathcal{D}^+ is a positive basis and \mathcal{D}_G^+ is the set of vectors obtained by restraining members of \mathcal{D}^+ to grid $\mathcal{G}(0, \Delta)$. Then the following relation holds*

$$\epsilon_0(\mathcal{D}_G^+) \geq \frac{\epsilon_0(\mathcal{D}^+) - \delta_0 / \|d^{\min}\|}{1 + \delta_0 / \|d^{\min}\|} \quad (2.9)$$

where $d^{\min} = \arg \min_{d \in \mathcal{D}^+} \|d\|$.

GRID RESTRAINED NELDER-MEAD ALGORITHM

Proof: Let x denote an arbitrary member of \mathbb{R}^n and d the corresponding member of \mathcal{D}^+ , for which (2.1) holds. Further let $d' = \mathcal{R}(\mathcal{G}, d) = d - \delta$. Due to $\|\delta\| \leq \delta_0$ we have

$$\|d'\| \leq \|d\| + \delta_0 \quad (2.10)$$

From (2.1) the following estimate can be obtained

$$x^T d' = x^T d + x^T \delta \geq \epsilon_0 \|x\| \|d\| - \delta_0 \|x\| = \|x\| \|d\| (\epsilon_0 - \delta_0 / \|d\|) \quad (2.11)$$

ϵ_0 is short for $\epsilon_0(\mathcal{D}^+)$. If we take into account (2.10) and (2.11) we get

$$\frac{x^T d'}{\|x\| \|d'\|} \geq \frac{\epsilon_0 - \delta_0 / \|d\|}{1 + \delta_0 / \|d\|} \quad (2.12)$$

This is an estimate for the cosine of the angle between x and basis vector $d' \in \mathcal{D}_G^+$ obtained with grid restraintment of $\arg \max_{d \in \mathcal{D}^+} x^T d / (\|x\| \|d\|)$. δ_0 is the upper bound on restraintment error.

From (2.12) the lower bound on $\epsilon_0(\mathcal{D}_G^+)$ can now be expressed

$$\begin{aligned} \min_{x \in \mathbb{R}^n, \|x\| \neq 0} \max_{d' \in \mathcal{D}_G^+} \frac{x^T d'}{\|x\| \|d'\|} &\geq \min_{x \in \mathbb{R}^n, \|x\| \neq 0} \max_{d \in \mathcal{D}^+} \frac{x^T \mathcal{R}(\mathcal{G}, d)}{\|x\| \|\mathcal{R}(\mathcal{G}, d)\|} \geq \min_{d \in \mathcal{D}^+} \frac{\epsilon_0 - \delta_0 / \|d\|}{1 + \delta_0 / \|d\|} \\ &= \frac{\epsilon_0 - \delta_0 / \|d^{\min}\|}{1 + \delta_0 / \|d^{\min}\|} \end{aligned}$$

□

Lemma 2.7 ensures that a positive basis \mathcal{D}^+ will remain a positive basis after grid restraintment if

$$\delta_0 / \|d^{\min}\| < \epsilon_0(\mathcal{D}^+) \quad (2.13)$$

The next lemma represents the basis for the convergence theorems of many direct search algorithms.

Lemma 2.8. *Let $\{x_k\}_{k=1}^{\infty}$ be a sequence of points from \mathbb{R}^n , $\{\mathcal{B}_k^+\}_{k=1}^{\infty}$ a sequence of positive bases for \mathbb{R}^n , and $o(x)$ a function for which $\lim_{x \rightarrow 0} o(x)/x = 0$. Suppose the following requirements are satisfied*

1. *there exists a compact set \mathcal{C} subject to $\{x : f(x) \leq f(x_1)\} \subseteq \mathcal{C}$,*
2. *$f(x)$ is C^1 ,*
3. *$\forall k : f(x_k) \leq f(x_{k+1})$,*
4. *the sequence $\{\mathcal{B}_k^+\}_{k=1}^{\infty}$ satisfies requirements (2.5), (2.6), and (2.7),*

5. there exists a sequence of scalars $\{\xi_k\}_{k=1}^\infty$ such that

$$f(x_k + \xi_k b) \geq f(x) - o(\xi_k \|b\|) \quad \forall b \in \mathcal{B}_k^+, \quad (2.14)$$

$$\lim_{k \rightarrow \infty} \xi_k = 0. \quad (2.15)$$

Then $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$.

Proof: Due to requirements 1 and 3 the sequence $\{x_k\}_{k=1}^\infty$ remains inside the compact set \mathcal{C} and thus admits at least one limit point. We can always choose a subsequence of $\{x_k\}_{k=1}^\infty$ that converges to x_∞ and the corresponding subsequence of $\{\mathcal{B}_k^+\}_{k=1}^\infty$ that converges to \mathcal{B}_∞^+ for any combination of limit points x_∞ and \mathcal{B}_∞^+ . Let the set \mathcal{K} denote the indices of this subsequence.

For all $b \in \mathcal{B}_k^+$ we have

$$\begin{aligned} f(x_k + \xi_k b) - f(x_k) &= \int_{t=0}^{\xi_k} b^T (\nabla f(x_k + tb) - \nabla f(x_k) + \nabla f(x_k)) dt \\ &= \int_{t=0}^{\xi_k} b^T (\nabla f(x_k + tb) - \nabla f(x_k)) dt + \xi_k b^T \nabla f(x_k) \end{aligned}$$

Let $M(x, a)$ denote $\max_{0 \leq t \leq 1} \|\nabla f(x + ta) - \nabla f(x)\|$. If we also consider (2.14) we get

$$M(x_k, \xi_k b) + \frac{b^T \nabla f(x_k)}{\|b\|} \geq -\frac{o(\xi_k \|b\|)}{\xi_k \|b\|} \quad \forall b \in \mathcal{B}_k^+ \quad (2.16)$$

Now take the limit of (2.16) as $k \in \mathcal{K}$ approaches infinity. Due to requirement 2 and (2.15) $M(x, a)$ approaches 0. Since the right hand side also approaches zero we get

$$\frac{b^T \nabla f(x_\infty)}{\|b\|} \geq 0 \quad \forall b \in \mathcal{B}_\infty^+$$

From requirement 4 and Lemma 2.3 it follows then $\|\nabla f(x_\infty)\| = 0$. Since x_∞ is any limit point of $\{x_k\}_{k=1}^\infty$, we conclude $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$. \square

3. The grid-restrained simplex algorithm

To ensure convergence of the original NM algorithm we propose the restraintment of the examined points to a grid. The grid can be refined when certain conditions are satisfied in order to increase the precision of the search. Similarly as in [12] the proposed algorithm reshapes the simplex in order to satisfy the convergence requirements. The algorithm requires only simple descent, therefore the acceptance requirements for the outer contraction point (x^{oc}) and the inner contraction point (x^{ic}) are stricter, as these two are the last resort for the algorithm when it can't find an improvement over f^{n+1} . Acceptance requirements $f^{oc} < f^n$ and $f^{ic} < f^n$ are proposed to replace the ones in steps 4 and 5 of Algorithm 1. Making

the inequality more strict ensures that only a finite number of points are visited by the simplex algorithm if the grid remains unchanged. Shrink steps are also omitted and replaced by steps 7–9 of Algorithm 2 which constitute an algorithm conforming to the requirements of Lemma 2.8. This way convergence is guaranteed as is demonstrated later in this section.

In order to form a positive basis around x^1 the so-called pseudo-expand step was introduced in [12]. It is calculated using the centroid of the n points with highest CF value ($x^{cw} = \frac{1}{n} \sum_{i=2}^{n+1} x^i$) as $x^{pe} = x^1 + (\gamma_e/\gamma_r - 1)(x^1 - x^{cw})$. This point is also used in the grid-restrained algorithm.

The algorithm starts with constructing a simplex around the initial point x_0 . The initial grid $\mathcal{G}(z, \Delta)$ with $z = x_0$ is chosen. The CF is evaluated at the vertices of the initial simplex. Then iterations of Algorithm 2 are repeated until some stopping condition is satisfied.

Let $v^i = x^{i+1} - x^1$, $i = 1, 2, \dots, n$ denote the simplex side vectors, v^{\max} the longest, and v^{\min} the shortest side vector. In his paper [16] Tseng stated that the interior angles of the simplex should stay bounded away from zero. This requirement implies the following:

$$|\det V| = |\det[v^1, v^2, \dots, v^n]| \geq c \|v^{\max}\|^n. \quad (3.1)$$

In [12] a simple and effective way for calculating the determinant (3.1) was presented. The method relies on the fact that the points x^r , x^e , x^{oc} , and x^{ic} are colinear with x^{cb} and x^{n+1} , and that x^{pe} is colinear with x^{cw} and x^1 . Due to grid restraintment the colinearity is broken. We propose a different method which is somewhat more expensive, yet applicable to the grid-restrained simplex. The approach uses QR decomposition of V . Since Q is an orthogonal matrix whose columns have norm 1, the determinant of V is equal to the determinant of R . Since R is upper triangular (3.1) can be expressed as $\det V = \prod_{i=1}^n R_{ii}$. The same Q and R used for determining the simplex shape can later be used for reshaping the simplex. This way the price of doing a QR decomposition is justified.

Algorithm 2. The grid-restrained simplex algorithm.

1. Repeat iterations of the original NM algorithm without shrink steps and with modified acceptance criteria for contraction points. Instead of (1.2) use $x(\gamma) = \mathcal{R}(\mathcal{G}, x^{cb} + \gamma(x^{cb} - x^{n+1}))$.
When an iteration not replacing x^{n+1} (NM failure) is encountered, go to step 2.
2. $x^{best} = \arg \min_{x \in \{x^1, x^2, \dots, x^{n+1}\}} f(x)$ and $f^{best} = f(x^{best})$.
3. If the simplex shape violates (3.1), reshape it by forming an orthogonal basis $\mathcal{D} = \{d^1, d^2, \dots, d^n\}$ subject to $\lambda \frac{n^{1/2} \|\Delta\|}{2} \leq \|d^i\| \leq \Lambda \frac{n^{1/2} \|\Delta\|}{2}$ for all $i = 1, 2, \dots, n$. Construct a simplex comprising x^1 and $x^{i+1} = \mathcal{R}(\mathcal{G}, x^1 + d^i)$ where $i = 1, 2, \dots, n$, and evaluate CF at the new simplex vertices.
4. Order the simplex and evaluate CF at the grid-restrained pseudo-expand point to obtain $f^{pe} = f(\mathcal{R}(\mathcal{G}, x^{pe}))$. If $\min(f^{pe}, f^1, f^2, \dots, f^{n+1}) \geq f^{best}$ go to step 7
5. If $f^{pe} < f^{best}$ replace $x^i = x^{best}$ with x^{pe} .
6. Go to step 1.
7. If a reshape happened at step 3 set $l = 1$, otherwise set $l = 0$.
8. Repeat the following steps.

- (a) If no reshape happened in this iteration of the algorithm, reshape now. Otherwise reverse the basis vectors d^i .
 - (b) If $l \geq 2$ and $l \bmod 2 = 0$
 - Shrink the basis vectors by a factor of $0 < \gamma_s < 1$.
 - If $\|d^{\min}\| < \lambda \frac{n^{1/2}\|\Delta\|}{2}$ refine the grid by setting $z = z^{\text{new}} = x^1$ and $\Delta = \Delta^{\text{new}}$.
 - (c) Evaluate CF at $\mathcal{R}(\mathcal{G}, x^1 + d^i)$ for $i = 1, 2, \dots, n$.
 - (d) Set $l = l + 1$.
- Until stopping condition is satisfied or $\min_{d \in \mathcal{D}} f(\mathcal{R}(\mathcal{G}, x^1 + d)) < f^1$.
9. Construct a new simplex comprising x^1 and $\mathcal{R}(\mathcal{G}, x^1 + d^i)$ where $i = 1, 2, \dots, n$.
 10. If stopping condition is satisfied finish, else go to step 1.

d^{\max} and d^{\min} denote the longest and the shortest member of \mathcal{D} . λ and Λ are constants used in bounds on the simplex side length. For the sake of convergence analysis we assume that the optimization takes an infinite number of iterations described in Algorithm 2. This can be achieved by removing the stopping condition. Let x^m denote the point with the highest CF value encountered by Algorithm 2. The first step towards proving the convergence to a stationary point is the following lemma.

Lemma 3.1. *If there exists a compact set \mathcal{C} such that $\{x : f(x) \leq f(x^m)\} \subseteq \mathcal{C}$, only a finite number of CF evaluations may take place before the grid is refined.*

Proof: The intersection of \mathcal{C} and grid $\mathcal{G}(z, \Delta)$ is a finite set of points. Since all examined points come from $\mathcal{G}(z, \Delta)$, only a finite number of points can be accepted into the simplex. Strict inequality with respect to the second worst point in all the acceptance criteria for the candidate point of the simplex algorithm makes sure that the algorithm can't get trapped in an infinite loop. Such a situation occurs when due to grid restraintment the worst point of the simplex doesn't change regardless of the simplex steps that are performed. Therefore the number of consecutive ordinary NM iterations in step 1 is finite. Before another sequence of such iterations is started a point with CF lower than f^1 can appear in steps 3, 4, or 8. Such a point is of course accepted into the simplex. But this can happen only a finite number of times if the grid isn't refined. So eventually the algorithm ends up at step 8b with $l \geq 2$ and $l \bmod 2 = 0$ causing a shrink step to be performed. At most $\lceil \log(\lambda/\Lambda) / \log(\gamma_s) \rceil + 1$ shrink steps are needed for $\|d^{\min}\| < \lambda \frac{n^{1/2}\|\Delta\|}{2}$ to hold. At that point the grid is refined. \square

The consequence of Lemma 3.1 is that there exists an infinite sequence $\{x_k\}_{k=1}^{\infty}$ of simplex points with the lowest CF value (x^1) collected immediately before steps 8(a) that are followed by a grid refinement. Let $\{\mathcal{D}_k\}_{k=1}^{\infty}$, $\{\Delta_k\}_{k=1}^{\infty}$, and $\{\mathcal{G}_k\}_{k=1}^{\infty}$ denote the corresponding sequences of orthogonal bases, grid scaling vectors, and grids.

Theorem 3.2. *Suppose that there exists a compact set \mathcal{C} such that $\{x : f(x) \leq f(x^m)\} \subseteq \mathcal{C}$ and that $f(x)$ is C^1 . If $\|\Delta_k\| \rightarrow 0$ and $1 < \lambda \leq \Lambda$ then $\|\nabla f(x_k)\| \rightarrow 0$.*

Proof: Obviously requirements 1, 2, and 3 of Lemma 2.8 hold.

To show that requirement 4 of Lemma 2.8 holds, we construct a sequence of maximal orthogonal positive bases $\{\mathcal{A}_k^+\}_{k=1}^{\infty}$ from sequence $\{\mathcal{D}_k\}_{k=1}^{\infty}$ where $\mathcal{A}_k^+ = \{d_k^1, d_k^2, \dots, d_k^n, -d_k^1\}$,

$-d_k^2, \dots, -d_k^n$. Since \mathcal{D}_k is orthogonal, $\epsilon_0(\mathcal{A}_k) = n^{-1/2}$. Now create a sequence $\{\mathcal{B}_k^+\}_{k=1}^\infty$ by restraining members of \mathcal{A}_k^+ to \mathcal{G}_k and then scaling them with $\frac{2}{n^{1/2}\|\Delta_k\|}$. Since \mathcal{G}_k is an orthogonal grid the restraintment error δ is bound from above by $\delta_0 = \|\Delta_k\|/2$. Taking into account the restrictions in step 3 of Algorithm 2

$$\lambda \frac{n^{1/2}\|\Delta_k\|}{2} \leq \|d_k^i\| \leq \Lambda \frac{n^{1/2}\|\Delta_k\|}{2} \quad \forall k, i : k > 0, i = 1, 2, \dots, n$$

After the grid restraintment of \mathcal{A}_k^+ we have

$$\max\left(0, \lambda \frac{n^{1/2}\|\Delta_k\|}{2} - \delta_0\right) \leq \|\mathcal{R}(\mathcal{G}, \pm d_k^i)\| \leq \Lambda \frac{n^{1/2}\|\Delta_k\|}{2} + \delta_0$$

$$\forall k, i : k > 0, i = 1, 2, \dots, n$$

Finally after scaling

$$\max(0, \lambda - n^{-1/2}) \leq \|b_k^i\| \leq \Lambda + n^{-1/2} \quad \forall k, i : k > 0, i = 1, 2, \dots, 2n \quad (3.2)$$

Taking into account $\lambda > 1$ and (3.2) it follows that members of $\{\mathcal{B}_k^+\}_{k=1}^\infty$ satisfy (2.5) and (2.6). \mathcal{B}_k^+ is obtained with grid restraintment of \mathcal{A}_k^+ , so we have (see lemma 2)

$$\epsilon_0(\mathcal{B}_k^+) \geq \frac{\epsilon_0(\mathcal{A}_k^+) - \delta_0/\|d^{\min}\|}{1 + \delta_0/\|d^{\min}\|} \geq n^{-1/2} \frac{1 - \lambda^{-1}}{1 + \lambda^{-1}n^{-1/2}} \quad (3.3)$$

Due to $\lambda > 1$ and (3.3) it follows that (2.7) is satisfied by every member of $\{\mathcal{B}_k^+\}_{k=1}^\infty$.

Algorithm 2 requires no sufficient descent so we can set $o(x) = 0$ in (2.14). By choosing $\xi_k = \frac{n^{1/2}\|\Delta_k\|}{2}$ we satisfy (2.15). $x_k + \xi_k b_k^i$ for $i = 1, 2, \dots, 2n$ represent the $2n$ grid-restrained points examined around x_k in steps 3 and 8c of Algorithm 2. These points failed to produce descent. So (2.14) holds, fulfilling requirement 5 of Lemma 2.8 and resulting in $\|\nabla f(x_k)\| \rightarrow 0$. \square

4. Implementation

Algorithm 2 was implemented in MATLAB R12 and tested on an AMD ATHLON 2100XP computer. Following values were chosen for the simplex scaling coefficients: $\gamma_r = 1$, $\gamma_e = 1.2$, $\gamma_{oc} = 0.5$, $\gamma_{ic} = -0.5$, and $\gamma_s = 0.25$. Note that the value of γ_s is the same as the value of κ in [12] and has the same role in GRNMA as in SDNMA. γ_e also differs from the value used in [11] and [12] (where $\gamma_e = 2$). A more detailed explanation can be found in the section dealing with the numerical testing of GRNMA.

The initial simplex is constructed around the initial point x^0 ($x^1 = x^0$). Further n points are obtained by perturbing one of the n coordinates of x^0 by 5% or 0.00025 if the respective coordinate value is zero. The initial grid origin is set to x^0 . Initial grid spacing is chosen to be $\Delta^1 = \Delta^2 = \dots = \Delta^n = \frac{1}{10} \min_{i=1,2,\dots,n} \|x^{i+1} - x^1\|$. The initial simplex is not restrained to the grid.

Whenever an iteration of the NM algorithm fails step 3 checks the shape of the simplex. For that purpose the side vectors v^i are ordered so that $\|v^1\| \geq \|v^2\| \geq \dots \geq \|v^n\|$ and a matrix $V = [v^1, v^2, \dots, v^n]$ is constructed. V is then factored using QR decomposition in orthogonal matrix $Q = [q^1, q^2, \dots, q^n]$ and upper triangular matrix R . Since $\det V = \det R = \prod_{i=1}^n R_{ii}$, a lower bound on $|\det V|$ can be imposed by monitoring the diagonal elements of R . A reshape is performed if $\min_{i=1,2,\dots,n} |R_{ii}| \geq \psi \frac{n^{1/2} \|\Delta\|}{2}$ (where ψ is a constant) is violated.

Simplex reshaping constructs an orthogonal basis $\mathcal{D} = \{d^1, d^2, \dots, d^n\}$ from columns of Q .

$$d^i = \text{sign}(R_{ii}) \max \left(\lambda \frac{n^{1/2} \|\Delta\|}{2}, \min \left(|R_{ii}|, \Lambda \frac{n^{1/2} \|\Delta\|}{2} \right) \right) q^i$$

$$\text{sign}(R_{ii}) = \begin{cases} 1 & R_{ii} \geq 0 \\ -1 & R_{ii} < 0 \end{cases}$$

The following values were used in the implementation: $\psi = 10^{-6}$, $\lambda = 2$, and $\Lambda = 2^{52}$.

As the simplex is shrinking, the grid must be refined. Let z^i , Δ^i , z^{new} , $\Delta^{\text{new},i}$, and $d^{\text{min},i}$ be the components of z , Δ , z^{new} , Δ^{new} , and d^{min} respectively. Then components of Δ^{new} are expressed as $\Delta^{\text{new},i} = \min(\max(\frac{|d^{\text{min},i}|}{250\lambda n}, \frac{\|d^{\text{min}}\|}{250\lambda n^{3/2}}), \Delta^i)$. New grid origin is set to $z^{\text{new}} = x^1$.

Since $|d^{\text{min},i}| \leq \|d^{\text{min}}\|$, we can write $\Delta^{\text{new},i} \leq \frac{\|d^{\text{min}}\|}{250\lambda n}$. Due to the requirement $\|d^{\text{min}}\| < \lambda \frac{n^{1/2} \|\Delta\|}{2}$ that must be fulfilled in order for the grid refinement to take place, we have $\Delta^{\text{new},i} < \frac{\|\Delta\|}{500n^{1/2}}$. Finally we can conclude that $\|\Delta^{\text{new}}\| < \frac{\|\Delta\|}{500}$. This guarantees $\|\Delta\| \rightarrow 0$ required by Theorem 3.2.

Due to finite precision of floating point numbers (for 64-bit IEEE representation relative precision is $\tau_r = 2^{-52}$ and absolute precision is $\tau_a = 10^{-323}$) the allowed grid spacing $\|\Delta\|$ is limited. To prevent Δ^i from going below machine precision, an additional lower bound $\max(\tau_r z^{\text{new},i}, \tau_a)$ is enforced. So in step 8(b) of Algorithm 2 we actually have $\Delta^i = \max(\Delta^{\text{new},i}, \tau_r z^{\text{new},i}, \tau_a)$ for all $i = 1, 2, \dots, n$. In order to leave margin for division $\tau_a = 10^{-100}$ is used.

The simplex is ordered before the stopping condition is checked. Let $v^{i,j}$ and $x^{1,j}$ denote the j -th component of v^i and x^1 . The following stopping condition was used $\max_{i=2,3,\dots,n+1} |f^i - f^1| < \max(\beta_f, \beta_r |f^1|)$ and $\max_{i=1,2,\dots,n} |v^{i,j}| < \max(\beta_x, \beta_r |x^{1,j}|)$ for $j = 1, 2, \dots, n$. The following values of coefficients were used: $\beta_r = 10^{-15}$, $\beta_x = 10^{-8}$, and $\beta_f = 10^{-15}$.

5. Results of numerical testing

The algorithm was tested on the Moré-Garbow-Hillstom set of test problems [10]. The starting simplex was chosen in the same manner as in [12]. Additionally the standard quadratic function and the McKinnon function were tested. For the McKinnon function two runs were performed, one with the standard starting simplex, and one with the starting simplex proposed by McKinnon (marked with alt) that causes failure of the original NM algorithm. The results are summarized in Table 1.

GRID RESTRAINED NELDER-MEAD ALGORITHM

Table 1. Performance of SDNMA and GRNMA on a set of test functions. MS denotes the percentage of CF evaluations caused by non-NM steps. For GRNMA QR and QR_f denote the total number of QR decompositions and the number of QR decompositions where the result is not used for reshaping the simplex. The last column (N_s) contains the number of reshapes due to bad simplex shape.

Function	n	SDNMA		GRNMA				
		NF	Minimum	NF	Minimum	MS (%)	QR/QR_f	N_s
Rosenbrock	2	285	1.39058e-17	517	1.79285e-17	9.5	7/1	2
Freudenstein and Roth	2	217	48.9843	274	48.9843	18.2	6/0	1
Powell badly scaled	2	969	4.23980e-25	1245	1.87891e-25	3.7	8/2	1
Brown badly scaled	2	498	7.99797e-17	595	4.45581e-17	19.8	10/1	1
Beale	2	191	2.07825e-18	183	1.13556e-18	9.8	2/0	0
Jennrich and Sampson	2	157	124.362	149	124.362	16.1	4/0	0
McKinnon	2	426	-0.250000	380	-0.250000	27.6	37/26	0
McKinnon (alt)	2	351	-0.250000	210	-0.250000	35.2	14/6	0
Helical valley	3	342	9.83210e-16	591	1.64083e-16	8.1	6/0	3
Bard	3	1134	17.4287	427	8.21488e-3	7.7	3/0	1
Gaussian	3	194	1.12793e-8	252	1.12793e-8	20.2	9/4	1
Meyer	3	2801	87.9459	7269	87.9459	7.7	207/176	13
Gulf research	3	529	5.44511e-19	955	2.92451e-21	4.6	5/1	0
Box	3	478	8.70459e-21	923	1.91130e-20	3.1	5/1	1
Powell singular	4	1045	6.73509e-26	1280	3.43198e-25	8.9	14/5	1
Wood	4	656	2.57400e-16	1177	2.50092e-17	4.6	6/1	0
Kowalik and Osborne	4	653	3.07506e-4	566	3.07506e-4	12.4	6/2	0
Brown and Dennis	4	603	85822.2	620	85822.2	18.4	10/1	1
Quadratic	4	440	2.15350e-17	427	2.82657e-17	6.1	2/0	0
Penalty (1)	4	1848	2.24998e-5	1596	2.24998e-5	4.0	4/0	0
Penalty (2)	4	4689	9.37629e-6	2274	9.37629e-6	3.8	7/0	1
Osborne (1)	5	1488	5.46489e-5	1766	5.46489e-5	6.0	6/0	0
Brown almost linear	5	648	1.08728e-18	769	4.03372e-18	6.2	3/0	0
Biggs EXP6	6	4390	1.16131e-20	2877	1.12896e-20	3.3	11/5	0
Extended Rosenbrock	6	3110	1.35844e-14	2345	9.06455e-18	2.2	3/0	0
Brown almost linear	7	1539	1.51163e-17	1473	4.83079e-18	3.0	2/0	0
Quadratic	8	1002	8.07477e-17	1124	1.96893e-16	5.2	2/0	2
Extended Rosenbrock	8	5314	3.27909e-17	2996	1.50285e-17	3.1	4/0	0
Variably dimensional	8	2563	1.24784e-15	2634	7.66228e-16	2.6	4/0	0
Extended Powell	8	7200	6.43822e-24	7014	1.63762e-25	1.9	10/3	0
Watson	9	5256	1.39976e-6	5394	1.39976e-6	4.3	9/1	0
Extended Rosenbrock	10	7629	2.22125e-16	6208	1.77981e-17	1.3	3/0	0
Penalty (1)	10	9200	7.08765e-5	11514	7.08765e-5	1.6	6/0	0
Penalty (2)	10	32768	2.93661e-4	31206	2.93661e-4	0.8	19/7	2
Trigonometric	10	2466	2.79506e-5	1521	1.49481e-16	5.4	2/0	1
Osborne (2)	11	6416	0.0401377	3263	0.0401377	3.8	4/0	0
Extended Powell	12	20076	1.11105e-20	12846	5.51619e-28	0.9	3/0	1
Quadratic	16	2352	1.41547e-16	3639	4.70425e-16	3.1	2/0	0
Quadratic	24	4766	1.21730e-15	6067	4.06413e-16	2.8	2/0	0

A total of 39 test functions with dimension ranging from 2 to 24 were minimized. In all cases GRNMA found a stationary point of the CF. The results were compared to the results of the sufficient descent based algorithm proposed by Price, Coope, and Byatt (SDNMA).

Out of 39 test problem GRNMA found the same CF value on 15 problems and a better CF value on 17 problems. Upon examining the results for the remaining 7 problems, the coordinates of the point obtained by GRNMA agreed with the known function minimum in at least 6 significant digits. If we consider the number of CF evaluations GRNMA outperformed SDNMA (required less CF evaluations) on 19 test problems. In two out of these 19 problems (Bard 3D and trigonometric 10D function) GRNMA found a better minimum than SDNMA so the number of iterations is not comparable. On the other hand SDNMA found a better CF value than GRNMA on 7 test problems and required less CF evaluations on 20 test problems.

GRNMA produced the same or a better final CF value with fewer CF evaluations than SDNMA on 16 test problems (not counting the two problems which are not comparable due to different minima found by both algorithms). SDNMA produced the same or better final CF value with fewer CF evaluations than GRNMA on 13 test problems.

The quadratic functions exhibit a somewhat slower convergence with GRNMA than with SDNMA. On the other hand the number of GRNMA CF evaluations is still acceptable if we consider the fact that constructing a quadratic approximation of the n -dimensional CF takes $O(n^2)$ function evaluations. Such an approach yields an exact solution for quadratic functions after a finite number of CF evaluations. GRNMA obtained a solution with coordinate values matching 7 significant digits of the function's minimum. The quotient of the number of iterations needed for obtaining this result and n^2 gradually decreases with increasing n .

The percentage of function evaluations caused by non-NM steps in GRNMA is higher for functions with $n \leq 5$. For functions with $n > 5$ this value is around 6% or even less. The total percentage of non-NM steps across the whole test suite is 3.1%, meaning that NM steps are executed most of the time.

A QR decomposition is performed whenever the NM algorithm fails to produce descent. In such case the simplex shape is checked by evaluating the determinant of R . This check is often followed by a reshape (either in step 3 or step 8a) where the information obtained from the QR decomposition is used. If however no reshape happens, the QR decomposition is used solely for evaluating the determinant. SDNMA requires no decomposition for evaluating the determinant so in such a case GRNMA is in a somewhat worse position than SDNMA. The QR decomposition is however necessary due to the fact that grid restraintment makes it impossible to use SDNMA's incremental procedure for evaluating the determinant.

The total number of QR decompositions in GRNMA on the proposed set of test problems is 467 (this is also the number of pseudo-expand steps). 243 QR decompositions are not followed by a reshape. This and the fact that a QR decomposition happens only in 0.37% of all CF evaluations (126566) justifies the choice of QR decomposition for simplex shape monitoring. A total of 33 reshapes (last column of Table 1) happened in step 3 (these reshapes are caused by bad simplex shape).

Finally let us note that GRNMA uses $\gamma_e = 1.2$ and $\gamma_s = 0.25$ whereas the original NM algorithm uses $\gamma_e = 2$ and $\gamma_s = 0.5$. The comparison of the GRNMA using the

GRID RESTRAINED NELDER-MEAD ALGORITHM

Table 2. Performance of GRNMA for the original Nelder-Mead values $\gamma_e = 2, \gamma_s = 0.5$ and for the values used to obtain the GRNMA results in Table 1 ($\gamma_e = 1.2, \gamma_s = 0.2$). E and E_f denote the total number of expansion steps and the number of failed expansion steps.

Function	n	GRNMA, original NM			GRNMA in Table 1		
		NF	Minimum	E/E_f	NF	Minimum	E/E_f
Rosenbrock	2	390	1.07374e-17	56/25	517	1.79285e-17	151/86
Freudenstein and Roth	2	282	48.9843	35/15	274	48.9843	58/8
Powell badly scaled	2	1069	4.33716e-25	237/112	1245	1.87891e-25	401/119
Brown badly scaled	2	657	2.22190e-12	87/46	595	4.45581e-17	139/9
Beale	2	204	8.379224e-18	20/10	183	1.13556e-18	28/7
Jennrich and Sampson	2	163	124.3622	14/6	149	124.362	16/7
McKinnon	2	524	-0.250000	35/19	380	-0.250000	63/8
McKinnon (alt)	2	216	-0.250000	7/6	210	-0.250000	23/7
Helical valley	3	342	0.000000	50/14	591	1.64083e-16	133/54
Bard	3	390	0.00821488	40/11	427	0.00821488	86/11
Gaussian	3	207	1.12793e-8	19/16	252	1.12793e-8	30/11
Meyer	3	3656	87.9459	719/311	7269	87.9459	2064/466
Gulf research	3	666	1.27931e-22	94/32	955	2.92451e-21	262/43
Box	3	100008	7.57183e-4	6/1	923	1.91130e-20	256/52
Powell singular	4	1156	5.25964e-27	177/76	1280	3.43198e-25	291/42
Wood	4	876	1.83011e-17	152/79	1177	2.50092e-17	281/122
Kowalik and Osborne	4	712	3.07506e-4	98/42	566	3.07506e-4	116/30
Brown and Dennis	4	607	85822.2	75/32	620	85822.2	114/24
Quadratic	4	324	0.000000	26/11	427	2.82657e-17	66/25
Penalty (1)	4	2931	2.24998e-5	584/231	1596	2.24998e-5	443/83
Penalty (2)	4	3784	9.37629e-6	866/431	2274	9.37629e-6	644/92
Osborne (1)	5	1179	5.46490e-5	183/98	1766	5.46489e-5	426/98
Brown almost linear	5	710	1.09649e-17	96/53	769	4.03372e-18	135/43
Biggs EXP6	6	3878	1.21407e-19	711/317	2877	1.12896e-20	708/165
Extended Rosenbrock	6	1806	4.24663e-17	314/182	2345	9.06455e-18	540/152
Brown almost linear	7	1436	4.79774e-17	200/115	1473	4.83079e-18	283/103
Quadratic	8	1046	1.83990e-16	98/65	1124	1.96893e-16	163/74
Extended Rosenbrock	8	5110	3.82635e-16	906/527	2996	1.50285e-17	597/206
Variably dimensional	8	3579	7.69947e-16	608/297	2634	7.66228e-16	596/110
Extended Powell	8	4452	3.40101e-21	722/354	7014	1.63762e-25	1703/394
Watson	9	6470	1.39976e-6	1087/540	5394	1.39976e-6	1347/120
Extended Rosenbrock	10	11791	5.21129e-16	2006/1130	6208	1.77981e-17	1304/412
Penalty (1)	10	14256	7.08765e-5	2412/1278	11514	7.08765e-5	2680/582
Penalty (2)	10	35113	2.93661e-4	5426/2440	31206	2.93661e-4	7218/1355
Trigonometric	10	2344	4.47357e-7	317/241	1521	1.49481e-16	226/123
Osborne (2)	11	4323	0.0401377	646/485	3263	0.0401377	524/244
Extended Powell	12	21965	2.28441e-27	3709/2299	12846	5.51619e-28	2855/679
Quadratic	16	5389	6.35333e-17	626/475	3639	4.70425e-16	480/250
Quadratic	24	8454	1.25183e-15	867/697	6067	4.06413e-16	707/374

original NM values and the GRNMA presented here is in Table 2. The version with $\gamma_e = 2$ produced a better CF value on 8 test functions, the same CF value on 16 test functions, and it required less CF evaluations on 17 test functions. Note however that for the trigonometric 10D function the two versions are not comparable since they don't finish at the same local minimum. For the box 3D function the algorithm failed to converge to a local minimizer in 100000 CF evaluations, upon which it was stopped.

On the other hand the version using $\gamma_e = 1.2$ (also compared to SDNMA in Table 1) found a better CF value on 15 test problems, and it required less CF evaluations on 21 test problems (not counting the trigonometric 10D function). It 'won' (produced the same or better CF value in less iterations than its counterpart) on 20 test problems (again not counting the trigonometric 10D function), whereas the algorithm with the original value $\gamma_e = 2$ 'won' only on 12 problems. Therefore we conclude that on the set of test problems the modified values of γ_e and γ_s improve the performance of GRNMA with respect to the version using the original NM values. This is probably due to the fact that the percentage of failed expansion steps is smaller when $\gamma_e = 1.2$ is used (24% for $\gamma_e = 1.2$ compared to 54% for $\gamma_e = 2$). Expansion steps are the only ones that increase the simplex size (beside pseudo-expand steps and some reshape steps, both rather scarce in the process of optimization according to Table 1).

6. Conclusion

The grid-restrained Nelder-Mead algorithm (GRNMA) was presented. A proof for its convergence to a stationary point of C^1 cost functions was provided. GRNMA achieves convergence by restraining the examined points to a grid which is then gradually refined. The consequence of grid restraintment is that the algorithm requires only simple descent for converging to a stationary point. This is completely in the spirit of the original Nelder-Mead algorithm, where only the relative ordering of CF values matters.

Within the Moré-Garbow-Hillstrom test suite GRNMA exhibits similar performance as the sufficient descent Nelder-Mead algorithm (SDNMA). GRNMA performance was also tested with the standard NM expansion (γ_e) and shrink (γ_s) coefficient values. The obtained results are better if nonstandard values are used for γ_e and γ_s . This is probably due to the fact that more expansion steps get accepted.

GRNMA is yet another convergent modification of the well known, but provably not convergent original Nelder-Mead algorithm. Hopefully GRNMA and its convergent relatives (e.g. [16] and [12]) will help in shifting the scientific and engineering community away from using the unreliable original Nelder-Mead algorithm.

Acknowledgments

The authors would like to thank the anonymous referee for all the comments and suggestions that helped significantly in improving this paper.

The research was co-funded by the Slovenian Research Agency (Agencija za raziskovalno dejavnost RS) through the programme P2-0246 Algorithms and optimization methods in telecommunications.

References

1. D. Byatt, "A convergent variant of the Nelder-Mead algorithm," Master's thesis, Mathematics and Statistics Department, University of Canterbury, Christchurch, NZ, 2000.
2. I.D. Coope and C.J. Price, "Frame based methods for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 107, pp. 261–274, 2000.
3. I.D. Coope and C.J. Price, "On the convergence of grid-based methods for unconstrained optimization," *SIAM J. Optim.*, vol. 11, pp. 859–869, 2001.
4. C. Davis, "Theory of positive linear dependence," *American Journal of Mathematics*, vol. 76, pp. 733–746, 1954.
5. C.T. Kelley, "Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition," *SIAM Journal on Optimization*, vol. 10, pp. 43–55, 1999.
6. J.C. Lagarias, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, pp. 112–147, 1998.
7. R.M. Lewis and V.J. Torczon, "Rank ordering and positive bases in pattern search algorithms," ICASE NASA Langley Research Center, Hampton, VA 23681-0001, USA, Tech. Report 96-71, 1996.
8. R.M. Lewis, V.J. Torczon, and M.W. Trosset, "Direct search methods: Then and now," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191–207, 2000.
9. K.I.M. McKinnon, "Convergence of the Nelder-Mead simplex method," *SIAM Journal on Optimization*, vol. 9, pp. 148–158, 1998.
10. J.J. Moré, B.S. Garbow, and K.E. Hillstom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software*, vol. 7, pp. 17–41, 1981.
11. J.A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.
12. C.J. Price, I.D. Coope, and D. Byatt, "A convergent variant of the Nelder-Mead algorithm," *Journal of Optimization Theory and Applications*, vol. 113, pp. 5–19, 2002.
13. A.S. Rykov, "Simplex algorithms for unconstrained minimization," *Problems of Control and Information Theory*, vol. 12, pp. 195–208, 1983.
14. V.J. Torczon, "Multi-directional search: A direct search algorithm for parallel machines," PhD thesis, Department of Mathematical Sciences, Rice University, Houston, TX 77251-1892, USA, 1989.
15. V.J. Torczon, "On the convergence of pattern search methods," *SIAM J. Optim.*, vol. 7, pp. 1–25, 1997.
16. P. Tseng, "Fortified-descent simplicial search method: A general approach," *SIAM Journal on Optimization*, vol. 10, pp. 269–288, 1999.
17. M.H. Wright, "Direct search methods: Once scorned, now respectable," in *Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*, Dundee, UK, 1996, pp. 191–208.