# Gilding the Lily: a Variant of the Nelder-Mead Algorithm*

Larry Nazareth† and Paul Tseng‡

November 3, 1998

**Abstract**

We propose a variant of the Nelder-Mead algorithm derived from a reinterpretation of univariate golden-section direct search. In the univariate case, convergence of the variant can be analyzed analogously to golden-section search. In the multivariate case, we modify the variant by replacing strict descent with fortified descent and maintaining the interior angles of the simplex bounded away from zero. Convergence of the modified variant can be analyzed by applying results for a fortified-descent simplicial search method.

**Key Words.** Unconstrained minimization, Nelder-Mead algorithm, golden-section direct search.

---

* The second author is supported by National Science Foundation Grant CCR-9731273.
† Department of Pure and Applied Mathematics, Washington State University, Pullman, WA 99164-2930, U.S.A. Email: *nazareth@amath.washington.edu*.
‡ Department of Mathematics, University of Washington, Seattle, WA 98195, U.S.A. Email: *tseng@math.washington.edu*.
† ‡ This work was completed when its authors were Members of the Mathematical Sciences Research Institute (MSRI), Foundations of Computational Mathematics Program (Fall, 1998), whose support is gratefully acknowledged.

## 1 Introduction

Consider the problem of finding a (local) minimum of a real-valued function $f(x)$ of $n$ real variables. A well-known class of methods for solving this problem is that of direct search. Such methods iteratively update an initial guess of a solution using a few sampled function values along linearly independent directions (see [1, 3, 4, 6, 12, 16] and references therein). In the univariate case of $n = 1$, a popular direct search method is the golden-section search. This algorithm contains no heuristic (arbitrary) parameters, is easily shown to be convergent when $f$ is strictly unimodal, and has a close relationship with Fibonacci search, which is known to be optimal in a minimax sense [6]. In the multivariate case of $n > 1$, a popular direct search method is the Nelder-Mead (NM) algorithm [10], which extends the simplex method of Spendley et al. [13] by allowing non-isometric reflection steps.

In contrast to the golden-section search, the NM algorithm is an essentially heuristic approach that contains arbitrary parameters and possesses little in the way of convergence theory. In particular, McKinnon [8] constructed a family of strictly convex, coercive functions $f$ of $n = 2$ variables with different degrees of smoothness, on which the simplices generated by the NM algorithm, with particular choices for the initial simplex, contract to a non-stationary point. If $f$ is strictly convex and coercive (not necessarily differentiable), Lagarias et al. [5] showed that the simplices generated by the NM algorithm converge to the unique minimizer of $f$ in the case of $n = 1$ variable and that the diameter of the simplices converge to zero in the case of $n = 2$ variables. A nice survey of the NM algorithm and its variants is given in [16].

A simple reformulation of the golden-section search shows it to be quite closely related to the NM algorithm in one dimension. This suggests ways to enhance the NM algorithm, leading to a conceptually appealing variant that is more amenable to theoretical analysis and possibly also more effective in practice. In this paper, we consider such a variant. This variant inherits the convergence properties of the golden-section search for strictly unimodal $f$ in the univariate case and, by employing certain safeguards discussed in [15], similar convergence properties can be shown for pseudoconvex $f$ in the multivariate case. Some numerical experience with this variant is also given.

Throughout, we denote by $\Re^n$ the space of $n$-dimensional real column-vectors and, for any $x \in \Re^n$, by $x^T$ the transpose of $x$ and by $\|x\|$ the 2-norm of $x$ (i.e., $\|x\| = \sqrt{x^T x}$). For any set $S$ of $n+1$ vectors $x_1, ..., x_{n+1}$ in $\Re^n$, we denote $d(S) = \max_{i,j} \|x_i - x_j\|$ (the "diameter" of $S$) and

$$\nu(S) = |\det [\, x_2 - x_1 \quad \cdots \quad x_{n+1} - x_1 \,]\,| / d(S)^n,$$

i.e., $\nu(S)/n!$ is the volume of the $n$-dimensional simplex with vertices $(x_i - x_1)/d(S)$, $i = 1, ..., n+1$, and a diameter of 1. Thus, $\nu(S) = 0$ if and only if the corresponding simplex has one of its interior angles equal to zero or, equivalently, the edges emanating from each vertex of this simplex are linearly dependent.

## 2 Golden-Section Search

Figure 1 summarizes the golden-section direct search. Given boundary points $A$ and $E$ such that the minimum of the function is contained in the interval defined by them, two points $B$ and $C$ are placed so that $AC/AE = BE/AE = \alpha \equiv 1/\rho$, where $\rho = (\sqrt{5}+1)/2 \approx 1.618$ is the golden ratio (so $\rho^2 = \rho + 1$ and $\alpha^2 = 1 - \alpha$). If the function value at $C$ is no greater than that at $B$, then one can reduce the interval containing the minimum to that defined by $B$ and $E$. The use of the golden ratio ensures that $CE/BE = 1/\rho = \alpha$. Thus,

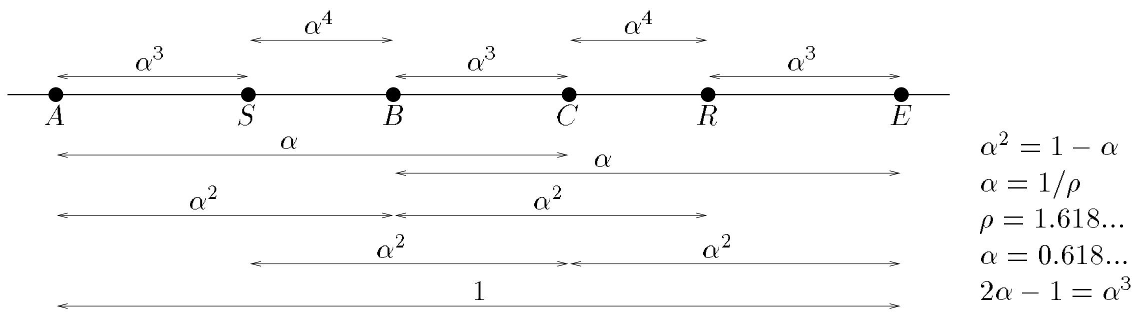


Figure 1: Golden Section Search

only one additional point, say $R$, needs to be placed in the new interval with $BR/BE = 1/\rho$. (An analogous statement holds for $A$ and $C$ when the value at $C$ is no smaller than that at $B$ leading to the placement of an additional point $S$.) Then the procedure is repeated. Luenberger [6] is a good reference for the procedure.

Figure 1 lists some additional properties satisfied by the intervals. Note in particular that $AB = BR$ and analogously $CE = SC$. Note also that $AB/BE = \alpha$.

When applied to a strictly unimodal function, the golden-section procedure converges to the unique minimum at a linear rate.

## 3 Nelder-Mead Algorithm

Figure 2 summarizes the NM algorithm for the case $n = 2$. For a general description see Bertsekas [1] or Wright [16]. The figure follows the one given in the latter reference. The point $z_r$ is an isometric reflection of $x_3$ in the centroid $\bar{x}$ of the other two points of the simplex, namely, $x_1$ and $x_2$. The points $z_2$, $z_1$ and $z_{-1}$ are the usual extrapolation and contraction points used in the NM algorithm.

## 4 Gilding the Lily

Let us reformulate the golden-section procedure summarized by Figure 1 in a way that closely resembles the operations of the NM algorithm in one dimension. The reformulation generates the golden-section points in a different order from that of Section 2 as follows:

Assume that the unique minimum of a strictly unimodal function $f$ is between the points $A$ and $E$. Assume that the function has been evaluated at $A$, $B$ and $E$ (henceforth these points are called a *golden-section triple*). Assume additionally that $f_A \geq f_B \leq f_E$, where $f_X$ means the value of $f$ at $X$. (Observe that $AB/BE = \alpha$. ) *Treat the line joining A and*

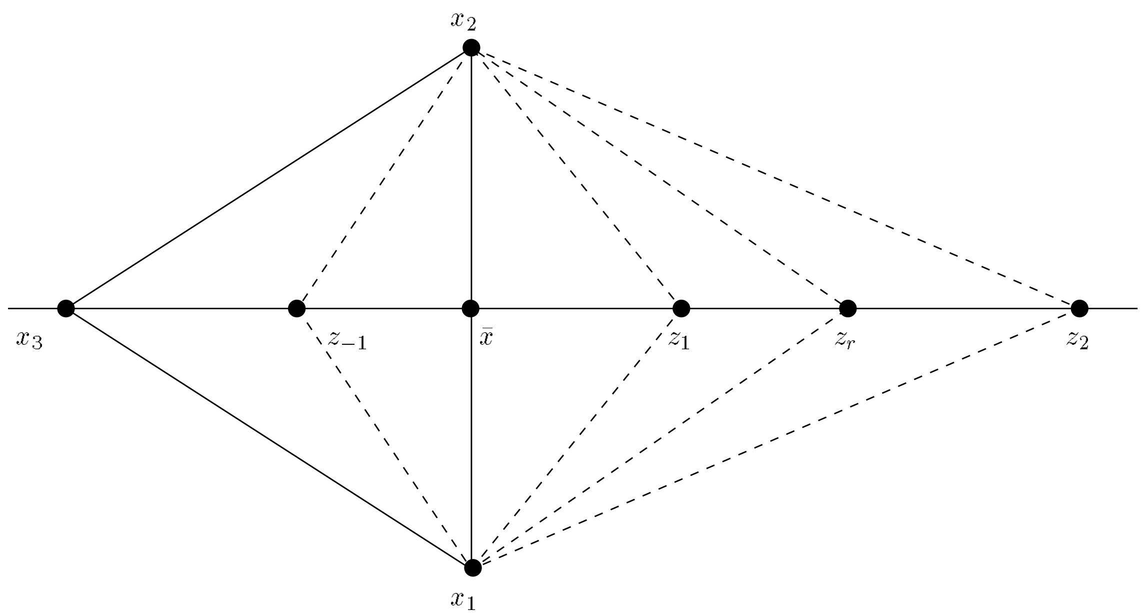


Figure 2: NM Algorithm

*B as a simplex in unidimensional space.*

1. Isometrically reflect $A$ about the point $B$ yielding $R$ and its associated function value $f_R$.
2. Suppose $f_R < f_B$. Then take the new simplex to be the line joining $E$ and $R$ and redefine new triple as follows: $A \leftarrow E$, $E \leftarrow B$ and $B \leftarrow R$. Repeat the procedure at this new golden-section triple.
3. Suppose $f_R \geq f_B$. Then evaluate the function at $C$.
If $f_R < f_A$ and $f_C < f_A$ either:

a) if $f_C \leq f_B$ then take the new simplex to be the line joining $R$ and $C$ and redefine the triple by $A \leftarrow R$, $E \leftarrow B$, $B \leftarrow C$
or b) if $f_C > f_B$ (but, as we know already, $f_C < f_A$) then redefine the simplex to be the points joining $C$ and $B$ and redefine the triple by $A \leftarrow C$, $E \leftarrow A$ and $B$ unchanged.
After a) or b) above, repeat the procedure at the new triple.
4. Otherwise, evaluate the function at $S$. If $f_S < f_A$ (as must happen because the function is strictly unimodal) then take the new simplex to be the line joining $B$ and $S$ and redefine the triple as $E \leftarrow A$, $A \leftarrow B$ and $B \leftarrow S$. Repeat the procedure.

The foregoing procedure generates a sequence of triples that span intervals of strictly decreasing length and converges at a linear rate. The resemblance of this procedure to the NM algorithm in one dimension[1] helps to motivate our incorporation of golden-section direct search techniques into the NM algorithm in multidimensions and our analysis of the resulting NM variant in one dimension. In essence, we superimpose Figure 1 on Figure 2 and make the identification:

$$x_3, z_{-1}, \bar{x}, z_1, z_r, z_2 \text{ with, respectively, } A, S, B, C, R, E.$$

Additionally, there may be advantages to defining $\bar{x}$ as a *weighted* centroid of points other than the worst one ($x_3$ in the case of $n = 2$ as in Figure 2), where the non-negative weight associated with the vertex $i$ is the difference between the function value at $x_3$ and the function value at vertex $i$. A weighted function value $\bar{f}$ (obtained in the same way as the weighted centroid by substituting the function value at vertex $i$ in place of the point $x_i$) is associated with $\bar{x}$. Then the initial simplex is defined by $A \equiv x_3$ and $B \equiv \bar{x}$ with associated function values $f(x_3)$ and $\bar{f}$ (estimated).

Some features of our proposed approach are as follows:

- By uniting a standard unidimensional direct search method with the multivariate NM approach, in a natural way, it helps to diminish the heuristic character of the latter and provides a rationale for choosing (otherwise arbitrary) parameters.

- One can guarantee convergence of the proposed NM variant on strictly unimodal univariate functions since it then reduces essentially to a form of golden-section search (see the proof of Theorem 1). This is in marked contrast to the approach taken in Lagarias et al. [5] where considerable effort is expended to show convergence of the (original) NM algorithm in the univariate case and only on strictly convex functions. With the proposed NM variant, one obtains convergence in the univariate case (for free) on the broader class of strictly unimodal functions.

- One can ensure convergence of the variant on pseudoconvex continuously differentiable functions of multivariables by employing safeguards as in Tseng [15] (see the next section and Theorem 2).

Below we describe formally our variant of the NM algorithm derived from golden-section search. At each iteration $k = 0, 1, ...$, the method updates an $n$-dimensional simplex, with vertex set $S^{(k)}$, by either moving the worst vertex (highest $f$-value) in the direction of a centroid of the remaining vertices, with a stepsize of $1/\rho$ or $\rho$ or 2 or $\rho^2$, to achieve descent (see Steps 3 and 4a below) or else contracting the simplex towards its best vertex (see Step 4b below).

[1] Notice that it is not completely *equivalent* to the NM algorithm in one dimension because this golden-section variant is permitted to change more than one vertex of the simplex at each iteration.

**Algorithm 1** *Choose any set* $S^{(0)}$ *of* $n+1$ *vectors in* $\Re^n$ *satisfying* $\nu(S^{(0)}) > 0$. *Choose any* $\theta_1 \in (0,1)$. *Let* $\rho = (\sqrt{5}+1)/2 = 1.618...$ *and let* $r = \log_\rho(2)$. *For* $k = 0, 1, ...$, *we generate* $S^{(k+1)}$ *and* $x^{(k)}$ *from* $S^{(k)}$ *as follows:*

**Step 1.** *(Reflect worst vertex about centroid). Express* $S = S^{(k)} = \{x_i\}_{i=1}^{n+1}$ *in ascending order of function value, i.e.,* $f_1 = f(x_1) \leq ... \leq f_{n+1} = f(x_{n+1})$. *Choose any set of scalars* $\{\mu_i\}_{i=1}^n$ *exceeding* $\theta_1/n$ *and summing to 1, and let*

$$\bar{x} = \sum_{i=1}^n \mu_i x_i, \qquad \bar{f} = \sum_{i=1}^n \mu_i f_i, \tag{1}$$

$$z_t = \rho^t(\bar{x} - x_{n+1}) + x_{n+1}, \qquad S_t = (S \backslash \{x_{n+1}\}) \cup \{z_t\}, \tag{2}$$

*where* $t > 0$. *Let* $x^{(k)} = \bar{x}$. *Go to Step 2.*

**Step 2.** *(Check reflected vertex for descent). If*

$$f(z_r) < f_n, \tag{3}$$

*then go to Step 3; else go to Step 4a.*

**Step 3.** *(Check to expand the reflection). If* $f(z_r) < f_1$ *and*

$$f(z_2) \leq f(z_r), \tag{4}$$

*then let* $S^{(k+1)} = S_2$; *else let* $S^{(k+1)} = S_r$. *Exit.*

**Step 4a.** *(Check to contract towards centroid). If* $f(z_r) < f_{n+1}$ *and*

$$f(z_1) < f_{n+1}, \tag{5}$$

*then let* $S^{(k+1)} = S_1$ *and exit; else if*

$$f(z_{-1}) < f_{n+1}, \tag{6}$$

*then let* $S^{(k+1)} = S_{-1}$ *and exit; else go to Step 4b.*

**Step 4b.** *(Contract simplex towards best vertex). Let* $S^{(k+1)} = x_1 + (S - x_1)/\rho^2$ *and exit.*

There are many choices for the weights $\mu_i$, such as $\mu_i = 1/n$ (corresponding to $\theta_1 = 1$) and, if $f_1 \neq f_{n+1}$,

$$\mu_i = (1-\theta_1)\frac{f_{n+1} - f_i}{\sum_{j=1}^n (f_{n+1} - f_j)} + \frac{\theta_1}{n}. \tag{7}$$

Also, the above algorithm admits variants such as (i) replacing (3) by $f(z_r) < \bar{f}$ or (ii) replacing (4) by $f(z_2) < f_1$ or (iii) replacing $z_r, z_1, S_r, S_1$ by, respectively, $z_1, z_{r-1}, S_1, S_{r-1}$. The subsequent convergence theory hold for these variants, but their practical performance may vary. In particular, using the isometric reflection $z_r$ seems crucial for good performance, as it better maintains the simplex from becoming deformed after a reflection.

## 5 Safeguards for Convergence and Termination

McKinnon's example of nonconvergence for the NM algorithm suggests that Algorithm 1 is unlikely to have desirable convergence properties for $n > 1$ unless safeguards are employed. We consider two safeguards discussed in [15], namely, replacing strict descent by a stronger criterion of fortified descent and maintaining the interior angles of the simplex $S$ bounded away from zero (i.e., $\nu(S)$ bounded away from zero). This leads to the following modified algorithm.

**Algorithm 2** *Choose any continuous functions* $\sigma : \Re_+ \mapsto \Re_+$ *and* $\beta : \Re_+ \mapsto \Re_+$ *satisfying* $\lim_{t\to 0} \sigma(t)/t = \lim_{t\to 0} \beta(t)/t = 0$ *and* $\inf_{t\geq a} \sigma(t) > 0$ *for all* $a > 0$. *Then execute as in Algorithm 1 but with the following two modifications, where* $\theta_2 \in (0, \nu(S^{(0)})]$ *is arbitrarily chosen.*

**1.** *Replace the criteria* (3), (4), (5), (6) *by, respectively,*

$$f(z_r) \leq f_n - \min\Big\{\sigma(d(S)), \theta_1\Big(f_{n+1} - \bar{f}\Big) - \beta(d(S))\Big\}, \tag{8}$$

$$f(z_2) \leq f(z_r), \qquad \nu(S_2) \geq \theta_2, \tag{9}$$

$$f(z_1) \leq f_{n+1} - \sigma(d(S)), \qquad \nu(S_1) \geq \theta_2. \tag{10}$$

$$f(z_{-1}) \leq f_{n+1} - \sigma(d(S)), \qquad \nu(S_{-1}) \geq \theta_2, \tag{11}$$

**2.** *At the beginning of Step 2, if* $\nu(S_r) < \theta_2$, *then instead let* $\tilde{z} = 2x_1 - x_{n+1}$ *and check if*

$$f(\tilde{z}) \leq f_1 - \min\Big\{\sigma(d(S)), \theta_1\Big(f_{n+1} - f_1\Big) - \beta(d(S))\Big\}. \tag{12}$$

*If yes, then let* $S^{(k+1)} = 2x_1 - S$ *and exit; else go to Step 4b.*

From a numerial standpoint, the criteria of fortified descent employed in the first modification is only slightly stronger than strict descent since we can choose $\alpha$ to be small everywhere (e.g., $\sigma(t) = 10^{-5}\min\{t^2, 1\}$), $\beta$ to have fast growth away from zero (e.g., $\beta(t) = 10^5 t^2$) and $\theta_1$ to be near 0. From a theoretical standpoint, the difference between the two is significant since convergence cannot be assured when fortified descent is relaxed to strict descent. The second modification is motivated by the multidimensional search method of Torczon [2, 14], where we consider reflecting all vertices about the best vertex $x_1$. In contrast to the latter method, we check the $f$-value at only one of the reflected vertices. This modification is needed to ensure that $\nu(S)$ stays above its threshold of $\theta_2$ (since reflecting all vertices about one vertex maintains $\nu$-value constant while reflecting one vertex about a centroid of the remaining vertices may decrease the $\nu$-value). The criterion (9) can alternatively be replaced by (cf. (8))

$$f(z_2) \leq f_1 - \min\Big\{\sigma(d(S)), \theta_1\Big(f_{n+1} - \bar{f}\Big) - \beta(d(S))\Big\}, \quad \nu(S_2) \geq \theta_2$$

(possibly with different $\alpha$, $\beta$ and $\theta_1$ than in Step 2) and our convergence result (i.e., Theorem 2) would still hold. However, in our tests, this alternative was not as effective as (9). Related discussions can be found in [15].

As described, Algorithms 1 and 2 each generates an infinite sequence of vectors in $\Re^n$. For practical implementations, a suitable termination criterion is needed. One such criterion, discussed in [15], is to terminate the method with the set of vectors $S = \{x_1, ..., x_{n+1}\}$ in $\Re^n$ whenever both $d(S)$ and the 2-norm of

$$\begin{bmatrix} (x_2 - x_1)^T \\ \vdots \\ (x_{n+1} - x_1)^T \end{bmatrix}^{-1} \begin{bmatrix} f_2 - f_1 \\ \vdots \\ f_{n+1} - f_1 \end{bmatrix} \tag{13}$$

are below a prespecified tolerance $\epsilon$, where $f_i = f(x_i)$. [It can be seen that if $\nu(S)$ is bounded away from zero and $d(S)$ tends to zero, then the vector (13) approaches $\nabla f(x_1)$, assuming $f$ is continuously differentiable. Thus, this criterion yields $\nabla f(x_1) \approx 0$ on termination.] If we wish to avoid the matrix inversion, we can replace the vector (13) by

$$\begin{bmatrix} (f_2 - f_1)/\|x_2 - x_1\| \\ \vdots \\ (f_{n+1} - f_1)/\|x_{n+1} - x_1\| \end{bmatrix}. \tag{14}$$

As long as $\nu(S)$ is bounded away from zero, this changes the 2-norm of the vector by only a constant factor. The above criterion worked well in our tests (see Sec. 7).

## 6 Convergence

In this section we analyze the convergence properties of Algorithm 1 for the case of $n = 1$ and of Algorithm 2 for the general case. The analysis of Algorithm 1 exploits its connection to golden-section search.

**Theorem 1** *Assume $n = 1$ and $f$ is strictly unimodal on $\Re$. Let $\{(S^{(k)}, x^{(k)})\}_{k=0,1,...}$ be generated by Algorithm 1. Then, either $\{d(S^{(k)})\} \to \infty$ and $f$ has no minimizer or $\{d(S^{(k)})\} \to 0$ linearly and $\{x^{(k)}\}$ converges to the unique minimizer of $f$.*

**Proof.** If an expansion step *is* taken at every iteration, then necessarily $\{d(S^{(k)})\} \to \infty$ and $\{f(x^{(k)})\}$ is strictly decreasing. Since $f$ is strictly unimodal, this implies $f$ has no minimizer. Suppose instead the algorithm does not take an expansion step at some iteration $k$. At the beginning of this iteration, we have a simplex $S^{(k)} = \{x_1, x_2\}$ with $f(x_1) \le f(x_2)$. Without loss of generality, we assume $x_2$ is to the left of $x_1$ on the real line. We consider the two cases: (i) $f(z_r) < f(x_1)$ or (ii) $f(z_r) \geq f(x_1)$. [We can identify $x_2, z_{-1}, x_1 = \bar{x}, z_1, z_r, z_2$ with, respectively, $A, S, B, C, R, E$ in Figure 1.]

In case (i), we go to Step 3. Since we do not take an expansion step, we must have $f(z_2) > f(z_r)$ so the new simplex is $S^{(k+1)} = \{x_1, z_r\}$. Moreover, $f(x_1) \geq f(z_r) \leq f(z_2)$ with $|x_1 - z_r|/|z_r - z_2| = 1/\alpha$. The strict unimodality of $f$ implies points to the right of $z_r$ have $f$-values greater than $f(z_r)$, so the next iteration $k+1$ must do a contraction, giving the next simplex either $S^{(k+2)} = \{z_r, z_2\}$ with $d(S^{(k+2)}) = d(S^{(k)})\alpha$ or $S^{(k+2)} = \{z_1, z_r\}$ with $d(S^{(k+2)}) = d(S^{(k)})\alpha^2$. Moreover, the following iteration $k + 2$ cannot take an expansion step.

In case (ii), we go to Step 4a. We have two subcases: (iia) $f(z_r) < f(x_2)$ and $f(z_1) < f(x_2)$ or (iib) $f(z_r) \geq f(x_2)$ or $f(z_1) \geq f(x_2)$. In subcase (iia), the new simplex is $S^{(k+1)} = \{x_1, z_1\}$ with $d(S^{(k+1)}) = d(S^{(k)})\alpha$. Moreover, depending on whether $f(x_1)$ or $f(z_1)$ is less, either $f(x_2) \geq f(x_1) \leq f(z_1)$ with $|x_1 - z_1|/|x_2 - x_1| = \alpha$ or $f(x_1) \geq f(z_1) \leq f(z_r)$ with $|x_1 - z_1|/|z_1 - z_r| = 1/\alpha$. In subcase (iib), the new simplex is always $S^{(k+1)} = \{z_{-1}, x_1\}$ with $d(S^{(k+1)}) = d(S^{(k)})\alpha^2$, regardless of whether we go to Step 4b. If $f(z_{-1}) \leq f(x_1)$, then we have $f(x_2) \geq f(z_{-1}) \leq f(x_1)$ with $|z_{-1} - x_1|/|x_2 - z_{-1}| = \alpha$. Otherwise, since we are in subcase (iib) and $f(x_2) \geq f(x_1)$, either $f(z_{-1}) \geq f(x_1) \leq f(z_1)$ with $|z_{-1} - x_1|/|x_1 - z_1| = \alpha$ or $f(z_{-1}) \geq f(x_1) \leq f(z_r)$ with $|z_{-1} - x_1|/|x_1 - z_r| = \alpha^2$. Thus, $S^{(k+1)} = \{a, b\}$ together with some $e$ satisfying $f(a) \geq f(b) \leq f(e)$ and $|a - b|/|b - e| \in \{1/\alpha, \alpha, \alpha^2\}$. In the subcase of $|a - b|/|b - e| \in \{1/\alpha, \alpha\}$, the strict unimodality of $f$ implies points outside the interval between $a$ and $e$ have $f$-value greater than that of $a$ or $e$, depending on which side of the interval the point lies, so the next iteration $k + 1$ cannot take an expansion step. In the case of $|a - b|/|b - e| = \alpha^2$, we must be in subcase (iib) and $a = z_{-1}, b = x_1, e = z_r$. Then, if an expansion step is taken at iteration $k + 1$, the next simplex must be $S^{(k+2)} = \{x_1, z_1\}$ with $f(x_1) > f(z_1)$. This together with $f(x_1) \leq f(z_r)$ implies that the following iteration $k + 2$ cannot take an expansion step. Moreover, since the simplex diameter can increase by a factor of at most $1/\alpha$ at each iteration, we have $d(S^{(k+2)}) \leq d(S^{(k+1)})/\alpha \leq d(S^{(k)})\alpha$.

To summarize, if we do not take an expansion step at an iteration $k$, then either (a) we do not take an expansion step at iteration $k+1$ and $d(S^{(k+1)}) \leq d(S^{(k)})\alpha$ or (b) we do not take an expansion step at iteration $k+2$ and $d(S^{(k+2)}) \leq d(S^{(k)})\alpha$. In addition, $S^{(k)} = \{a^{(k)}, b^{(k)}\}$ together with some $e^{(k)}$ satisfy $f(a^{(k)}) \geq f(b^{(k)}) \leq f(e^{(k)})$ and $|a^{(k)} - b^{(k)}|/|b^{(k)} - e^{(k)}| \in \{1/\alpha, \alpha, \alpha^2\}$ and $[a^{(k+1)}, e^{(k+1)}] \subset [a^{(k)}, e^{(k)}]$, etc. ■

The above proof shows that, in the univariate and strictly unimodal case, if Algorithm 1 does not take an expansion step at some iteration, then at subsequent iterations each generated simplex $S = \{a, b\}$ forms a "generalized" golden-section triple with some $e \in \Re$ in the sense that $b$ is between $a$ and $e$ and satisfies $f(a) \geq f(b) \leq f(e)$ and $|a - b|/|b - e| \in \{1/\alpha, \alpha, \alpha^2\}$.[2] So, for example, in case (i), either $S^{(k+2)} = \{z_r, z_2\}$ forms such a triple with $x_1$

[2]The need to generalize the golden-section triple introduced in Section 4 arises because only one vertex

or $S^{(k+2)} = \{z_1, z_r\}$ forms such a triple with $z_2$. Similarly, in subcase (iia), $S^{(k+1)} = \{x_1, z_1\}$ forms such a triple with either $x_2$ or $z_2$, depending on whether $f(x_1) \leq f(z_1)$.

The analysis of Algorithm 2 exploits the fact that this algorithm may be viewed as a special case of the method in [15] so that the convergence result therein may be applied. Recall that $f$ is said to be *quasiconvex* on $\Re^n$ [7] if

$$f(x + \gamma(y - x)) \leq \max\{f(x), f(y)\} \quad \forall \gamma \in [0, 1], \ \forall x, y \in \Re^n.$$

Notice that quasiconvexity generalizes unimodality to the multivariable case and it is weaker than convexity.

**Theorem 2** *Assume that $f$ is continuously differentiable and quasiconvex on $\Re^n$ and is uniformly continuous on $\{x \in \Re^n : f(x) \leq \min_{x \in S^{(0)}} f(x)\}$. Also assume $\inf_{x \in \Re^n} f(x) > -\infty$. Let $\{(S^{(k)}, x^{(k)})\}_{k=0,1,\ldots}$ be generated by Algorithm 2. Then $\{d(S^{(k)})\} \to 0$ and every cluster point of $\{x^{(k)}\}$ is a stationary point of $f$.*

**Proof.** Consider a special case of the fortified-descent simplicial search method in [15] whereby at each iteration we either choose (i) $m = \min\{n, F(S) * F(S^k)\}$, $S_0 = S_1$ in Step 1, $\Sigma_2 = S_2$, $\Sigma_3 = S_3$ in Step 2 or we choose (ii) $m = 1$ in Step 1, $\Sigma_2 \subset \arg\max_{x \in S} f(x)$ in Step 2. [The preceding notations are from [15].] As is remarked in [15, Sec. 3], since $f$ is quasiconvex, we have $F(S) * F(S^k) \geq n$ in Step 1 always and hence $m = n$ whenever we choose (i). Then it is straightforward, though a bit tedious, to check that Algorithm 2 is a special case of this method (with an iteration of Algorithm 2 corresponding to a visit to Step 1 in this method). Then, applying the convergence result for this method, namely [15, Lemma 3.1(b), Cor. 3.1(b)], and our theorem follows. ∎

As a consequence of Theorem 2, we have that if, in addition to the assumptions therein, it is assumed that $f$ has a unique stationary point on the level set $\{x \in \Re^n : f(x) \leq \min_{x \in S^{(0)}} f(x)\}$, then $\{x^{(k)}\}$ generated by Algorithm 2 converges to this stationary point (which in fact would be the global minimizer of $f$). We note that the assumptions on $f$ made in Theorem 2 allow for the possibility of nonunique minimizers or even unbounded set of minimizers. Also, by employing additional safeguards in Algorithm 2 as described in [15], it is possible to remove the assumption of $f$ being quasiconvex from Theorem 2.

# 7 Some Numerical Experience

To gain insight into the numerical behavior of the preceding algorithms and the effects of the algorithm parameters, we implemented and tested Matlab versions of Algorithms 1 and 2, and we describe our experience below.

---

of the simplex is changed at each iteration of Algorithm 1, in contrast to the motivating unidimensional algorithm at the beginning of Section 4—see also footnote 1.

We experimented with both choices of $\mu_i = 1/n$ and $\mu_i$ given by (7) and we settled on the former which yields much fewer function values than the latter. This seems to be because the latter leads to greater deformation of the simplex (i.e., small $\nu$-value), so the search directions are not as well aligned with directions of sufficient descent. As was noted earlier, maintaining the simplex from being too deformed seems to be a key to good performance. The parameter choices for fortified descent in Algorithm 2 are the same as those used in [15]: $\theta_1 = .01$, $\theta_2 = 10^{-5}$, $\sigma(t) = 10^{-5}\min\{.5t^2, t\}$, $\beta(t) = 10^6 t^2$. Also, whenever $S_r$ is too deformed (i.e., $\nu(S_r) < \theta_2$), we replace $\mu_i = 1/n$ with

$$\mu_i = (1-\theta_1)/|I| \ \forall i \in I, \qquad \mu_i = \theta_1/(n-|I|) \ \forall i \in \{1,...,n\}\backslash I,$$

where $I = \{i \in \{1,...,n\} : \min_{j\in\{1,...,n\}}(x_{n+1}-x_i)^T(x_j-x_i) < 0\}$. This backup choice of $\mu_i$ yields an $S_r$ that is less deformed, as it puts more weight on vertices making an obtuse angle with the worst vertex and another vertex.

Table 1 tabulates the performance of Algorithms 1 and 2 on eight test functions with $n = 4, 3, 2, 3, 3, 3, 4, 4$, respectively. The second and fourth test functions are from [11] and [17], and the other test functions are least square problems described in [9] (i.e., numbers 1, 11-14, 16). These functions were chosen because they are widely used and are easy to code. For each test function, the initial simplex was constructed by taking the starting vector used in the above references and adding to this vector the $i$th unit coordinate vectors in $\Re^n$ for $i = 1, ..., n$. Termination occurs when both $d(S)$ and the $\infty$-norm of the vector (14), where $S = \{x_1, ..., x_{n+1}\}$ denotes the current simplex, are below $10^{-3}$. This ensures $\nabla f(x_1) \approx 0$ upon termination and, as can be seen from Table 1, the corresponding $f$-value is within good accuracy of the global minimum. For comparison, we also implemented in Matlab the NM algorithm (as interpreted from the original paper [10] with the recommended parameter setting of $\alpha = 1$, $\gamma = 2$, $\beta = 1/2$) using the same initial simplex and termination criterion, and report its performance on the test functions. As can be seen from Table 1, Algorithms 1 and 2 have about equal performance except on the first and the fifth functions, due to Algorithm 2 preventing the simplex $S$ from being too deformed (i.e., $\nu(S) \geq \theta_2$). While we can always set $\theta_2$ very small so that Algorithm 2 has similar performance as Algorithm 1, the above result suggests that judiciously setting $\theta_2$ to a higher value, possibly chosen dynamically, can be beneficial. Also, we note that the second modification (12) rarely was invoked in Algorithm 2, for otherwise the convergence can be slow. Algorithms 1 and 2 perform better than the NM algorithm on two of the test functions, worse on two other functions, and about equal on the remaining functions. Thus, while Algorithms 1 and 2 have nicer theoretical convergence properties than the NM algorithm, their practical performance seem to be comparable to the latter.

As with the NM algorithm and related simplicial search methods, Algorithm 2 can suffer from poor performance even for moderately large $n$. In particular, the algorithm also exhibited slow convergence on a quadratic example cited by Wright [16, Sec. 7] in which

| | NM algorithm | | Algorithm 1 | | Algorithm 2 | |
|---|---|---|---|---|---|---|
| **Function** | **#$f$-eval.**[1] | **$f$-value**[2] | **#$f$-eval.**[1] | **$f$-value**[2] | **#$f$-eval.**[1] | **$f$-value**[2] |
| Powell1 | 236 | $4.3 \cdot 10^{-6}$ | 182 | $4.1 \cdot 10^{-9}$ | 169 | $9.5 \cdot 10^{-9}$ |
| Powell2 | 95 | $-3.0000$ | 90 | $-3.0000$ | 90 | $-3.0000$ |
| Rosenbrock | 149 | $1.1 \cdot 10^{-7}$ | 182 | $1.6 \cdot 10^{-8}$ | 182 | $1.6 \cdot 10^{-8}$ |
| Zangwill | 86 | $3.1 \cdot 10^{-7}$ | 95 | $2.1 \cdot 10^{-7}$ | 95 | $2.1 \cdot 10^{-7}$ |
| Gulf | 685 | $3.5 \cdot 10^{-11}$ | 440 | $1.1 \cdot 10^{-12}$ | 580 | $3.3 \cdot 10^{-11}$ |
| Box | 252 | $2.9 \cdot 10^{-10}$ | 255 | $1.6 \cdot 10^{-10}$ | 254 | $1.6 \cdot 10^{-10}$ |
| Wood | 584 | $6.3 \cdot 10^{-8}$ | 601 | $6.7 \cdot 10^{-8}$ | 605 | $4.5 \cdot 10^{-8}$ |
| Brown-Dennis | 342 | 85823 | 322 | 85822 | 322 | 85822 |

Table 1: Performance of the NM algorithm and Algorithms 1 and 2 on eight test functions.
[1] This is the number of times that $f$ was evaluated upon termination.
[2] This is the value of $f$ at the best vertex upon termination.

$n = 32$ and $f(x) = \|x\|^2$, and the initial simplex was constructed by taking $(1, 2, ..., 32)^T$ and adding to this vector the $i$th unit coordinate vector in $\Re^{32}$, for $i = 1, ..., 32$.

## References


[1] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, 1995.

[2] J. E. Dennis, Jr. and V. Torczon, *Direct search methods on parallel machines*, SIAM Journal on Optimization, 1 (1991), 448–474.

[3] R. Fletcher, *Practical Methods of Optimization: Volume 1*, John Wiley and Sons, New York, NY, 1980.

[4] A. G. Kuznetsov, *Nonlinear optimization toolbox*, Report OUEL 1936/92, Department of Engineering Science, University of Oxford, Oxford, United Kingdom, 1992.

[5] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the Nelder-Mead simplex algorithm in low dimensions*, Technical Report 96-4-07, Comuting Science Research Center, Bell Laboratories, Murray Hill, NJ, April 1996.

[6] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.

[7] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, NY, 1969.

[8] K. I. M. McKinnon, *Convergence of the Nelder-Mead simplex method to a non-stationary point*, Technical Report 96-006, Department of Mathematics and Computer Science, University of Edinburgh, Edinburgh, May 1996; to appear in SIAM Journal on Optimization.

[9] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, 7 (1981), 17–41.

[10] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Computer Journal, 7 (1965), 308–313.

[11] M. J. D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Computer Journal, 7 (1964), 155–162.

[12] M. J. D. Powell, *Direct search algorithms for optimization calculations*, Acta Numerica, 7 (1998), 287–336.

[13] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimisation and evolutionary operation*, Technometrica, 4 (1962), 441–461.

[14] V. Torczon, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Rice University, Houston, TX, 1989; available as Technical Report 90-7, Department of Mathematical Sciences, Rice University, Houston, TX.

[15] P. Tseng, *Fortified-descent simplicial search method: a general approach*, Department of Mathematics, University of Washington, Seattle, February 1995; revised January 1998.

[16] M. H. Wright, *Direct search methods: once scorned, now respectable*, in Numerical Analysis 1995 (Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis), D. F. Griffiths and G. A. Watson, eds., Addison-Wesley Longman, Harlow, United Kingdom, 1996, 191–208.

[17] W. I. Zangwill, *Minimizing a function without calculating derivatives*, Computer Journal, 10 (1967), 293–296.