Comput Optim Appl
DOI 10.1007/s10589-010-9329-3

Implementing the Nelder-Mead simplex algorithm with adaptive parameters

Fuchang Gao · Lixing Han

Received: 13 January 2010

Abstract In this paper, we first prove that the expansion and contraction steps of the Nelder-Mead simplex algorithm possess a descent property when the objective function is uniformly convex. This property provides some new insights on why the standard Nelder-Mead algorithm becomes inefficient in high dimensions. We then propose an implementation of the Nelder-Mead method in which the expansion, contraction, and shrink parameters depend on the dimension of the optimization problem. Our numerical experiments show that the new implementation outperforms the standard Nelder-Mead method for high dimensional problems.

Keywords Nelder-Mead method · Simplex · Polytope · Adaptive parameter · Optimization

1 Introduction

The Nelder-Mead simplex algorithm [14] is the most widely used direct search method for solving the unconstrained optimization problem

$$\min f(\mathbf{x}), \tag{1.1}$$

F. Gao was supported in part by NSF Grant DMS-0405855.
L. Han was supported in part by a Research and Creative Activities Grant from UM-Flint.

F. Gao
Department of Mathematics, University of Idaho, Moscow, ID 83844, USA
e-mail: fuchang@uidaho.edu

L. Han (✉)
Department of Mathematics, University of Michigan-Flint, Flint, MI 48502, USA
e-mail: lxhan@umflint.edu

Published online: 04 May 2010

where $f : \mathbb{R}^n \to \mathbb{R}$ is called the objective function and n the dimension. A simplex is a geometric figure in n dimensions that is the convex hull of $n + 1$ vertices. We denote a simplex with vertices $\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_{n+1}$ by Δ.

The Nelder-Mead method iteratively generates a sequence of simplices to approximate an optimal point of (1.1). At each iteration, the vertices $\{\mathbf{x}_j\}_{j=1}^{n+1}$ of the simplex are ordered according to the objective function values

$$f(\mathbf{x}_1) \le f(\mathbf{x}_2) \le \cdots \le f(\mathbf{x}_{n+1}). \tag{1.2}$$

We refer to $\mathbf{x}_1$ as the *best* vertex, and to $\mathbf{x}_{n+1}$ as the *worst* vertex. If several vertices have the same objective values, consistent tie-breaking rules such as those given in Lagarias et al. [10] are required for the method to be well-defined.

The algorithm uses four possible operations: *reflection*, *expansion*, *contraction*, and *shrink*, each being associated with a scalar parameter: α (reflection), β (expansion), γ (contraction), and δ (shrink). The values of these parameters satisfy $\alpha > 0$, $\beta > 1$, $0 < \gamma < 1$, and $0 < \delta < 1$. In the standard implementation of the Nelder-Mead method (see for example, [5, 7, 10, 11, 14, 18, 19]), the parameters are chosen to be

$$\{\alpha, \beta, \gamma, \delta\} = \{1, 2, 1/2, 1/2\}. \tag{1.3}$$

Let $\bar{\mathbf{x}}$ be the centroid of the n best vertices. Then

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i. \tag{1.4}$$

We now outline the Nelder-Mead method, which is the version given in Lagarias et al. [10].

One iteration of the Nelder-Mead algorithm

1. **Sort.** *Evaluate f at the $n+1$ vertices of Δ and sort the vertices so that (1.2) holds.*
2. **Reflection.** *Compute the reflection point $\mathbf{x}_r$ from*

$$\mathbf{x}_r = \bar{\mathbf{x}} + \alpha(\bar{\mathbf{x}} - \mathbf{x}_{n+1}).$$

   Evaluate $f_r = f(\mathbf{x}_r)$. If $f_1 \le f_r < f_n$, replace $\mathbf{x}_{n+1}$ with $\mathbf{x}_r$.
3. **Expansion.** *If $f_r < f_1$ then compute the expansion point $\mathbf{x}_e$ from*

$$\mathbf{x}_e = \bar{\mathbf{x}} + \beta(\mathbf{x}_r - \bar{\mathbf{x}})$$

   and evaluate $f_e = f(\mathbf{x}_e)$. If $f_e < f_r$, replace $\mathbf{x}_{n+1}$ with $\mathbf{x}_e$; otherwise replace $\mathbf{x}_{n+1}$ with $\mathbf{x}_r$.
4. **Outside Contraction.** *If $f_n \le f_r < f_{n+1}$, compute the outside contraction point*

$$\mathbf{x}_{oc} = \bar{\mathbf{x}} + \gamma(\mathbf{x}_r - \bar{\mathbf{x}})$$

   and evaluate $f_{oc} = f(\mathbf{x}_{oc})$. If $f_{oc} \le f_r$, replace $\mathbf{x}_{n+1}$ with $\mathbf{x}_{oc}$; otherwise go to step 6.

5. **Inside Contraction.** *If* $f_r \geq f_{n+1}$, *compute the inside contraction point* x_{ic} *from*

$$\mathbf{x}_{ic} = \bar{\mathbf{x}} - \gamma(\mathbf{x}_r - \bar{\mathbf{x}})$$

and evaluate $f_{ic} = f(\mathbf{x}_{ic})$. *If* $f_{ic} < f_{n+1}$, *replace* $\mathbf{x}_{n+1}$ *with* $\mathbf{x}_{ic}$; *otherwise, go to step* 6.

6. **Shrink.** *For* $2 \leq i \leq n+1$, *define*

$$\mathbf{x}_i = \mathbf{x}_1 + \delta(\mathbf{x}_i - \mathbf{x}_1).$$

The Nelder-Mead method may fail to converge to a critical point of f. In [12], Mckinnon constructs three problems in dimension two, where the objective functions are strictly convex and the Nelder-Mead method can converge to a non-critical point of f. One of these objective functions has continuous second derivatives. Aimed at having better convergence, several variants of the simplex method have been proposed (see for example, [2, 4, 8, 15–17]).

Although lacking a satisfactory convergence theory, the Nelder-Mead method generally performs well for solving small dimensional real life problems and continuously remains as one of the most popular direct search methods [9, 10, 19, 20]. It has been observed by many researchers, however, that the Nelder-Mead method can become very inefficient for large dimensional problems (see, for example, [2, 16, 19]). This is the so-called effect of dimensionality [19]. Through numerical experiments, Torczon [16] suggests that this may be due to the search direction becomes increasingly orthogonal to the steepest descent direction. In [6], Han and Neumann study this problem by considering to apply the standard Nelder-Mead method to the quadratic function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ and use the minimizer as one of the initial vertices. They show that the oriented lengths of simplices converge to zero with an asymptotic linear rate and the rate constant rapidly grows to 1 as the dimension increases. It is not known if this type of analysis can be extended to a general initial simplex. The effect of dimensionality deserves further investigation [20].

In this paper, we first prove that the expansion and contraction steps of the Nelder-Mead simplex algorithm possess a descent property when the objective function is uniformly convex. This property offers some new insights on why the standard Nelder-Mead algorithm becomes inefficient in high dimensions, complementing the existing explanations given in Torczon [16] and in Han and Neumann [6]. We then propose an implementation of the Nelder-Mead method in which the expansion, contraction, and shrink parameters depend on the dimension of the optimization problem. We also provide some numerical results which show that the new implementation outperforms the standard Nelder-Mead method for high dimensional problems.

2 A sufficient descent property of the expansion and contraction steps

Following [21], a function on $\mathbb{R}^n$ is called uniformly convex if there exists a strictly increasing function $\rho : [0, \infty) \to [0, \infty)$ such that $\rho(0) = 0$, and for any $x, y \in \mathbb{R}^n$ and any $0 < t < 1$,

$$f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}) - t(1-t)\rho(\|\mathbf{x} - \mathbf{y}\|), \tag{2.1}$$

where $\|\mathbf{x}-\mathbf{y}\|$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$. We remark that in [1], uniformly convex functions are defined using a special choice of ρ: $\rho(t)=\frac{c}{2}t^2$ for some constant $c>0$. It is proved in [1] that if f is a twice continuously differentiable function, then f is uniformly convex (in the sense of [1]) if and only if its Hessian matrix is uniformly positive definite.

Theorem 2.1 *Suppose $n\geq 2$. Assume that f is uniformly convex and the Nelder-Mead method uses the standard parameters (1.3). Let Δ be a simplex in $\mathbb{R}^n$ with vertices $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_{n+1}\}$ and $D(\Delta)$ its diameter. Define the functional*

$$F(\Delta)=f(\mathbf{x}_1)+f(\mathbf{x}_2)+\cdots+f(\mathbf{x}_{n+1}).$$

If $\mathcal{T}$ is an expansion, inside contraction, or outside contraction in the Nelder-Mead method, then

$$F(\mathcal{T}\Delta)-F(\Delta)\leq-\frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}D(\Delta)\right). \tag{2.2}$$

Proof Let $\bar{\mathbf{x}}$ be the centroid of the n best vertices as defined in (1.4). We denote the face of the simplex with the vertices $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n$ by F_n. By the uniform convexity of f, for any $1\leq i\leq n$, we have

$$\begin{aligned}
f(\bar{\mathbf{x}}) &= f\left(\frac{1}{n}\mathbf{x}_i+\frac{n-1}{n}\cdot\frac{\mathbf{x}_1+\cdots+\mathbf{x}_{i-1}+\mathbf{x}_{i+1}+\cdots+\mathbf{x}_n}{n-1}\right)\\
&\leq \frac{1}{n}f(\mathbf{x}_i)+\frac{n-1}{n}f\left(\frac{\mathbf{x}_1+\cdots+\mathbf{x}_{i-1}+\mathbf{x}_{i+1}+\cdots+\mathbf{x}_n}{n-1}\right)\\
&\quad-\frac{n-1}{n^2}\rho\left(\left\|\mathbf{x}_i-\frac{\mathbf{x}_1+\cdots+\mathbf{x}_{i-1}+\mathbf{x}_{i+1}+\cdots+\mathbf{x}_n}{n-1}\right\|\right)\\
&\leq \frac{f(\mathbf{x}_1)+f(\mathbf{x}_2)+\cdots+f(\mathbf{x}_n)}{n}\\
&\quad-\frac{n-1}{n^2}\rho\left(\left\|\mathbf{x}_i-\frac{\mathbf{x}_1+\cdots+\mathbf{x}_{i-1}+\mathbf{x}_{i+1}+\cdots+\mathbf{x}_n}{n-1}\right\|\right)\\
&= \frac{f(\mathbf{x}_1)+f(\mathbf{x}_2)+\cdots+f(\mathbf{x}_n)}{n}-\frac{(n-1)}{n^2}\rho\left(\frac{n}{n-1}\|\mathbf{x}_i-\bar{\mathbf{x}}\|\right)\\
&\leq f(\mathbf{x}_{n+1})-\frac{(n-1)}{n^2}\rho\left(\|\mathbf{x}_i-\bar{\mathbf{x}}\|\right).
\end{aligned}$$

Because the inequality above holds for all $1\leq i\leq n$, we have

$$f(\bar{\mathbf{x}})\leq f(\mathbf{x}_{n+1})-\frac{(n-1)}{n^2}\rho\left(\max_{1\leq i\leq n}\|\mathbf{x}_i-\bar{\mathbf{x}}\|\right). \tag{2.3}$$

If $\mathcal{T}$ is an expansion of $\mathbf{x}_{n+1}$ through the face F_n, then $\mathcal{T}\Delta$ is the simplex with vertices $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ and $\mathbf{x}_e = 3\bar{\mathbf{x}} - 2\mathbf{x}_{n+1}$. Since $\mathbf{x}_r = 2\bar{\mathbf{x}} - \mathbf{x}_{n+1}$ and expansion is used, we have $f(\mathbf{x}_e) < f(\mathbf{x}_r) < f(\bar{\mathbf{x}})$. Therefore we have

$$\begin{aligned} f(\bar{\mathbf{x}}) &= f\left(\frac{1}{2}\mathbf{x}_{n+1} + \frac{1}{2}\mathbf{x}_r\right) \\ &\le \frac{1}{2}f(\mathbf{x}_{n+1}) + \frac{1}{2}f(\mathbf{x}_r) - \frac{1}{4}\rho(\|\mathbf{x}_{n+1} - \mathbf{x}_r\|) \\ &\le \frac{1}{2}f(\mathbf{x}_{n+1}) + \frac{1}{2}f(\bar{\mathbf{x}}) - \frac{1}{4}\rho(2\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|). \end{aligned} \tag{2.4}$$

Applying (2.3) to the right-hand side of (2.4), we obtain

$$\begin{aligned} f(\bar{\mathbf{x}}) &\le f(\mathbf{x}_{n+1}) - \frac{(n-1)}{2n^2}\rho\left(\max_{1\le i\le n}\|\mathbf{x}_i - \bar{\mathbf{x}}\|\right) - \frac{1}{4}\rho(2\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|) \\ &\le f(\mathbf{x}_{n+1}) - \frac{(n-1)}{2n^2}\left[\rho\left(\max_{1\le i\le n}\|\mathbf{x}_i - \bar{\mathbf{x}}\|\right) + \rho(\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|)\right]. \end{aligned}$$

Because

$$\max\left\{\max_{1\le i\le n}\|\mathbf{x}_i - \bar{\mathbf{x}}\|, \|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|\right\} \ge \frac{1}{2}D(\Delta), \tag{2.5}$$

where $D(\Delta)$ is the diameter of Δ, we obtain

$$f(\bar{\mathbf{x}}) \le f(\mathbf{x}_{n+1}) - \frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}D(\Delta)\right),$$

which implies

$$\begin{aligned} F(\mathcal{T}\Delta) - F(\Delta) = f(\mathbf{x}_e) - f(\mathbf{x}_{n+1}) &\le f(\bar{\mathbf{x}}) - f(\mathbf{x}_{n+1}) \\ &\le -\frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}D(\Delta)\right). \end{aligned}$$

Similarly, if $\mathcal{T}$ is an inside contraction, then

$$\begin{aligned} F(\mathcal{T}\Delta) - F(\Delta) &= f\left(\frac{1}{2}\mathbf{x}_{n+1} + \frac{1}{2}\bar{\mathbf{x}}\right) - f(\mathbf{x}_{n+1}) \\ &\le \frac{1}{2}f(\mathbf{x}_{n+1}) + \frac{1}{2}f(\bar{\mathbf{x}}) - \frac{1}{4}\rho(\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|) - f(\mathbf{x}_{n+1}) \\ &\le -\frac{(n-1)}{2n^2}\rho\left(\max_{1\le i\le n}\|\mathbf{x}_i - \bar{\mathbf{x}}\|\right) - \frac{1}{4}\rho(\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|) \\ &\le -\frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}D(\Delta)\right), \end{aligned} \tag{2.6}$$

where in the last two inequalities we used (2.3) and (2.5).

Finally, if $\mathcal{T}$ is an outside contraction, then

$$\begin{aligned} F(\mathcal{T}\Delta) - F(\Delta) &= f\left(\frac{1}{2}\mathbf{x}_r + \frac{1}{2}\bar{\mathbf{x}}\right) - f(\mathbf{x}_{n+1}) \\ &\le \frac{1}{2}f(\mathbf{x}_r) + \frac{1}{2}f(\bar{\mathbf{x}}) - \frac{1}{4}\rho(\|\mathbf{x}_r - \bar{\mathbf{x}}\|) - f(\mathbf{x}_{n+1}) \\ &\le \frac{1}{2}f(\mathbf{x}_{n+1}) + \frac{1}{2}f(\bar{\mathbf{x}}) - \frac{1}{4}\rho(\|\mathbf{x}_{n+1} - \bar{\mathbf{x}}\|) - f(\mathbf{x}_{n+1}) \\ &\le -\frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}D(\Delta)\right), \end{aligned}$$

where the last inequality follows from (2.6). □

Corollary 2.2 *Suppose that $n \ge 2$. If the objective function f is uniformly convex and the standard Nelder-Mead method uses infinitely many expansion or contraction steps, then the diameter of the simplices converges to* 0.

Proof Let $\{\Delta_m\}$ be the sequence of simplices generated by the Nelder-Mead algorithm. We consider the decreasing sequence $F(\Delta_m)$. By the assumption, we can find an infinite subset $\{m_k\}$ such that the operator from Δ_{m_k} to Δ_{m_k+1} is not a reflection. If the diameter of Δ_n does not converge to 0, then there is $\epsilon_1 > 0$ and an infinite subset $K \subset \{m_k\}_{k=1}^{\infty}$, such that for each $k \in K$, $D(\Delta_k) > \epsilon_1$ and the operator from Δ_k to Δ_{k+1} is neither a reflection, nor a shrink. By Theorem 2.1, for such a k, we have

$$F(\Delta_k) - F(\Delta_{k+1}) \ge \frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}\epsilon_1\right).$$

This implies that

$$\begin{aligned} F(\Delta_1) - f_{min} &\ge \sum_{k=1}^{\infty}[F(\Delta_k) - F(\Delta_{k+1})] \\ &\ge \sum_{k\in K}[F(\Delta_k) - F(\Delta_{k+1})] \\ &\ge \sum_{k\in K}\frac{(n-1)}{2n^2}\rho\left(\frac{1}{2}\epsilon_1\right) \\ &= \infty. \end{aligned}$$

This is impossible. Therefore, the diameter of Δ_m converges to 0. □

In [10], Lagarias et al. prove that if f is strictly convex and $n = 2$, the diameter of simplices generated by the standard Nelder-Mead method converges to 0. It is certainly desirable to extend this result to $n \ge 3$, even under the assumption that f is uniformly convex. From Corollary 2.2, we only need to show that starting from any nondegenerate simplex, the Nelder-Mead method cannot always use the reflection

steps. This is true when $n = 2$. However, the situation becomes rather complicated for $n \geq 3$. We will illustrate this in the following example.

Example 2.3 Let $n = 3$. We shall show that for any integer $m \geq 2$, even if we start with a regular simplex, there always exists a uniformly convex function such that the first m iterations of the Nelder-Mead algorithm are all reflections.

To construct such a function, we let $\theta = \cos^{-1}(1/3)$. Because θ/π is irrational, the points $\pm(\frac{\sqrt{3}}{2}\cos k\theta, \frac{\sqrt{3}}{2}\sin k\theta)$, $0 \leq k \leq m$, are distinct points on the circle with center $(0, 0)$ and radius $\frac{\sqrt{3}}{2}$. Let δ be the minimal arc length between these points, and denote

$$r_k = 1 + \frac{k}{m}[\sec\delta - 1].$$

Then the points $\pm(\frac{\sqrt{3}}{2}r_k\cos k\theta, \frac{\sqrt{3}}{2}r_k\sin k\theta)$, $0 \leq k \leq m$, are the vertices of a symmetric convex polygon. To see this, we relabel these points and use polar coordinates to express them as $Q_j = (\rho_j, \theta_j)$, $1 \leq j \leq 2m+2$, where $0 \leq \theta_1 < \theta_2 < \cdots < \theta_{2m+2} < 2\pi$, and $\rho_j > 0$. Because the set $\{Q_1, Q_2, \ldots, Q_{2m+2}\}$ is symmetric about the origin O, the polygon $Q_1Q_2\cdots Q_{2m+2}Q_1$ is symmetric about the origin O. To show it is convex, we prove that for any adjacent points Q_i and Q_j on the polygon, $\angle OQ_iQ_j$ and $\angle OQ_jQ_i$ are both acute. Indeed, because $\angle Q_iOQ_j \geq 2\delta/\sqrt{3} > \delta$, while $|OQ_i|/|OQ_j| \leq r_m/r_0 = \sec\delta$ and $|OQ_j|/|OQ_i| \leq r_m/r_0 = \sec\delta$, $\angle OQ_iQ_j$ and $\angle OQ_jQ_i$ are both acute angles.

Now, we choose $R \geq 2r_m$. At each vertex Q_i, $1 \leq i \leq 2m+2$, we can find a closed disk D_i of radius R such that D_i contains the polygon, and $Q_i \in \partial D_i$, where ∂K denotes the boundary of region K. Let $S = \bigcap_{i=1}^{2m+2} D_i$, and $D = S \cap (-S)$. Being an intersection of closed disks, D is convex. Furthermore ∂D contains all the vertices $Q_1, \ldots, Q_{2m+2}$.

Now, define the function $f : \mathbb{R}^2 \to \mathbb{R}$ as the square of the norm introduced by D, that is, $f(x, y) = r^2$ where $r = \min\{s : (x, y) \in sD\}$. For this function f, we have that: (i) It is uniformly convex; (ii) on ∂D, $f(x, y) = 1$; (iii) $f(0, 0) = 0$; and (iv) because $(\frac{\sqrt{3}}{2}\cos k\theta, \frac{\sqrt{3}}{2}\sin k\theta) \in \partial r_k^{-1}D$, we have $f(\frac{\sqrt{3}}{2}\cos k\theta, \frac{\sqrt{3}}{2}\sin k\theta) = r_k^{-2}$, which decreases as k increases. In particular, we have

$$\cos^2\delta = r_m^{-2} \leq f\left(\frac{\sqrt{3}}{2}\cos k\theta, \frac{\sqrt{3}}{2}\sin k\theta\right) \leq r_1^{-2} < 1$$

for $k = 1, 2, \ldots, m$. Note that when $m \geq 2$, δ is at most $\sqrt{3}\pi/6$. Thus, we have $\cos^2\delta \geq 0.37 > 1/4$, and hence $1/4 < f(\frac{\sqrt{3}}{2}\cos k\theta, \frac{\sqrt{3}}{2}\sin k\theta) < 1$ for $k = 1, 2, \ldots, m$.

Now, we define function $g(x, y, z) = f(x, y) + \frac{1}{2}z^2$. If we choose the initial simplex Δ_0 as the regular simplex with vertices $\mathbf{x}_1 = (0, 0, 1/2)$, $\mathbf{x}_2 = (0, 0, -1/2)$, $\mathbf{x}_3 = (\sqrt{3}/6, \sqrt{2}/\sqrt{3}, 0) = (\frac{\sqrt{3}}{2}\cos\theta, \frac{\sqrt{3}}{2}\sin\theta, 0)$, and $\mathbf{x}_4 = (\sqrt{3}/2, 0, 0)$, then, we have $g(\mathbf{x}_1) = g(\mathbf{x}_2) = 1/8$, $1/4 < g(\mathbf{x}_3) < 1$ and $g(\mathbf{x}_4) = 1$. It is easy to check that when applying Nelder-Mead algorithm to g with initial simplex Δ_0, the first m iterations are all reflections. Indeed, the k-th simplex has vertices

$\mathbf{x}_1^{(k)} = (0, 0, 1/2)$, $\mathbf{x}_2^{(k)} = (0, 0, -1/2)$, $\mathbf{x}_3^{(k)} = (\frac{\sqrt{3}}{2}\cos k\theta, \frac{\sqrt{3}}{2}\sin k\theta, 0)$, and $\mathbf{x}_4^{(k)} = (\frac{\sqrt{3}}{2}\cos(k-1)\theta, \frac{\sqrt{3}}{2}\sin(k-1)\theta, 0)$.

Despite the above example, we conjecture that for a continuous convex objective function, if the level set $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq f(\mathbf{x}_w)\}$ is bounded, where $\mathbf{x}_w$ is the worst vertex of the initial simplex, then the number of consecutive reflections is always finite. In what follows we show that it is true if the reflections follow a repeated pattern—a concept that will become clear as we proceed. In fact, we only need the continuity and the bounded level set of the objective function, not the convexity in this case.

In $\mathbb{R}^n$, let $\mathbf{x}_p^{(0)} = (x_{p1}^{(0)}, x_{p2}^{(0)}, \ldots, x_{pn}^{(0)})$, $1 \leq p \leq n+1$ be the vertices of an n-dimensional simplex. For notational convenience, we do not sort them based on their corresponding function values here. If $\mathbf{x}_k^{(0)}$ is the moving vertex due to a reflection step, then the new simplex has vertices $\mathbf{x}_p^{(1)} = (x_{p1}^{(1)}, x_{p2}^{(1)}, \ldots, x_{pn}^{(1)})$ determined by the equation

$$(x_{pj}^{(1)})_{1\leq p\leq n+1, 1\leq j\leq n} = M_k (x_{pj}^{(0)})_{1\leq p\leq n+1, 1\leq j\leq n}$$

for some k ($1 \leq k \leq n+1$), where the matrix $M_k = (m_{pq})_{1\leq p,q\leq n+1}$ satisfies $m_{kk} = -1$, $m_{pp} = 1$ for $p \neq k$; $m_{kq} = 2/n$ for $q \neq k$, and $m_{pq} = 0$ for all other cases. With this notion, the vertices after b reflections can be expressed as

$$(x_{pq}^{(b)})_{1\leq p\leq n+1, 1\leq q\leq n} = M_{k_b} M_{k_{b-1}} \cdots M_{k_1} (x_{pq}^{(0)})_{1\leq p\leq n+1, 1\leq q\leq n}.$$

Note that the new vertices are all distinct. If we assume that after the first b (where $b \geq 0$ is a finite integer) steps, the M_{k_j}'s appear in the product in a repeated pattern (say repeating in every s steps) as

$$(M_{k_s} M_{k_{s-1}} \cdots M_{k_1})(M_{k_s} M_{k_{s-1}} \cdots M_{k_1}) \cdots (M_{k_s} M_{k_{s-1}} \cdots M_{k_1})$$

then for $l = 1, 2, \ldots$, we can write

$$(x_{pq}^{(ls+b)})_{1\leq p\leq n+1, 1\leq q\leq n} = T^l (x_{pq}^{(b)})_{1\leq p\leq n+1, 1\leq q\leq n},$$

where $T = M_{k_s} M_{k_{s-1}} \cdots M_{k_1}$. Then the Ergodic Theory of linear operators leads to

Claim 2.4 *If f is continuous and its level set $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq f(\mathbf{x}_w)\}$ is bounded, then the moving vertex in step b is a limit point of the new vertices that appear later.*

However, this is impossible for a continuous objective function because the function values at later vertices are strictly decreasing. Therefore, the number of consecutive reflections is always finite in this case.

For the convenience of the readers, we now give a self-contained proof of the claim.

Proof of the Claim We denote by Y_l the $(n+1) \times n$ matrix $(x_{pq}^{(ls+b)})_{1\leq p\leq n+1, 1\leq q\leq n}$. With this notation, we can write $Y_l = T^l Y_0$. Let $\lambda_1, \lambda_2, \ldots, \lambda_{n+1}$ be the eigenvalues of T. By Jordan decomposition, we can write $T = P^{-1} J P$ where P is invertible and

J is the Jordan matrix with $\lambda_1, \ldots, \lambda_{n+1}$ on its diagonal. Note that $T^l = P^{-1} J^l P$. We will prove that J is diagonal and $|\lambda_1| = |\lambda_2| = \cdots = |\lambda_{n+1}| = 1$.

Let $Z_l = PY_l$ for each l. Then $\{Z_l\}_{l=1}^{\infty}$ is bounded due to the boundedness of the level set of f. Moreover, we have $Z_l = J^l Z_0$. If $|\lambda_k| > 1$ for some k, because Z_l is bounded, the k-th row of Z_0 must be zero. Also, because $Z_{l+j} = J^l Z_j$, so the k-th row of Z_j must be zero for all $j = 1, 2, \ldots$. On the other hand, because $|\det(T)| = 1$, the assumption $|\lambda_k| > 1$ implies $|\lambda_j| < 1$ for some $j \neq k$. This implies that the j-th row of Z_l goes to zero as l goes to infinity. Now we end up with the following situation for matrix Z_l: the k-th row is identically zero and the j-th row goes to zero as $l \to \infty$. In particular, this implies that the row vectors of $P^{-1} Z_l$ asymptotically lie on an $n-1$ dimensional hyperplane in $\mathbb{R}^n$. Recall that $Y_l = P^{-1} Z_l$ consist of vertices from a non-degenerated n-dimensional simplex whose volume is a constant. This is impossible. Thus $|\lambda_k| \leq 1$ for all k. Together with the fact that $|\det(T)| = 1$, we conclude that $|\lambda_1| = |\lambda_2| = \cdots = |\lambda_{n+1}| = 1$.

Next, we show that J is diagonal. Suppose J is not diagonal. Let k be the largest integer such that $J_{k-1,k} = 1$, then the boundedness of the $(k-1)$-th row of Z_l would implies the k-th row of Z_0 is identically zero, which further implies that the k-th row of Z_1 is identical zero. Similarly, by analyzing the identity $Z_{l+1} = J^l Z_1$, we see the $(k-1)$-th row of Z_1 is identical zero. Thus, the matrix Z_1 has two rows that are identical zero. But this is impossible because Z_l has rank n.

Now that J is diagonal and $|J_{11}| = |J_{22}| = \cdots = |J_{n+1,n+1}| = 1$, we can write $J^l = \mathrm{diag}(e^{l\alpha_1 i}, e^{l\alpha_2 i}, \ldots, e^{l\alpha_{n+1} i})$, where $i = \sqrt{-1}$. Because $T^l \neq I$ for any integer $l > 0$, at least one of the α_k is an irrational multiple of π. If α_j is a rational multiple of π, then we can find an integer r such that $e^{r\alpha_j i} = 1$; if α_j is an irrational multiple of π, then $\{e^{l\alpha_j i}\}_{l=1}^{\infty}$ is dense on the unit circle. Therefore, it is not difficult to see that we can find a subsequence J^{l_j}, $j = 1, 2, \ldots$, that converges to I. Note that

$$(x_{pq}^{(b)})_{1\leq p\leq n+1, 1\leq q\leq n} = P^{-1} J^{-l_j} P (x_{pq}^{(l_j s+b)})_{1\leq p\leq n+1, 1\leq q\leq n}.$$

In particular, this implies that the moving vertex at step b is a limit point of the vertices appearing in the steps $b + l_j s$. The proof is complete. □

3 Why the Nelder-Mead method becomes inefficient in high dimensions: some new insights

Theorem 2.1 shows that for a uniformly convex objective function, the expansion and contraction steps of the Nelder-Mead simplex algorithm possess a sufficient descent property if the diameter of the simplex is not too small. We will use this property to give some new explanations about the effect of dimensionality on the standard Nelder-Mead algorithm, which complement the existing ones given in Torczon [16] and in Han and Neumann [6]. We assume that f is uniformly convex.

According to Theorem 2.1, the reduction in the functional $F(\Delta)$ caused by an expansion or contraction step depends on the factor $\frac{n-1}{2n^2}$. It decreases for $n \geq 2$ and converges to 0 as $n \to \infty$. This indicates that the efficiency of the expansion and contraction steps diminishes as the dimension n increases.

We also see that the larger the diameter of the simplex Δ is, the more reduction in the functional $F(\Delta)$ an expansion or contraction step can make. This observation can explain why the strategy of restarting with a larger simplex can help improve the performance of the Nelder-Mead method in high dimensions. It also suggests that starting with a large initial simplex may be beneficial if $\mathbf{x}_0$ is far away from a minimizer.

For uniformly convex objective functions, the Nelder-Mead method never uses the shrink step (see for example, [10]). Therefore, Theorem 2.1 indicates that the inefficiency of the standard Nelder-Mead method in high dimensions is likely due to a large number of reflections being used. This is confirmed by our numerical experiment, in which, we used the Matlab implementation of the Nelder-Mead method [11]: `FMINSEARCH`.

`FMINSEARCH` uses the standard choice of the parameters (1.3). It forms the initial simplex by choosing a starting point $\mathbf{x}_0$ as one of the initial simplex vertices. The remaining n vertices are then generated by setting as $\mathbf{x}_0 + \tau_i \mathbf{e}_i$, where $\mathbf{e}_i$ is the unit vector in the ith coordinate, and τ_i is chosen as

$$\tau_i = \begin{cases} 0.05 & \text{if } (\mathbf{x}_0)_i \neq 0, \\ 0.00025 & \text{if } (\mathbf{x}_0)_i = 0. \end{cases} \tag{3.1}$$

For given `TolFun`, `TolX`, `MaxIter`, and `MaxFunEvals` values, `FMINSEARCH` terminates when one of the following three criteria is met:

(T1) $\max_{2\le i\le n+1} |f_i - f_1| \le$ `TolFun` and $\max_{2\le i\le n+1} \|\mathbf{x}_i - \mathbf{x}_1\|_\infty \le$ `TolX`.
(T2) Number of iterations exceeding `MaxIter`.
(T3) Number of function evaluations exceeding `MaxFunEvals`.

In our experiment, we used the following values for termination:

$$\begin{aligned} &\texttt{TolFun} = 10^{-8}, \qquad \texttt{TolX} = 10^{-8}, \qquad \texttt{MaxIter} = 10^6, \\ &\texttt{MaxFunEvals} = 10^6. \end{aligned} \tag{3.2}$$

We ran `FMINSEARCH` on the Quadratic function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ with varying dimensions $n = 2, 4, \ldots, 100$, using the initial point $\mathbf{x}_0 = [1, 1, \ldots, 1]^T \in R^n$. We plotted the fractions of the reflection steps of the Nelder-Mead method on the Quadratic problem as a function of dimension n in Fig. 1. From this figure we can clearly see that the standard Nelder-Mead method uses an overwhelmingly large number of reflections in high dimensions.

One might think that the number of reflections increases because the simplices are close to be degenerate. This is not always the case, as illustrated in Example 2.3, in which the starting simplex is regular. We notice that in the example, the vertices $\mathbf{x}_1$ and $\mathbf{x}_2$ are not moving. In $\mathbb{R}^3$, we can produce similar examples so that $\mathbf{x}_1$ is the only vertex that is not moving in the consecutive reflections. This fact further suggests that when the dimension n gets higher, there are an increasing number of ways to cause consecutive reflections.

So far in this section we have focused on uniformly convex objective functions. We believe that the explanations also shed some light on the behavior of the Nelder-Mead method for general objective functions.

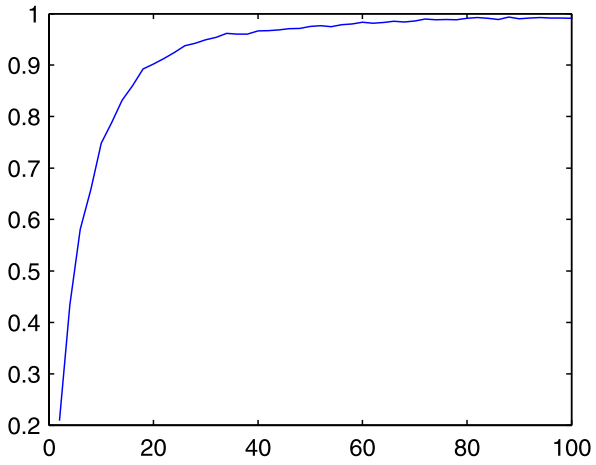

Fig. 1 Fraction of steps which are reflections of standard Nelder Mead plotted against problem dimension for the quadratic $\mathbf{x}^T\mathbf{x}$

4 Implementation using adaptive parameters

4.1 New implementation

From the results in the previous two sections, reducing the chances of using reflection steps and avoiding the rapid reduction in the simplex diameter should help improve the performance of the Nelder-Mead simplex method for large dimensional problems. Based on these observations, we propose to choose the expansion, contraction, and shrinkage parameters adaptively according to the problem dimension n. In particular, we choose for $n \geq 2$,

$$\alpha = 1, \qquad \beta = 1 + \frac{2}{n}, \qquad \gamma = 0.75 - \frac{1}{2n}, \qquad \delta = 1 - \frac{1}{n}. \tag{4.1}$$

We shall call the Nelder-Mead method with this choice of parameters the Adaptive Nelder-Mead Simplex (ANMS) method and the Nelder-Mead method with the standard choice of parameters the Standard Nelder-Mead Simplex (SNMS) method. Note that when $n = 2$, ANMS is identical to SNMS.

Choosing $\beta = 1 + 2/n$ can help prevent the simplex from bad distortion caused by expansion steps in high dimensions. Using $\gamma = 0.75 - \frac{1}{2n}$ instead of 0.5 can alleviate the reduction of the simplex diameter when n is large. The purpose for using $\delta = 1 - 1/n$ instead of $1/2$ is to prevent the simplex diameter from sharp reduction when n is large. This can make the subsequent expansion or contraction steps reduce the objective function more according to Theorem 2.1.

Through numerical experiments, we have observed that using parameters (4.1) instead of (1.3) in the Nelder-Mead algorithm can help reduce the use of reflection steps for uniformly convex functions. We illustrate this in Fig. 2, which shows the fractions of the reflection steps of the ANMS method on the Quadratic problem $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ with varying dimensions $n = 2, 4, \ldots, 100$, using the initial point $\mathbf{x}_0 = [1, 1, \ldots, 1]^T \in R^n$. Comparing it with Fig. 1, we can see the ANMS method significantly reduces the use of reflection steps for this problem. Therefore, the ANMS method is expected to perform well on this problem. The numerical results in the next section confirms this.

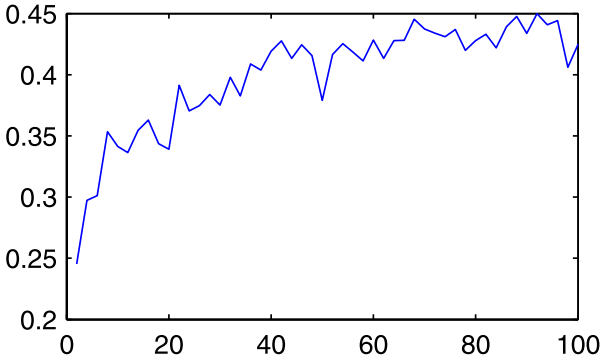

Fig. 2 Fraction of steps which are reflections of adaptive Nelder-Mead plotted against problem dimension for the quadratic $\mathbf{x}^T\mathbf{x}$. Note that the fraction of reflections is at most 0.45 for all dimensions

4.2 Numerical results

In this subsection we report some numerical experiments on the ANMS method. The experiments were done on a Dell XPS 1330 laptop computer with a Windows Vista operating system. We used `FMINSEARCH` with the adaptive parameter values (4.1) to carry out our tests. The initial simplex was the default construction in `FMINSEARCH` (see (3.1)). The following values were used in the termination criteria:

$$\begin{aligned} &\texttt{TolFun} = 10^{-4}, \qquad \texttt{TolX} = 10^{-4}, \qquad \texttt{MaxIter} = 10^6, \\ &\texttt{MaxFunEvals} = 10^6. \end{aligned} \tag{4.2}$$

To assess the performance of the ANMS method on uniformly convex functions, we tested the ANMS method against the SNMS method on the following uniformly convex function, which is a modified version of the quartic function in [3]:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = \mathbf{x}^T D\mathbf{x} + \sigma(\mathbf{x}^T B\mathbf{x})^2, \tag{4.3}$$

where D is a positive definite matrix of the form

$$D = \operatorname{diag}([1+\epsilon, (1+\epsilon)^2, \ldots, (1+\epsilon)^n])$$

and B the positive definite matrix

$$B = U^T U, \quad U = \begin{bmatrix} 1 & \ldots & 1 \\ & \ddots & \vdots \\ & & 1 \end{bmatrix},$$

where $\epsilon \geq 0$ is a parameter that controls the condition number of D and $\sigma \geq 0$ is a parameter that controls the deviation from quadratic. The unique minimizer of problem (4.3) is $\mathbf{x}^* = [0, 0, \ldots, 0]^T$ with minimal value $f(\mathbf{x}^*) = 0$. Note that if $\sigma = 0$, then the objective function f is a quadratic function. In particular, if $\epsilon = \sigma = 0$, then f is the quadratic function $\mathbf{x}^T\mathbf{x}$. In our tests, we used the parameters values $(\epsilon, \sigma) = (0, 0), (0.05, 0), (0, 0.0001)$ and $(0.05, 0.0001)$. The starting point was $[1, 1, \ldots, 1]^T$. We summarize the numerical results in Table 1, in which `Dim` denotes the problem dimension; `Final f` and `nfeval` denote the final function value and

Table 1 Comparison of ANMS and SNMS on problem (4.3)

(ϵ, σ)	Dim	SNMS		ANMS	
		nfeval	Final f	nfeval	Final f
(0, 0)	10	1228	1.4968×10^{-8}	898	5.9143×10^{-9}
(0, 0)	20	12614	1.0429×10^{-7}	2259	1.1343×10^{-8}
(0, 0)	30	38161	7.9366×10^{-7}	4072	1.5503×10^{-8}
(0, 0)	40	75569	2.4515×10^{-4}	7122	1.7631×10^{-8}
(0, 0)	50	106197	6.2658×10^{-4}	9488	2.0894×10^{-8}
(0, 0)	60	114377	6.1295×10^{-5}	13754	3.5012×10^{-8}
(0.05, 0)	10	1123	1.1166×10^{-7}	910	9.0552×10^{-9}
(0.05, 0)	20	9454	2.7389×10^{-7}	2548	1.8433×10^{-8}
(0.05, 0)	30	55603	5.3107×10^{-3}	5067	2.6663×10^{-8}
(0.05, 0)	40	99454	1.5977×10^{-2}	8598	3.6816×10^{-8}
(0.05, 0)	50	215391	1.6906×10^{-1}	13167	6.7157×10^{-8}
(0.05, 0)	60	547475	$1.2685 \times 10^{+1}$	20860	6.8945×10^{-8}
(0, 0.0001)	10	1587	2.0101×10^{-8}	1088	1.4603×10^{-8}
(0, 0.0001)	20	24313	2.2788×10^{0}	4134	2.8482×10^{-8}
(0, 0.0001)	30	43575	$4.5166 \times 10^{+2}$	13148	4.0639×10^{-8}
(0, 0.0001)	40	60153	$3.9387 \times 10^{+2}$	21195	9.3001×10^{-8}
(0, 0.0001)	50	117827	$8.1115 \times 10^{+2}$	42403	8.1755×10^{-8}
(0, 0.0001)	60	195333	$5.4100 \times 10^{+3}$	59626	1.0557×10^{-6}
(0.05, 0.0001)	10	1787	3.1878×10^{-8}	994	6.0454×10^{-9}
(0.05, 0.0001)	20	20824	$1.2984 \times 10^{+1}$	3788	1.5294×10^{-8}
(0.05, 0.0001)	30	39557	$1.8108 \times 10^{+2}$	10251	4.0331×10^{-8}
(0.05, 0.0001)	40	71602	$4.3797 \times 10^{+2}$	18898	5.7407×10^{-8}
(0.05, 0.0001)	50	87660	$8.0726 \times 10^{+2}$	37282	4.7431×10^{-7}
(0.05, 0.0001)	60	136991	$1.5369 \times 10^{+3}$	61259	2.0786×10^{-7}

the number of function evaluations respectively when the algorithm terminates. We observe from this table that the ANMS method is always able to find a good approximation of the solution for this set of problems. On the other hand, the SNMS method terminates prematurely on the two problems when $\sigma = 0.0001$ for $n \geq 20$ after a large number of function evaluations being used. It also fails to find a good approximate solution for the problem with $(\epsilon, \sigma) = (0.05, 0)$ when n is large, after many function evaluations. Overall, the ANMS method substantially outperforms SNMS on the uniformly convex problem (4.3) with these (ϵ, σ) values.

To assess the performance of the ANMS method on general functions, we tested the ANMS method against the SNMS method on the standard test problems from Moré, Garbow, and Hilstrom [13]. First, we did experiments on the 18 Moré-Garbow-Hilstrom unconstrained optimization problems with small dimensions ($2 \leq n \leq 6$), using the standard starting points given in [13]. The numerical results are summarized in Table 2. From this table we see that the performance of the ANMS method is comparable to that of the SNMS method in many cases on this set of small dimensional problems. We note that there are a few cases where ANMS seems to need

Table 2 Comparison of ANMS and SNMS on Moré-Garbow-Hilstrom test problems $2 \leq n \leq 6$

Problem	Dim	SNMS		ANMS	
		nfeval	Final f	nfeval	Final f
Helical valley	3	142	3.5759×10^{-4}	224	2.6665×10^{-4}
Biggs EXP6	6	916	5.6556×10^{-3}	3923	5.5203×10^{-13}
Gaussian	3	62	1.1889×10^{-8}	70	1.2330×10^{-8}
Powell badly scaled	2	700	1.4223×10^{-17}	700	1.4223×10^{-17}
Box 3D	3	480	7.5589×10^{-2}	424	7.5589×10^{-2}
Variably dimensional	4	519	1.1926×10^{-8}	542	5.0684×10^{-9}
Variably dimensional	6	1440	5.3381×10^{-9}	1170	5.9536×10^{-9}
Watson	4	579	5.3381×10^{-2}	730	6.9588×10^{-2}
Watson	6	903	8.3670×10^{-2}	1846	2.2877×10^{-3}
Penalty I	4	583	2.3546×10^{-5}	1436	2.2500×10^{-5}
Penalty I	6	3792	3.8005×10^{-5}	3252	3.8005×10^{-5}
Penalty II	4	2726	9.3805×10^{-6}	197	9.4755×10^{-6}
Brown badly scaled	2	275	2.0036×10^{-9}	275	2.0036×10^{-9}
Brown & Dennis	4	333	$8.5822 \times 10^{+4}$	405	$8.5822 \times 10^{+4}$
Gulf	3	641	9.9281×10^{-7}	2718	9.8888×10^{-19}
Trigonometric	4	203	3.0282×10^{-4}	197	3.0282×10^{-4}
Trigonometric	6	448	2.7415×10^{-4}	437	1.8442×10^{-9}
Extended Rosenbrock	2	159	8.1777×10^{-10}	159	8.1777×10^{-10}
Extended Rosenbrock	4	1345	2.2923×10^{-10}	568	7.3907×10^{-10}
Powell singular	4	305	1.3906×10^{-6}	353	1.7814×10^{-7}
Beale	2	107	1.3926×10^{-10}	107	1.3926×10^{-10}
Wood	4	527	1.9448×10^{-9}	711	9.1293×10^{-9}
Chebyquad	2	57	1.4277×10^{-8}	57	1.4277×10^{-8}
Chebyquad	6	630	1.5008×10^{-7}	414	1.9106×10^{-7}

significantly more function evaluations than SNMS. When we examined them more carefully, we found that: (1) For the Biggs EXP6 function, the two algorithms terminated at different minimizers. (2) For the Watson function ($n = 6$) and Gulf function, the ANMS found a much better approximation than the SNMS method.

We then tested the ANMS method against the SNMS method on the following unconstrained optimization problems of varying dimensions from the Moré-Garbow-Hilstrom collection [13]:

1. `band`: Broyden banded function;
2. `bv`: Discrete boundary value function;
3. `ie`: Discrete integral equation function;
4. `lin`: Linear function—full rank;
5. `pen1`: Penalty I function;
6. `pen2`: Penalty II function;
7. `rosenbrock`: Extended Rosenbrock function;
8. `singular`: Extended Powell singular function;

Table 3 Comparison of ANMS and SNMS on large dimensional problems

Prob	Dim	SNMS		ANMS	
		nfeval	Final f	nfeval	Final f
band	10	1069	2.0581×10^{-6}	741	2.2149×10^{-7}
band	20	5213	8.3121×10^{-5}	1993	5.1925×10^{-7}
band	30	29083	12.1584	3686	6.4423×10^{-7}
band	40	20824	2.4717×10^{-4}	6060	1.0892×10^{-6}
band	50	37802	2.6974×10^{-4}	8357	1.2359×10^{-6}
band	60	38755	2.3999×10^{-4}	10630	1.0002×10^{-6}
bv	10	863	9.5451×10^{-9}	1029	1.0388×10^{-9}
bv	20	5553	7.8216×10^{-6}	7535	3.1789×10^{-10}
bv	30	23150	1.0294×10^{-5}	3860	3.0035×10^{-5}
bv	40	862	1.6788×10^{-5}	1912	1.6110×10^{-5}
bv	50	1087	8.9953×10^{-6}	1905	8.8601×10^{-6}
bv	60	1507	5.3411×10^{-6}	2125	5.3085×10^{-6}
ie	10	1123	5.0253×10^{-9}	774	9.5926×10^{-9}
ie	20	6899	1.2029×10^{-5}	3320	1.0826×10^{-8}
ie	30	43231	0.0015	8711	2.1107×10^{-8}
ie	40	61575	3.7466×10^{-4}	18208	4.0741×10^{-8}
ie	50	155635	0.0031	25961	4.7628×10^{-8}
ie	60	148851	5.1304×10^{-4}	38908	2.2644×10^{-7}
lin	10	1974	1.7816×10^{-8}	1020	5.5242×10^{-9}
lin	20	15401	0.0104	3009	1.1136×10^{-8}
lin	30	57260	0.4494	5310	2.1895×10^{-8}
lin	40	83928	0.5527	8025	1.8607×10^{-8}
lin	50	183633	0.0493	11618	1.9984×10^{-8}
lin	60	475121	0.0280	16492	2.7243×10^{-8}
pen1	10	3909	7.5725×10^{-5}	5410	7.0877×10^{-5}
pen1	20	21680	$8.6799 \times 10^{+3}$	14995	1.5778×10^{-4}
pen1	30	64970	$5.1216 \times 10^{+5}$	45852	2.4773×10^{-4}
pen1	40	254995	$2.8566 \times 10^{+5}$	86293	3.3925×10^{-4}
pen1	50	287599	$4.4971 \times 10^{+6}$	198719	4.3179×10^{-4}
pen1	60	654330	$2.6319 \times 10^{+6}$	254263	5.2504×10^{-4}
pen2	10	4017	2.9787×10^{-4}	9741	2.9366×10^{-4}
pen2	20	27241	0.0065	11840	6.3897×10^{-3}
pen2	30	37774	0.0668	16882	0.0668
pen2	40	116916	29.6170	27211	0.5569
pen2	50	204871	4.2997	43444	4.2961
pen2	60	680176	48.1215	55346	32.2627

9. `trid`: Broyden tridiagonal function;
10. `trig`: Trigonometric function;
11. `vardim`: Variably dimensioned function

using the same starting points as in [13].

Table 4 Comparison of ANMS and SNMS on large dimensional problems

Prob	Dim	SNMS		ANMS	
		nfeval	Final f	nfeval	Final f
rosenbrock	6	2141	2.1314	1833	1.3705×10^{-9}
rosenbrock	12	6125	14.316	10015	3.3974×10^{-9}
rosenbrock	18	13357	22.000	29854	4.2290×10^{-9}
rosenbrock	24	17156	29.119	50338	4.2591×10^{-9}
rosenbrock	30	19678	50.889	156302	5.4425×10^{-9}
rosenbrock	36	43870	52.201	119135	1.6616×10^{-8}
singular	12	2791	9.5230×10^{-6}	5199	3.9417×10^{-8}
singular	24	15187	3.8012×10^{-4}	11156	4.8767×10^{-6}
singular	32	37754	8.4318×10^{-5}	37925	4.6217×10^{-6}
singular	40	80603	0.0039	38530	9.9115×10^{-6}
singular	52	120947	0.0032	73332	1.8319×10^{-5}
singular	60	233482	0.0024	71258	1.9181×10^{-5}
trid	10	908	6.6529×10^{-7}	740	2.5511×10^{-7}
trid	20	3308	2.7137×10^{-6}	3352	2.9158×10^{-7}
trid	30	7610	1.6093×10^{-5}	11343	3.6927×10^{-7}
trid	40	13888	9.8698×10^{-6}	23173	4.4076×10^{-7}
trid	50	24008	1.7782×10^{-5}	42013	5.0978×10^{-7}
trid	60	34853	2.0451×10^{-5}	64369	7.1834×10^{-7}
trig	10	2243	2.7961×10^{-5}	961	2.7952×10^{-5}
trig	20	12519	1.6045×10^{-6}	4194	1.3504×10^{-6}
trig	30	19754	3.5273×10^{-5}	8202	9.9102×10^{-7}
trig	40	23938	1.69412×10^{-5}	17674	1.5598×10^{-6}
trig	50	25328	2.9162×10^{-5}	19426	3.6577×10^{-7}
trig	60	33578	4.8213×10^{-5}	31789	9.6665×10^{-7}
vardim	6	1440	5.3381×10^{-9}	1170	5.9536×10^{-9}
vardim	12	3753	6.6382	4709	8.6227×10^{-9}
vardim	18	6492	8.8146	12815	1.0898×10^{-8}
vardim	24	13844	71.320	35033	1.1237×10^{-8}
vardim	30	19769	85.397	67717	1.5981×10^{-8}
vardim	36	32360	72.101	209340	1.8116×10^{-8}

We summarize the numerical results in Tables 3 and 4. From these tables the ANMS method clearly outperforms the SNMS method when they are used to solve this set of higher dimensional problems. For problems `bv`, `trid`, and `trig`, the SNMS method performs relatively well. We can see that it beats the ANMS method on `bv` and `trid` and loses to the ANMS method on `trig`. For Problems `band`, `ie`, `lin`, `pen1`, `pen2`, and `singular`, the ANMS method substantially outperforms the SNMS method. For each case of Problems `rosenbrock` and for `vardim` with $n \geq 12$, SNMS terminated at a point which is far away from the minimizer while ANMS was able to find a good approximation.

Table 5 SNMS on `rosenbrock` and `vardim` using smaller tolerances

Prob	Dim	nfeval	Final f	TolX=TolFun
rosenbrock	6	6958	1.3047×10^{-15}	10^{-7}
rosenbrock	12	58956	3.1150×10^{-20}	10^{-10}
rosenbrock	18	10^6	19.835	10^{-16}
rosenbrock	24	10^6	22.208	10^{-16}
rosenbrock	30	10^6	39.702	10^{-16}
rosenbrock	36	10^6	51.134	10^{-16}
vardim	12	10^6	4.5329	10^{-16}
vardim	18	10^6	6.7089	10^{-16}
vardim	24	10^6	48.673	10^{-16}
vardim	30	10^6	34.342	10^{-16}
vardim	36	10^6	70.602	10^{-16}

We note that for this set of higher dimensional problems, the ANMS method was always able to obtain a good approximation although it needs a large number of function evaluations in some cases (for example, `pen1, rosenbrock, vardim`). However, the SNMS method can stagnate at a nonminimizer in some situations (for example, `lin, pen1, pen 2`), even after a very large number of function evaluations.

Since it seems that the SNMS method terminated prematurely on Problems `rosenbrock` and `vardim` when the termination parameters (4.2) are used, we did our last experiment testing the SNMS method by using smaller tolerances `TolFun` and `TolX`. The numerical results are reported in Table 5. From this table we observe that the SNMS method can stagnate when it is used to solve these two problems for large dimensions.

5 Final remarks and future research

We have proven that the expansion and contraction steps of the standard Nelder-Mead simplex algorithm possess a descent property when the objective function is uniformly convex. This property offers some new insights on the behavior of the Nelder-Mead algorithm. We have also proposed an adaptive Nelder-Mead simplex (ANMS) method whose expansion, contraction, and shrink parameters depend on the dimension of the optimization problem. Our numerical results have shown that the ANMS method outperforms the standard Nelder-Mead method for high dimensional problems.

We remark that the edges of the initial simplex (3.1) in the Matlab implementation `FMINSEARCH` [11] are quite short. This may be a good strategy for small dimensional problems. For high dimensional problems, we have observed that using a larger initial simplex can improve the performance of the ANMS method. It is therefore desirable to study how to choose a suitable initial simplex. We also remark that

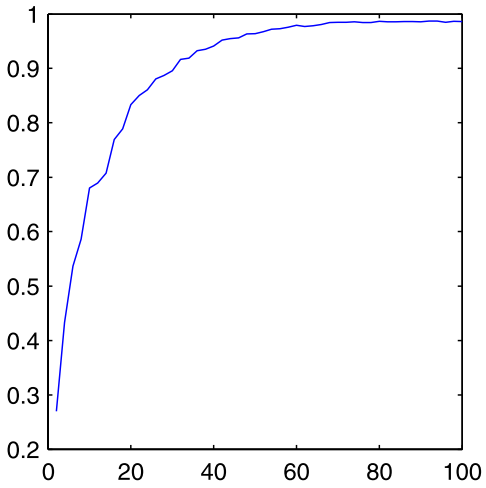

Fig. 3 Fraction of steps which are reflections of adaptive Nelder-Mead plotted against problem dimension for the Extended Rosenbrock function

our choice of parameters in (4.1) is based on the analysis in Sects. 2 and 3 as well as some numerical experiments. Given the significant improvement on the Nelder-Mead method by using these adaptive parameters, it is worth to investigate how to choose most suitable adaptive parameters. We leave both problems for future research.

We have not touched the convergence issues of the ANMS method in this paper. It may suffer similar non-convergence as the SNMS method does. One can remedy this by implementing the ANMS method in one of the convergent variants of the Nelder-Mead method, such as [8, 15, 17].

A natural question is why the ANMS method needs large numbers of function evaluations for Problems `pen1, rosenbrock, vardim` in high dimensions. These functions are certainly difficult for both implementations of the Nelder-Mead method. A more careful examination reveals that for these problems, the ANMS method still uses a large number of reflections in high dimensions. We illustrate this in Fig. 3, which shows the fractions of the reflection steps of the ANMS method on the Extended Rosenbrock function. An interesting future research direction is to develop a variant of the Nelder-Mead simplex method which uses fewer reflections for difficult problems in high dimensions.

Acknowledgement We are very grateful to the referees for their constructive comments and suggestions, which have helped improve the content and presentation of the paper.

References

1. Andrei, N.: Convex functions. Adv. Model. Optim. **9**(2), 257–267 (2007)
2. Byatt, D.: Convergent variants of the Nelder-Mead algorithm. Master's Thesis, University of Canterbury, Christchurch, New Zealand (2000)
3. Byrd, R., Nocedal, J., Zhu, C.: Towards a discrete Newton method with memory for large-scale optimization. In: Di Pillo, G., Giannessi, F. (eds.) Nonlinear Optimization and Applications. Plenum, New York (1996)
4. Dennis, J.E. Jr., Torczon, V.: Direct search methods on parallel machines. SIAM J. Optim. **1**, 448–474 (1991)
5. Dennis, J.E. Jr., Woods, D.J.: Optimization on microcomputers: The Nelder-Mead simplex algorithm. In: Wouk, A. (ed.) New Computing Environments: Microcomputers in Large-Scale Scientific Computing. SIAM, Philadelphia (1987)

6. Han, L., Neumann, M.: Effect of dimensionality on the Nelder-Mead simplex method. Optim. Methods Softw. **21**(1), 1–16 (2006)
7. Kelley, C.T.: Iterative Methods for Optimization. SIAM Publications, Philadelphia (1999)
8. Kelley, C.T.: Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. SIAM J. Optim. **10**, 43–55 (2000)
9. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
10. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.: Convergence properties of the Nelder-Mead simplex algorithm in low dimensions. SIAM J. Optim. **9**, 112–147 (1998)
11. Math Works: MATLAB 6, Release 12, The Math Works, Natick, MA (2000)
12. Mckinnon, K.I.M.: Convergence of the Nelder-Mead simplex method to a nonstationary point. SIAM J. Optim. **9**, 148–158 (1998)
13. Moré, J.J., Garbow, B.S., Hillstrom, B.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**(1), 17–41 (1981)
14. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
15. Price, C.J., Coope, I.D., Byatt, D.: A convergent variant of the Nelder-Mead algorithm. JOTA **113**, 5–19 (2002)
16. Torczon, V.: Multi-directional Search: A Direct Search Algorithm for Parallel Machines. Ph.D. Thesis, Rice University, TX (1989)
17. Tseng, P.: Fortified-descent simplicial search method: a general approach. SIAM J. Optim. **10**, 269–288 (2000)
18. Woods, D.J.: An Iterative Approach for Solving Multi-objective Optimization Problems. Ph.D. Thesis, Rice University, TX (1985)
19. Wright, M.H.: Direct search methods: Once scorned, now respectable. In: Griffiths, D.F., Watson, G.A. (eds.) Numerical Analysis 1995: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, pp. 191–208. Addison Wesley Longman, Harlow (1996)
20. Wright, M.H.: N&M@42: Nelder-Mead at $42''$, a talk given at University of Waterloo, June 2007
21. Zalinescu, C.: On uniformly convex functions. J. Math. Anal. Appl. **95**, 344–374 (1983)