

A direct search method for smooth and nonsmooth unconstrained optimization

C. J. Price¹ M. Reale² B. L. Robertson³

(Received 24 August 2006; revised 4 February 2008)

Abstract

A derivative free frame based method for minimizing C^1 and non-smooth functions is described. A ‘black-box’ function is assumed with gradients being unavailable. The use of frames allows gradient estimates to be formed. At each iteration a ray search is performed either along a direct search quasi-Newton direction, or along the ray through the best frame point. The use of randomly oriented frames and random perturbations is examined, the latter yielding a convergence proof on non-smooth problems. Numerical results on non-smooth problems show that the method is effective, and that, in practice, the random perturbations are more important than randomly orienting the frames. The method is applicable to nonlinear ℓ_1 and ℓ_∞ data fitting problems, and other nonsmooth problems.

See <http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/95> for this article, © Austral. Mathematical Soc. 2008. Published February 27, 2008. ISSN 1446-8735

Contents

| | | |
|----------|---|-------------|
| 1 | Introduction | C928 |
| 2 | Connections with global optimization | C931 |
| 3 | Frames | C935 |
| 4 | The algorithm | C936 |
| 4.1 | The ray search | C936 |
| 4.2 | Frame sizes and stopping conditions | C937 |
| 5 | Convergence | C940 |
| 6 | Numerical results, discussion and conclusion | C941 |
| | References | C945 |

1 Introduction

The unconstrained optimization problem is of the form

$$\min f(x) \quad \text{over } x \in R^n, \tag{1}$$

where a local minimizer is sought. The objective function f maps R^n into $R \cup \{+\infty\}$. For convenience f is given the value $+\infty$ at places where it cannot be evaluated. Allowing f to be infinite means that the method applies to barrier functions, as well as non-smooth penalty functions. The set Ω denotes the set of points at which f is finite, and f is assumed to be continuous on Ω .

Direct search methods enjoyed a resurgence in popularity over the last decade or so [9]. This produced many new methods with convergence results for the case when f is continuously differentiable [14, 8, 11, 3]. Audet

and Dennis [1] extended this to the case when f is Lipschitz, and strictly differentiable at each limit point of the sequence of iterates. When strict differentiability does not hold at some limit x , Audet and Dennis [1] showed that the Clarke derivative [4] at x is non-negative in each direction their method looks in infinitely often as x is approached. Audet and Dennis modify this method to look asymptotically in every direction [2], in which case the Clarke derivative is non-negative in all directions at each limit point. However, non-negativity of the Clarke derivative in all directions does not guarantee the non-existence of descent directions. A simple example is $f = -\|x\|_2$, which has its Clarke derivative at the origin equal to $\|v\|_2$ for every direction v , yet every non-zero direction is a descent direction at $x = 0$. In this article the Clarke derivative approach is replaced by one which directly searches for descent steps using global optimization techniques.

Section 2 shows that obtaining descent steps in nonsmooth optimization is closely related to global optimization. Two approaches are looked at in Sections 3 to 6. The first uses occasional random perturbations, and the second looks in at least one random direction every iteration. The latter approach is in the same spirit as the Mesh Adaptive Direct Search algorithm (or MADS) of Audet and Dennis [2]. Numerical experiments reported in Section 6 show that combining both approaches is the most effective, with the random perturbations being the more important of the two. Section 5 shows convergence when random perturbations are used.

Random perturbations are interspersed with short length local searches using a direct search quasi-Newton method. A direct search quasi-Newton method is a quasi-Newton method which uses finite differences to approximate the gradient, with the following proviso. The stepsize h for the finite differences is chosen in order to get a reasonable rate of convergence, and may be initially quite large (for example, 1 or more); h is *not* chosen in order to obtain accurate gradient estimates. The function evaluations performed for the finite differences also serve as an underlying direct search algorithm which guarantees convergence of the method on C^1 problems. This proof of

convergence [11] is independent of the accuracy of the gradient estimates: it is valid even if the gradient estimates are all set to zero, generated randomly, or are selected as perniciously as possible.

Such a local search has several advantages. The direct search nature of the method means it approximately solves certain types of global optimization problems [17], which is useful here. Moreover, as the local search mimics a quasi-Newton method, it dramatically improves the rate of convergence in areas where f is smooth. The length of each local search is kept to a small number of function evaluations, so that random perturbations are used regularly.

Algorithm 1 generates a sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty} \subset \Omega$. Each iterate $x^{(k+1)}$ is generated from its predecessor $x^{(k)}$ by evaluating f at a finite set of points around $x^{(k)}$, which form part of a frame. The orientation of this finite set of points may be randomly selected, or they may be aligned with the coordinate directions. A direct search quasi-Newton direction is generated, and a finite ray search along this direction is performed. Selected points from this ray search form the remainder of the frame. A second ray search is then performed along the ray from $x^{(k)}$ passing through the lowest point in the frame, yielding the next iterate $x^{(k+1)}$. This second ray search is superfluous when the lowest frame point was generated by the first ray search. This local search process is interrupted regularly, and a number of random perturbations (and their negatives) of the current best known point are considered. The local search continues from the perturbed point with the lowest function value, even if this point is higher than the current iterate. These random (and sometimes uphill) steps allow convergence to be obtained even when f is not smooth.

2 Connections with global optimization

This section shows that finding a descent direction of a non-smooth function in R^n is closely related to a global optimization problem in $n - 1$ dimensions. For clarity $\Omega \equiv R^n$ is assumed. Consider the global optimization problem

$$\min F(\zeta) \quad \text{over} \quad \zeta \in [-1, 1]^{n-1}, \quad (2)$$

where a point with a function value at most $\epsilon > 0$ more than the global minimum value F^* is sought. The global optimization problem is unchanged by the addition of a constant to F , so we subtract $F^* + \epsilon$ from F , which means that a point with a negative function value is now wanted. This problem can be expressed as the problem of finding a direction of descent for a non-smooth function $\Psi(t, z)$ in n dimensions. Ψ is defined in terms of an intermediate function

$$\phi(\zeta) = \begin{cases} F(\zeta), & \|\zeta\|_\infty \leq 1, \\ (\|\zeta\|_\infty - 1)T + (2 - \|\zeta\|_\infty) F(\zeta/\|\zeta\|_\infty), & 1 < \|\zeta\|_\infty \leq 2. \end{cases} \quad (3)$$

The continuous function ϕ extends F to $[-2, 2]^{n-1}$ in such a way that $\phi \equiv F$ on $[-1, 1]^{n-1}$. Outside of this region ϕ rises to a positive value T on $\|\zeta\|_\infty = 2$, where the positive constant T is greater than the *maximum* value of F on $[-1, 1]^{n-1}$. Using $x = (t, z)$ where $z \in R$, define

$$\Psi(t, z) = \begin{cases} \phi(2t/z) \sqrt{z^2 + \|t\|^2}, & \|t\|_\infty < z, \\ T \sqrt{z^2 + \|t\|^2}, & \text{otherwise.} \end{cases}$$

Since Ψ is linear along each ray emanating from the origin, locating a descent direction for Ψ is equivalent to locating a point (t, z) at which Ψ is negative, which implies $\phi(2t/z) < 0$. If $2\|t\|_\infty \leq z$, then $F(2t/z) < 0$, otherwise $F(t/\|t\|_\infty) < 0$ from the form of (3). In both cases a solution to (2) is obtained. As an illustration, $\Psi(t, z)$ is shown in Figure 1 for $F = 16.5 + 20\zeta \sin(20\zeta)$ on $\zeta \in [-1, 1]$.

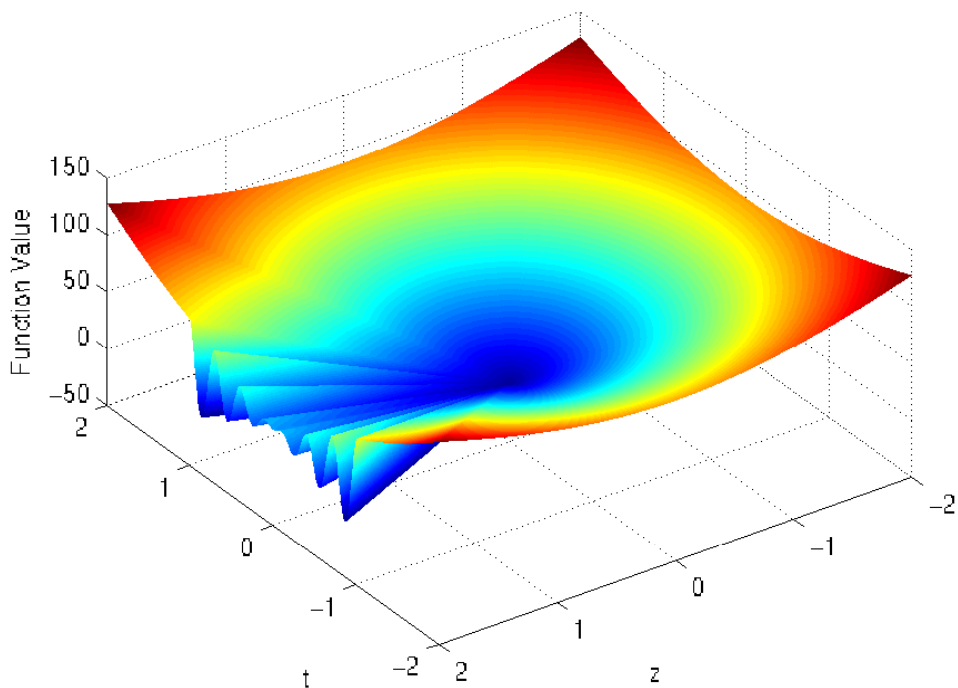


FIGURE 1: $\Psi(t, z)$ for the function $F = 16.5 + 20\zeta \sin(20\zeta)$ on $\zeta \in [-1, 1]$. The connection between finding a descent direction for Ψ at the origin, and finding a global minimizer of F is clear from the front left edge of the graph.

A simple stochastic global optimization method is Pure Random Search (PRS), which evaluates the function at a finite number of randomly selected points in the area of interest and takes the lowest of these as an approximation to a global minimum. PRS can be applied to the $n - 1$ dimensional set of directions emanating from $x^{(k)}$ by using randomly oriented frames. In global optimization literature the multistart method is widely regarded as an improvement of PRS [15]. Multistart generates points randomly, but performs a local search from each random seed point. Applying multistart to f in an n dimensional neighbourhood of $x^{(k)}$ yields a random perturbation method. The following example illustrates what the relative merits of these two approaches might be. Consider finding a descent direction at the origin for the function

$$f = \max\{a^T x, b^T x\}, \quad x \in R^n, \quad (4)$$

where a and b are distinct unit vectors. We assume that $a \neq -b$ so that a descent direction exists. Figure 2 illustrates such a function. Defining the angle θ as shown in Figure 2 via $a^T b = \cos(\pi - \theta)$ gives the angle of the wedge of descent directions in the plane containing a and b . The probability of PRS locating a descent direction by randomly picking a point in any hypersphere centred on the origin is $\theta/2\pi$. If $\theta \approx 0$, then the probability that PRS will succeed in a finite number of iterations can be made arbitrarily low.

In contrast, multistart with steepest descent and exact line searches will succeed *in one iteration* provided the starting point lies to the left of the vectors a and b in Figure 2. This includes all points satisfying $(a + b)^T x < 0$. Hence multistart has at least a 50% chance of success each iteration from a seed point randomly chosen from any hypersphere or box centred on the origin.

Multistart has been developed into more advanced algorithms, such as Multi-Level Single Linkage [12, 13]. The main idea behind these methods is to group the random points into clusters, where a local search from any point in a cluster is expected to find the same local minimum. Hence it is only necessary to do one local search per cluster. Herein a very crude version

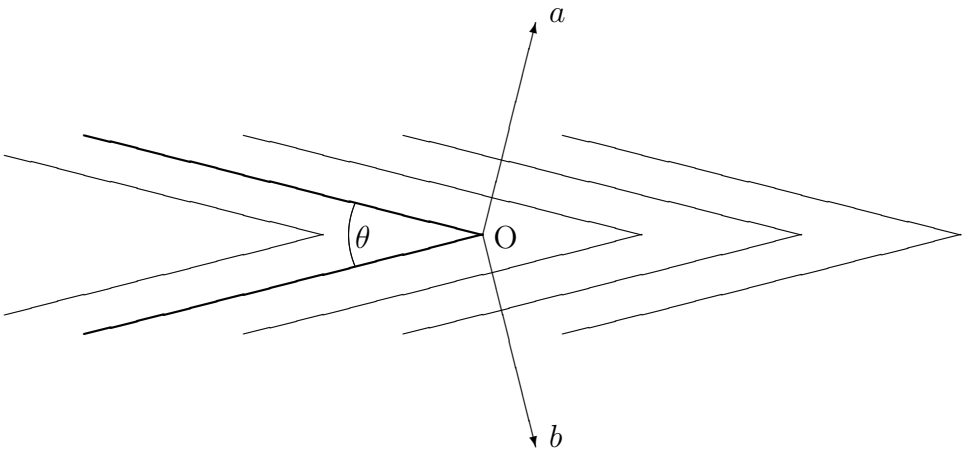


FIGURE 2: Level curves of the nonsmooth function $f = \max\{a^T x, b^T x\}$, with a and b unit vectors satisfying $a \neq -b$. The wedge of internal angle θ between the two thick lines gives the cone of descent directions at the origin O .

of clustering is used: a number of random points are generated and a local search is performed from the lowest of these.

3 Frames

A positive spanning set \mathcal{S}_+ is a finite set of vectors with the property that any vector in R^n can be written as a non-negative linear combination of these vectors. Here non-negative means that the coefficients in the linear combination are all non-negative. If no proper subset of \mathcal{S}_+ is a positive spanning set, then \mathcal{S}_+ is called a positive basis [7]. Positive spanning sets are used to form frames [5, 16]. A frame Φ is constructed around a central point x called the frame's centre:

$$\Phi(x, h, \mathcal{S}_+) = \{x + hv : v \in \mathcal{S}_+\}.$$

The positive quantity h is called the frame size. An upper bound

$$\|v\| \leq K \quad \text{for all } v \in \mathcal{S}_+ \quad (5)$$

is placed on the members of every \mathcal{S}_+ to ensure that the frames shrink to a point as $h \rightarrow 0$. Previously the term frame has only been used when \mathcal{S}_+ is a positive basis. Herein we make an innocuous change and allow \mathcal{S}_+ to be a finite positive spanning set.

Each positive spanning set \mathcal{S}_+ consists of the union of a positive basis and selected points generated by the quasi-Newton ray search. The positive basis $\mathcal{V}_+^{(k)}$ is of the form $\{\pm He_1, \dots, \pm He_n\}$ where H is a Householder matrix and e_1, \dots, e_n are the columns of the identity matrix I . Each Householder matrix is formed using a randomly chosen non-zero vector $u \in [-1, 1]^n$ via $H = I - 2uu^T / \|u\|^2$. These Householder matrices allow the algorithm to look in a random direction every iteration. A second order estimate $g^{(k)}$ of the gradient $\nabla f(x^{(k)})$ is also formed from these points. If some of these points

generated using the positive basis $\mathcal{V}_+^{(k)}$ have infinite function values, forward or backward differences are used to estimate the corresponding element(s) of $g^{(k)}$, if possible, otherwise these element(s) are set to zero.

4 The algorithm

The main part of the algorithm is listed in Algorithm 1. This algorithm calls a local search as a subroutine in step IV. This local search is listed as Algorithm 2. These algorithms, as listed, use both random perturbations and randomly oriented frames. Four different versions of the method were tested, where these variants used either both, one of, or neither of the random perturbations and randomly oriented frames. Random perturbations are eliminated by removing step III of the main algorithm (Algorithm 1) and using $x^{(k)}$ as the start point of the local search in step IV. Randomly oriented frames are eliminated by setting $H^{(k)} = I$ in step 1 of Algorithm 2. This yields fixed frames of the form $\{x \pm he_i : i = 1, \dots, n\}$.

In Algorithm 1, h_{meso} is half the side-length of the hypercube centred on $x^{(k)}$ in which f is polled randomly. Positive upper and lower limits H_{upr} and H_{lwr} are placed on h_{meso} for convergence purposes. The matrix $B^{(k)}$ is a positive definite approximation to the Hessian of f , if it exists.

At each iteration in Algorithm 2 the BFGS update is applied to $B^{(k)}$ yielding $B^{(k+1)}$. If $B^{(k+1)}$ is not positive definite, then $B^{(k+1)}$ is replaced with $B^{(k)}$. The bound on $\alpha\|p\|$ in step 3 is equivalent to that in (5).

4.1 The ray search

Herein we consider a simple descent forward tracking ray search along the ray $x + \alpha hs$ with $\alpha \geq 0$. An increasing sequence of α values $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{j_{\text{max}}}$,

Algorithm 1

- I Initialize: Set $k = 1$. Choose $x^{(1)}$ and $h^{(1)} > 0$. Set $B^{(1)} = I$.
- II Choose $h_{\text{meso}}^{(k)} \in [H_{\text{lwr}}, H_{\text{upr}}]$.
- III Calculate f at $5n/2$ pairs of points x and $2x^{(k)} - x$, where each x is chosen randomly from $x^{(k)} + h_{\text{meso}}^{(k)}[-1, 1]^n$. If f is infinite at all of these points, repeat this step until at least one finite function value is obtained.
- IV Execute the local search with the lowest point found in step III as the initial point. Set $x^{(k)}$ equal to the lowest known point.
- V If the stopping conditions are not satisfied, go to step II.
-

is used, where $\alpha_{j_{\text{max}}} h \|s\| \geq L$ and $L > 0$ is the minimum length over which the ray search must be performed. L is independent of k . Each α_j , $2 < j < j_{\text{max}}$, satisfies $\rho_1 \alpha_{j-1} \leq \alpha_j \leq \rho_2 \alpha_{j-1}$ for constants $1 < \rho_1 \leq \rho_2$ independent of j and k . For convenience $\alpha_0 = 0$ and $\alpha_1 = 1$ are used as these correspond to the points x and $x + hs$ at which the function values are already known. The ray search considers $j = 1, 2, \dots, j_{\text{max}}$ in succession until the inequality $f_j < f_{j-1}$ is violated or the sequence is exhausted. The predecessor of the α_j value which violated this inequality is chosen as $\alpha^{(k)}$. The algorithm may then choose any point as the next iterate provided it is not higher than the point found by the forward tracking ray search. Numerical results were generated with $L = \infty$ and $\rho_1 = \rho_2 = 2$.

4.2 Frame sizes and stopping conditions

Stopping conditions for the main algorithm are based on the number, N_{uls} , of unsuccessful local searches since the last successful local search. A local

Algorithm 2

1. Form $H^{(k)}$.
 2. Calculate f at the points $x^{(k)} + h^{(k)}v$, for all $v \in \mathcal{V}_+^{(k)}$.
 3. Form the direct search quasi-Newton direction p and conduct a forward tracking ray search along $x + \alpha p$. These αp values which satisfy $\alpha \|p\| \leq hK$, plus those in $\mathcal{V}_+^{(k)}$ yield $\mathcal{S}_+^{(k)}$.
 4. Choose $s^{(k)}$ as the member of $\mathcal{S}_+^{(k)}$ which yields the lowest function value.
 5. If $s^{(k)} \in \mathcal{V}_+^{(k)}$ then do a forward tracking ray search along $x^{(k)} + \alpha h^{(k)} s^{(k)}$ with $\alpha \geq 0$, yielding $x^{(k+1)}$.
 6. Adjust h to get $h^{(k+1)}$ and update the Hessian estimate B .
 7. Increment k . If the local search stopping conditions are not satisfied, go to step 1, otherwise return to the main algorithm.
-

search is successful if it improves the current best known point. The current function values of the last 15 iterations (listed in vector form as F_{15}) are also used, together with a positive accuracy parameter τ_{acc} provided by the user. The quantity

$$\Delta F = \frac{\max(F_{15}) - \min(F_{15})}{1 + |\min(F_{15})|}$$

is computed. The algorithm halts when $h_{\text{meso}} < 5\tau_{\text{acc}}$ and any one of the following conditions hold: $\Delta F < \tau_{\text{acc}}/10$; or $\Delta F < \tau_{\text{acc}}$ and $N_{\text{uls}} > 3$; or $\Delta F < 10\tau_{\text{acc}}$ and $N_{\text{uls}} > 7$; or $N_{\text{uls}} > 15$. Numerical results were generated using $\tau_{\text{acc}} = 10^{-5}$.

The local search halts after $10n^2$ function evaluations have been performed; or if f is finite at all points used to estimate the gradient, the gradient estimate g has 2-norm less than $\tau_{\text{acc}}(1 + |f|)$, and $h < 5\tau_{\text{acc}}$. The maximum number of function evaluations is enough for the quasi-Newton local search to minimize a quadratic 4 or 5 times (or to do the equivalent of 4 or 5 iterations of Newton's method, loosely speaking).

In generating the numerical results, h_{meso} was restricted to the interval $[H_{\text{lwr}}, H_{\text{upr}}] = [0.01 \min(\tau_{\text{acc}}, 1), 10]$. When h_{meso} is adjusted, it is increased by a factor 3/2 if the distance between the initial and final points of the most recent local search is at least $\sqrt{n}h_{\text{meso}}/2$ and $\Delta F > 10\tau_{\text{acc}}$. Otherwise it is decreased according to the following schedule: h_{meso} is scaled by 1/2, 0.66, 0.8 or 0.9 depending on whether it is at least 10^5 , 10^2 , 10, or 1 times H_{lwr} .

A lower limit of $h_{\text{min}} = 10^{-10}$ was placed on h , simply to keep h well above machine precision. If the step length in the most recent iteration of the local search was at most $5h$ and the reduction in f in that iteration was at most half the reduction in f in the previous iteration, then h is decreased according to the same schedule as h_{meso} , except that h_{min} replaces H_{lwr} . Otherwise it is increased by a factor 5/2 provided the value of α in the most recent ray search exceeded $2 + 2\sqrt{n}$ and the last local search step length exceeded $20h$.

5 Convergence

When random perturbations are absent the algorithm is provably convergent on continuously differentiable functions [6], with or without randomly oriented frames. Convergence is guaranteed by the ray searches through the best frame point in step 5 of the local search; it is independent of the quasi-Newton ray search in step 3.

When random perturbations are used the following theorem gives convergence under appropriate conditions. For convenience let $B(x, \epsilon) = \{y : \|y - x\|_\infty < \epsilon\}$, and let $\bar{B}(x, \epsilon)$ denote its closure in R^n .

Theorem 1 *Assume the sequence of iterates is bounded, and let x^* be a cluster point of the sequence of iterates such that*

$$f(x^*) = \liminf_{k \rightarrow \infty} f(x^{(k)}).$$

If there exists an $\epsilon > 0$ such that $\Omega \cap \bar{B}(x^, \epsilon)$ is equal to the closure of its own interior, and if f is continuous on $\Omega \cap \bar{B}(x^*, \epsilon)$, then x^* is a local minimizer with probability 1.*

Proof: Assume x^* is not a local minimizer of f . Then there exists a sequence $\{y_i\} \subset \Omega$ converging to x^* such that $f(y_i) < f(x^*)$ for all i . Since $\Omega \cap \bar{B}(x^*, \epsilon)$ is the closure of its own interior, there is a second sequence $\{z_i\}$ in the interior of Ω such that $\{z_i\}$ converges to x^* , and

$$f(z_i) < \frac{1}{2} [f(y_i) + f(x^*)] < f(x^*) \quad \text{for all } i \text{ sufficiently large.}$$

Choose i large enough so that $\|z_i - x^*\|_\infty < H_{\text{lwr}}/3$ and also choose $\eta \in (0, H_{\text{lwr}}/3]$ so that $f(x) < f(x^*)$ on $B(z_i, \eta)$. Hence for any k such that $\|x^{(k)} - x^*\|_\infty < H_{\text{lwr}}/3$ the probability that a random point generated by step III

lies in $B(z_i, \eta)$ is at least $(\eta/H_{\text{upr}})^n$. Since there is an infinite subsequence of $\{x^{(k)}\}$ converging to x^* it follows that the probability that x^* is not a local minimizer is 0, as required. ♠

This result is stronger than corresponding results for MADS in two ways. First, f need only be continuous on Ω near x^* rather than locally Lipschitz. Second, x^* is a local minimizer with probability 1, rather than the Clarke derivative being non-negative at x^* in all directions. Convergence is completely independent of the local search, but the local search can dramatically improve the rate of convergence, especially in regions where f is continuously differentiable, or smoother. Deleting the local search changes the method's nature from that of multistart to pure random search.

6 Numerical results, discussion and conclusion

The algorithm was tested on test functions from Moré, Garbow, and Hillstom [10]. Each of these functions takes a sum of squares form. Herein they are modified by re-writing them as a sum of absolute values

$$f(x) = \sum_{i=1}^m |f_i(x)|.$$

Only functions with an optimal function value of zero are considered because this ensures the sum of squares and sum of absolute values forms share the same optimal point.

The algorithm was tested in four different configurations: with both random perturbations and randomly oriented frames (Table 1); with random perturbations and fixed frames (Table 2); with no perturbations and ran-

TABLE 1: Results for the algorithm with both random perturbations and randomly oriented frames. The number of fails column has been omitted from this table as no fails occurred.

| Problem | n | m | nr f evals | median $\sum f_i $ | median $\sum f_i^2$ |
|----------------------|-----|-----|--------------|---------------------|---------------------|
| Rosenbrock | 2 | 2 | 4338 | 5e-8 | 5e-15 |
| Brown badly scaled | 2 | 3 | 10598 | 2e-3 | 3e-6 |
| Beale | 2 | 3 | 3638 | 4e-8 | 5e-16 |
| Helical valley | 3 | 3 | 8406 | 7e-8 | 3e-15 |
| Gulf | 3 | 99 | 15583 | 1e-5 | 3e-12 |
| Extended Powell | 4 | 4 | 11074 | 4e-7 | 1e-13 |
| Woods | 4 | 6 | 15610 | 3e-7 | 3e-14 |
| Trigonometric | 5 | 5 | 14209 | 5e-8 | 9e-16 |
| Variably Dimensioned | 8 | 10 | 34679 | 2e-7 | 2e-14 |

TABLE 2: Results with random perturbations using frames aligned with the coordinate axes.

| Problem | n | m | nr f evals | median $\sum f_i^2$ | nr fails |
|----------------------|-----|-----|--------------|---------------------|----------|
| Rosenbrock | 2 | 2 | 4442 | 8e-15 | - |
| Brown badly scaled | 2 | 3 | 9606 | 8e-6 | 2 |
| Beale | 2 | 3 | 3038 | 3e-16 | - |
| Helical valley | 3 | 3 | 6595 | 4e-15 | - |
| Gulf | 3 | 99 | 9159 | 7e-16 | 1 |
| Extended Powell | 4 | 4 | 8381 | 2e-11 | - |
| Woods | 4 | 6 | 15451 | 4e-15 | - |
| Trigonometric | 5 | 5 | 14367 | 8e-17 | - |
| Variably Dimensioned | 8 | 10 | 42545 | 3e-15 | - |

TABLE 3: Results with randomly oriented frames, but without random perturbations. Here n/a stands for not applicable.

| Problem | n | m | nr f evals | median $\sum f_i^2$ | nr fails |
|----------------------|-----|-----|--------------|---------------------|----------|
| Rosenbrock | 2 | 2 | n/a | n/a | 5 |
| Brown badly scaled | 2 | 3 | n/a | n/a | 5 |
| Beale | 2 | 3 | 2561 | 3e-17 | - |
| Helical valley | 3 | 3 | 9265 | 2e-16 | 4 |
| Gulf | 3 | 99 | 14607 | 7e-12 | 2 |
| Extended Powell | 4 | 4 | 7173 | 7e-9 | - |
| Woods | 4 | 6 | n/a | n/a | 5 |
| Trigonometric | 5 | 5 | 10515 | 7e-7 | - |
| Variably Dimensioned | 8 | 10 | 11665 | 3e-21 | - |

TABLE 4: Results with frames aligned with the coordinate axes, and without random perturbations.

| Problem | n | m | nr f evals | final $\sum f_i^2$ |
|----------------------|-----|-----|--------------|--------------------|
| Rosenbrock | 2 | 2 | failed | 2.1 |
| Brown badly scaled | 2 | 3 | 642 | 0 |
| Beale | 2 | 3 | 2734 | 9e-14 |
| Helical valley | 3 | 3 | 8254 | 9e-14 |
| Gulf | 3 | 99 | 5928 | 7e-18 |
| Extended Powell | 4 | 4 | 4755 | 3e-12 |
| Woods | 4 | 6 | failed | 7.9 |
| Trigonometric | 5 | 5 | failed | 0.002 |
| Variably Dimensioned | 8 | 10 | 7722 | 2e-22 |

domly oriented frames (Table 3); and with fixed frames without perturbations (Table 4). The first three forms of the algorithm use random elements, and so five runs were performed for each problem. The fixed frame method is deterministic, and so only one run per problem is needed. The legend for Tables 1–4 is as follows: n is the problem dimension; m is the number of absolute values summed to give f ; ‘nr f evals’ is the average number of function evaluations (including in the local searches) over the runs that were successful; ‘median $\sum f_i^2$ ’ is the median value of $\sum f_i^2$ over the successful runs; and similarly for ‘median $\sum |f_i|$.’

The algorithm succeeded on all runs with both random perturbations and randomly oriented frames. The method had 3 fails from 45 runs when random perturbations were used with fixed frames, whereas using randomly oriented frames without perturbations yielded 21 failures from 45 runs. Interestingly without any random elements the method failed on only 3 runs out of 9; a lower failure rate than with randomly oriented frames. This shows that frame based local searches have a significant capability to solve nonsmooth problems.

In order to make a direct comparison with MADS [2], the method was also used to minimize

$$f = (1 - \exp(-\|x\|^2)) \max\{\|x - c\|^2, \|x + c\|^2\},$$

where $c = (30, 40)$ and the initial point is $x = (-2.1, 1.7)$. This problem is non-smooth, but is strictly differentiable at the solution $x = 0$. Various versions of MADS required between 180 and 300 function evaluations to reduce f to 10^{-10} . Five runs of the current method used either two or three iterations of the main algorithm (Algorithm 1) with 105, 106, 122, 194, and 248 function evaluations respectively. This suggests our algorithm is significantly faster than MADS, and that random perturbations interspersed with short local searches may be beneficial to MADS [2].

In conclusion, Section 2 shows that nonsmooth local optimization is closely related to global optimization. A method for nonsmooth optimization

using a frame based local search with occasional random perturbations has been developed, proven convergent, and numerically tested. Use of randomly oriented frames improved the numerical performance of the method, enabling the algorithm to solve each test problem on every run.

Acknowledgement We thank the anonymous referee for many helpful comments, which have improved this article.

References

- [1] C. Audet and J. E. Dennis Jr, Analysis of generalized pattern searches, *SIAM J. Optim.*, **13** (2003), 889–903. doi:10.1137/S1052623400378742 C929
- [2] C. Audet and J. E. Dennis Jr, Mesh adaptive direct search algorithms for constrained optimization, *SIAM J. Optim.*, **17** (2006), 188–217. doi:10.1137/040603371 C929, C944
- [3] A. Burmen, J. Puhán, and T. Tuma, Grid restrained Nelder Mead Algorithm, *Comp. Optim. and Applic.* **34** (2006), 359–375. doi:10.1007/s10589-005-3912-z C928
- [4] F. H. Clarke, Optimization and non-smooth Analysis, *Classics Appl. Math.* **5**, SIAM, Philadelphia, 1990. C929
- [5] I. D. Coope and C. J. Price, Frame based methods for unconstrained optimization, *J. Optimization Theory Applic.*, **107** (2000), 261–274. doi:10.1023/A:1026429319405 C935
- [6] I. D. Coope and C. J. Price, Frame based ray search algorithms in unconstrained optimization, *J. Optimization Theory Applic.*, **116** (2003), 359–377. doi:10.1023/A:1022414105888 C940

- [7] C. Davis, Theory of positive linear dependence, *American Journal of Mathematics*, **76** (1954), 733–746. [C935](#)
- [8] U. M. Garcia-Palomares and J. F. Rodriguez, New sequential and parallel derivative free algorithms for unconstrained minimization, *SIAM J. Optim.*, **13** (2002), 79–96. [doi:10.1137/S1052623400370606](#) [C928](#)
- [9] T. G. Kolda, R. M. Lewis, and V. Torczon, Optimization by Direct Search: New perspectives on some classical and modern methods, *SIAM Review*, **45** (2003), 385–482. [doi:10.1137/S003614450242889](#) [C928](#)
- [10] J. J. Moré, B. S. Garbow, and K. E. Hillstom, Testing unconstrained optimization software, *ACM Trans. Math. Software*, **7:1** (1981), 17–41. [C941](#)
- [11] C. J. Price and I. D. Coope, Frames and grids in unconstrained and linearly constrained optimization: a non-smooth approach, *SIAM J. Optim.*, **14** (2003), 415–438. [doi:10.1137/S1052623402407084](#) [C928](#), [C930](#)
- [12] A. H. G. Rinnooy Kan and G. T. Timmer, Stochastic global optimization methods part I: clustering methods, *Math. Prog.*, **39** (1987), 27–56. [doi:10.1007/BF02592070](#) [C933](#)
- [13] A. H. G. Rinnooy Kan and G. T. Timmer, Stochastic global optimization methods part II: multi-level methods, *Math. Prog.*, **39** (1987), 57–78. [doi:10.1007/BF02592071](#) [C933](#)
- [14] V. Torczon, On the convergence of pattern search algorithms, *SIAM J. Optim.* **7** (1997), 1–25. [doi:10.1137/S1052623493250780](#) [C928](#)
- [15] A. Törn and A. Žilinskas, Global Optimization, *Lecture Notes in Computer Science*, **350**, (1989). [C933](#)

- [16] W.-C. Yu, Positive basis and a class of direct search techniques, *Scientia Sinica (Zhongguo Kexue)*, Special Issue 1 on Mathematics (1979), 53–68. [C935](#)
- [17] S. K. Zavriev, On the global optimization properties of finite difference local descent algorithms, *J. Global Optimization*, **3** (1993), 67–78. [doi:10.1007/BF01100240](https://doi.org/10.1007/BF01100240) [C930](#)

Author addresses

1. **C. J. Price**, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.
2. **M. Reale**, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.
3. **B. L. Robertson**, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand.