ANZIAM J. 59(2017), 215–231
doi:10.1017/S1446181117000323

A DIRECT SEARCH QUASI-NEWTON METHOD FOR NONSMOOTH UNCONSTRAINED OPTIMIZATION

C. J. PRICE[1]

(Received 9 March, 2016; accepted 11 April, 2017; first published online 23 October 2017)

Abstract

A direct search quasi-Newton algorithm is presented for local minimization of Lipschitz continuous black-box functions. The method estimates the gradient via central differences using a maximal frame around each iterate. When nonsmoothness prevents progress, a global direction search is used to locate a descent direction. Almost sure convergence to Clarke stationary point(s) is shown, where convergence is independent of the accuracy of the gradient estimates. Numerical results show that the method is effective in practice.

2010 *Mathematics subject classification*: 65K05.

Keywords and phrases: derivative free, nonconvex, Clarke generalized derivative.

1. Introduction

Many direct search methods for unconstrained optimization [15, 19] originated in the mid-twentieth century, as did the various quasi-Newton methods. Subsequent development saw the latter being preferred on smooth problems due to superior performance [29]. More recently, direct search methods have enjoyed a resurgence, with newer methods, such as the generalized pattern search (GPS) [27] and the mesh adaptive direct search (MADS) [4], having strong convergence theory. These newer classes of direct search methods also include variants which mimic gradient-based methods such as quasi-Newton, while retaining the desirable convergence properties of a modern direct search method [10].

The effectiveness of discrete quasi-Newton techniques on smooth problems justifies their use on nonsmooth problems for two reasons. First, continuous functions can be arbitrarily well approximated by smooth functions, allowing discrete quasi-Newton methods to generate very good steps in some circumstances. Second, a nonsmooth function might have substantial regions where it is smooth, but ill conditioned. Inability to cope with such regions may prevent a method from finding the solution.

[1]Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand; e-mail: C.Price@math.canterbury.ac.nz.

A direct search mimic of the classical Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method is described in this paper. It has the theoretical convergence properties of a direct search method, and a similar numerical performance to a quasi-Newton method on smooth problems. Numerical testing shows that the quasi-Newton nature of the method also substantially improves its performance on nonsmooth problems.

This paper addresses the unconstrained local optimization problem

$$\text{minimize } f(x) \quad x \in \mathbb{R}^n, \tag{1.1}$$

where f is the objective function, and a local minimizer of f is sought. We consider the "black-box" objective functions, where $f(x)$ can be calculated for selected points x, but no other information is available. The convergence analysis requires Lipschitz continuity for f in the region of interest.

Several types of method have been proposed for (1.1), including bundle-based methods [17], generalized gradient-based methods [1], smoothing techniques [20], gradient sampling [8, 26] and direct search methods [3, 5, 27]. Direct search methods are directly applicable to black-box functions as they require only function values. However, associated convergence results typically apply only to a restricted set of problems. Initially, convergence to one or more stationary points was shown [27] under the assumption of continuous differentiability. This was then reduced to strict differentiability at all limit points of interest [3]. Results were also developed for locally Lipschitz [4, 6] and discontinuous [28] functions. The results for locally Lipschitz functions are partial in the sense that convergence to a Clarke stationary point is shown, but such points might still have descent directions [23], unless an additional property such as local convexity also holds.

The GPS [27] minimizes a function by searching over a succession of increasingly fine nested meshes. Each iteration of the GPS uses both a "search" step and a "poll" step. The former is an optional arbitrary process that calculates f at a finite number of points on the current mesh. The poll step is compulsory, and is used to establish the convergence properties of the GPS. The GPS calculates f at a finite number of mesh points forming a frame around the iterate until either a descent step is found or the frame is complete. The frame has the property that every half-space with the iterate on the boundary contains at least one frame point in its interior. This property enables convergence to be shown for the GPS when f is continuously differentiable. However, the nature of the GPS restricts steps from an iterate to a frame point to a finite number of directions over *all iterations*. This prevents extension of the convergence theory to nonsmooth problems.

Audet and Dennis [4] sidestepped this aspect of the GPS by making the meshes become increasingly fine relative to the size of the frames. As the meshes become arbitrarily fine, this allows their algorithm (MADS) to asymptotically look in all directions, and thus allows convergence to Clarke stationary point(s) to be shown. Two variants, LT-MADS and ORTHOMADS [4, 5], have been proposed, employing random and quasi-random methods to choose search directions respectively. This implicitly

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

implements a global search strategy similar to pure random search over the set of directions from the current iterate to nearby mesh points.

The use of meshes in the GPS and the MADS implicitly implements a sufficient descent condition, provided all points that the algorithm looks at lie in a bounded set. This is because there are only a finite number of differences in the function values at two such mesh points. The largest negative difference is effectively the minimum sufficient descent. An alternative approach is to dispense with the meshes and impose a sufficient descent condition directly. The absence of a mesh permits an Armijo backtracking line search [14] to be implemented [21]. It also allows the use of more sophisticated global optimization strategies for locating a descent step from an iterate: herein a modified form of accelerated random search (ARS) is used.

This global direction strategy is combined with the quasi-Newton search as follows. The discrete quasi-Newton method is used while it makes progress. Once progress stalls we switch to the global search to locate a descent direction.

This paper is organized as follows. The next section describes the nonsmooth quasi-Newton (NSQN) method for solving (1.1). Section 3 establishes almost sure convergence and Section 4 numerically tests the NSQN algorithm. Concluding remarks are presented in Section 5.

2. The NSQN algorithm

The algorithm generates a sequence of iterates $\{x_k\}_{k=1}^{\infty}$ with the property that the corresponding sequence of function values $\{f_k\}$ is monotonically decreasing, where the notation $f_k \equiv f(x_k)$ has been used for convenience. At each iterate x_k, the algorithm calculates f at all points forming a frame around x_k. The frame yields a finite difference gradient g_k of f at x_k, which is used to form a quasi-Newton search direction.

Frames are constructed using maximal positive bases [12] of the form

$$\mathcal{V}_+ = \{e_1, e_2, \ldots, e_n, -e_1, -e_2, \ldots, -e_n\},$$

where e_i is the ith unit coordinate vector. A frame Φ is defined in terms of its centre x_k, the positive basis $\mathcal{V}_+$ and a frame size h_k via

$$\Phi_k = \Phi(x_k, h_k, \mathcal{V}_+) = \{x_k + h_k v \mid \text{ for all } v \in \mathcal{V}_+\}.$$

The frame centre x_k is not part of the frame. At each iteration, the function values are calculated at all points in the frame Φ_k around the current iterate. These points allow second-order estimates of the gradient using central differences. These central differences also furnish estimates $\gamma_{11}, \ldots, \gamma_{nn}$, of the nonmixed second partial derivatives of f. In the first iteration, these are used to initialize the Hessian estimate [14] B as the diagonal matrix with diagonal elements $B_{ii} = \max(\gamma_{ii}, 10^{-4})$. At each subsequent iteration, the previous iteration's Hessian estimate B_{k-1} is updated using the BFGS update, yielding B_k. If this update causes a loss of positive definiteness, then it is abandoned and B_k is set equal to B_{k-1}.

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

2.1. The basic strategy of the algorithm At each iteration the method attempts to get a sufficient reduction in f. First, it searches along a quasi-Newton direction. If that does not yield satisfactory descent, then it tries a search along the ray from x_k through the lowest point in the current frame. If that is also unsuccessful, a global search for a descent direction is performed, and a ray search is done along the best direction found. If none of these searches yield sufficient descent, then the frame size h is reduced.

Algorithm 1: The nonsmooth quasi-Newton (NSQN) algorithm.

Step 1 Choose an initial point $x_1 \in \mathbb{R}^n$. Set $k = 1$. Select $h_{\min} \geq 0$ and $h_1 > h_{\min}$. Choose $\tau_{\min}, \tau_{\text{acc}} > 0$.

Step 2 Form the frame $\Phi(x_k, h_k, \mathcal{V}_+)$. Calculate the finite difference gradient g_k. Let $w_k \in \mathcal{V}_+$; minimize $f(x_k + h_k v)$ over $v \in \mathcal{V}_+$.

Step 3 Update B_{k-1}, giving B_k. Form the quasi-Newton direction $p_{\text{qN}} = -B_k^{-1} g_k$.

Step 4 Perform a forward/back tracking ray search along $x_k + \alpha p_{\text{qN}}$, $\alpha > 0$, yielding the point t_k. If $f(t_k) < f_k - \max(\tau_{\min}, \tau_{\text{acc}} h_k)$ go to Step 7.

Step 5 Perform a forward tracking ray search along the ray $x_k + \alpha w_k$, $\alpha > 0$, yielding the point s_k. If $f(s_k) < f_k - \max(\tau_{\min}, \tau_{\text{acc}} h_k)$, go to Step 7.

Step 6 Perform a global search for a descent direction, and do a forward tracking ray search along this direction.

Step 7 Set x_{k+1} equal to the lowest known point. If $f_{k+1} \geq f_k - \tau_{\text{acc}} h_k$ or $\|x_{k+1} - x_k\| < h_k/3$, set $h_{k+1} = \max\{h_{\min}, 4h_k/5\}$. Go to Step 9.

Step 8 If $\alpha_k > 100$ and $\|x_{k+1} - x_k\| > 2h_k$, set $h_{k+1} = 3h_k/2$. Otherwise, set $h_{k+1} = h_k$.

Step 9 If stopping conditions hold, then halt. Otherwise, increment k and go to Step 2.

The quantities $\tau_{\min}$ and τ_{acc} define the least reduction $\max(\tau_{\min}, \tau_{\text{acc}} h_k)$ in f required to avoid having to perform the global direction search in Step 6.

The discrete quasi-Newton direction p_{qN} is calculated as follows. First, B_{k-1} is updated via the BFGS formula, giving B_k. The modified Cholesky factors [14] LDL^T of B_k are then calculated, where D is a diagonal matrix, and L is a lower triangular matrix with $L_{ii} = 1$ for all $i = 1, \ldots, n$. If $\min\{D_{ii} \mid i = 1, \ldots, n\} < 10^{-12}$, the updated matrix B_k is regarded as too close to indefiniteness, and the update is abandoned by setting $B_k = B_{k-1}$ and setting L and D equal to the modified Cholesky factors of B_{k-1}. The equation $LDL^T p_{\text{qN}} = -g_k$ can then be solved for p_{qN} via forward and backward substitution.

A standard forward or forward/back tracking ray search is used along each ray of the form $x_k + \alpha p$, $\alpha > 0$. If $f(x_k + p) < f(x_k)$, the method performs a forward tracking ray search. Otherwise, for a quasi-Newton direction p_{qN} only, it does a backtracking ray search. The forward tracking search tries α values in an increasing geometric sequence $\alpha = 1, \beta, \beta^2, \ldots$ with $\beta > 1$. It halts when the condition

$$f(x_k + \beta^{j+1} p) \geq f(x_k + \beta^j p)$$

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

is satisfied, and assigns $\alpha_k = \beta^j$. The backtracking ray search tries α values in a diminishing geometric sequence $\eta, \eta^2, \ldots$ with $0 < \eta < 1$, and assigns α_k to be the first value η^j to satisfy the sufficient descent condition

$$f(x_k + \eta^j p_{\mathrm{qN}}) < f(x_k) + \rho \eta^j g_k^T p_{qN}.$$

Here ρ is the Goldstein–Armijo parameter [14] with $0 < \rho < 1/2$.

The algorithm reduces h via $h_{k+1} = \max\{4h_k/5, h_{\min}\}$, whenever either

$$\|x_{k+1} - x_k\| < h_k/3 \quad \text{or } f_{k+1} \geq f_k - h_k \tau_{\mathrm{acc}}$$

holds. An immediate consequence of this reduction rule is that either $\{f_k\}$ is unbounded below or $h_k \to h_{\min}$ as $k \to \infty$. In the latter case when $h_{\min}$ is strictly positive, $h_k = h_{\min}$ will be achieved after a finite number of iterations.

Reducing h when the step taken is significantly shorter than h_k helps to keep the step size similar to h. To this end, the method increases h via $h_{k+1} = 3h_k/2$ when both $\alpha_k > 100$ and $\|x_k - x_{k-1}\| \geq 2h_k$.

2.1.1 *Stopping conditions.* The stopping conditions are as follows. Algorithm 1 halts if either:

(1) both $\|g_k\|_2 \leq \tau_{\mathrm{acc}}$ and $h_k \leq \tau_{\mathrm{h}}$; or
(2) $h_k = h_{\min}$ and $f_k \geq f_{k-1} - h_k \tau_{\mathrm{acc}}$.

The first of these is appropriate, when f is continuously differentiable. It says that the estimated gradient g_k is zero to within a tolerance τ_{acc}, and that h_k does not exceed the tolerance τ_{h}. Satisfaction of this latter condition means that it is reasonable to expect the estimated gradient g_k to be accurate. The second set of conditions addresses the case when f is not continuously differentiable.

2.2. The global direction search If a line search along the quasi-Newton direction does not yield sufficient descent, Algorithm 1 attempts to obtain sufficient descent, firstly by a ray search along the most promising frame direction and, if unsuccessful, by using a variant of accelerated random search [2] to locate a descent direction. The accelerated random search (ARS) searches for a global minimizer of a function subject to finite simple bounds on all variables. Herein the ARS is modified to minimize $f(x_k + h_k v)$ over the unit hypersphere $\mathbb{S}_n = \{v \in \mathbb{R}^n \mid \|v\|_2 = 1\}$. The search strategy used by the ARS to minimize a function f over a feasible region Ω is essentially as follows. At each iteration, the ARS has a candidate for the global minimizer called the control point c. The ARS polls randomly in a finite sequence $\{\Omega_j\}_{j=1}^J$ of nested regions, all of which contain the control point. This sequence of regions satisfies $\Omega_1 = \Omega$, and $\Omega_j \subset \Omega_{j-1}$ for all $j > 1$. The ARS works cyclically through the sequence of regions, randomly generating one sample point in each region until a better point than the control point is found. This better point becomes the new control point, and a new sequence of nested regions centred on the new control point is chosen. The ARS then starts polling from the beginning of the new sequence of regions. Almost sure convergence of the ARS follows

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

on noting that at least every Jth iterate is chosen randomly from Ω, where J is the number of regions in the sequence of nested regions.

An alternative view of the ARS is that it samples cyclically from a finite sequence of probability distributions. The first distribution in the sequence is the uniform distribution over Ω, which guarantees convergence almost surely, irrespective of the other distributions. The remaining distributions focus near the control point in the hope of increasing the chance of finding a better point. This view provides enhanced flexibility; many probability distributions do not correspond to random sampling over any region.

In light of this, the ARS is modified, yielding a hyperspherical accelerated random search (HARS), which seeks a global minimizer of a function over the hypersphere $\mathbb{S}_n \subset \mathbb{R}^n$, as follows. Sampling randomly over $\mathbb{S}_n$ can be done by generating a random sample q from the standard n-dimensional normal distribution, which is to say the normal distribution with zero mean and the identity as its covariance matrix. Normalizing q yields a random point on $\mathbb{S}_n$.

Nonuniform sampling uses a scale factor σ to regulate how much sample points are concentrated around the control point c. Initially, $\sigma = 1$ and the sample is drawn randomly from $\mathbb{S}_n$. Each time the sample point fails to improve on c, σ is reduced via $\sigma \leftarrow \sigma/\sqrt{2}$. Nonuniform sampling first chooses $q \in \mathbb{S}_n$ randomly. It then shifts q towards c along the circle defined by the intersection of $\mathbb{S}_n$ with the plane containing the origin, q and c. Let $\theta \in [0, \pi]$ be the angle between the vectors q and c. The sample point w is chosen to lie on this circle, such that the angle between w and c is $\sigma\theta$ and the angle between q and w is $(1 - \sigma)\theta$.

Algorithm 2: The HARS algorithm for locating a descent direction on nonsmooth problems.

Step 1 Randomly pick $c_1 \in \mathbb{S}_n$. Set $m = 1$, $\sigma = 1$ and $\sigma_{\min} = 10^{-8}$. Pick $m_{\max} > 1$.
Step 2 Choose $q \in \mathbb{S}_n$ randomly. Let $\theta \in [0, \pi]$ be the angle between c_m and q.
Step 3 Choose $w \in \text{span}\{c_m, q\} \cap \mathbb{S}_n$, such that the angle between w and c_m is $\sigma\theta$ and the angle between q and w is $(1 - \sigma)\theta$.
Step 4 If $f(x_k + h_k w) < f(x_k + h_k c_m)$, also calculate $f(x_k - h_k w)$.
Step 5 Set c_{m+1} equal to the best known point on $\mathbb{S}_n$.
Step 6 If $c_{m+1} \neq c_m$ or $\sigma < \sigma_{\min}$, set $\sigma = 1$; otherwise set $\sigma = \sigma/\sqrt{2}$.
Step 7 Increment m. If $f(x_k + h_k c_m) \geq f_k - h_k \tau_{\text{acc}}$ and $m < m_{\max}$, go to Step 2.
Step 8 Perform a forward tracking ray search along $x_k + \alpha h_k c_m$, $\alpha \geq 0$ and then exit the algorithm.

The maximum number of function evaluations $m_{\max}$ for the HARS is set at $40n$ when $h_k = h_{\min}$, and at $4n + 20$ otherwise. In the former case, if the HARS does not obtain sufficient descent, then the NSQN halts; this justifies a more thorough global search.

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

3. Convergence

The convergence properties of Algorithm 1 are examined when the stopping conditions are deactivated. This includes setting $h_{\min} = 0$, because Algorithm 1 halts when an unsuccessful global direction search is performed and $h_k = h_{\min}$.

The convergence results use the generalized derivative, which is also known as the Clarke(–Rockafellar) derivative [9]. The Clarke derivative f° of f at x^* in the direction v is given by

$$f^\circ(x^*; v) = \limsup_{y \to x^*, t \downarrow 0} \frac{f(y + tv) - f(y)}{t},$$

where f must be locally Lipschitz [9] at x^* to ensure that the limit supremum is finite.

Assumption 3.1. For each cluster point x^* of $\{x_k\}_{k=1}^\infty$, the objective function f is locally Lipschitz in a neighbourhood of x^*.

Convergence also requires the following assumption that the sequences of iterates and function values do not diverge.

Assumption 3.2. (a) The sequence of iterates $\{x_k\}_{k=1}^\infty$ lies in a compact subset of $\mathbb{R}^n$.
(b) The sequence of function values $\{f(x_k)\}_{k=1}^\infty$ is bounded below.

Assumption 3.2(b) ensures that Steps 5 and 6 of the NSQN method in Algorithm 1 are executed infinitely often. Together, Assumption 3.2(b) and the sufficient descent condition in Step 7 of Algorithm 1 imply that $h_k \to 0$ as $k \to \infty$.

Proposition 3.3. *Steps 5 and 6 of Algorithm 1 are performed infinitely often.*

Proof. At each iteration, Algorithm 1 either reduces f by at least $\tau_{\min}$ or it performs Steps 5 and 6. However, the sequence of function values $\{f_k\}$ is monotonically decreasing, and it is also bounded below by Assumption 3.2(b). Hence the number of times f can be reduced by $\tau_{\min}$ or more is finite. □

The next proposition asserts that if the Clarke derivative $f^\circ(x^*; v^*)$ is negative, then any finite difference quotient at a point and in direction sufficiently near x^* and v^* is also negative. This can be used to show that $f^\circ(x^*; v) \geq 0$ for all nonzero v is a necessary, but not sufficient, condition for x^* to be a local minimizer. A point x^* satisfying $f^\circ(x^*; v) \geq 0$ for all $v \neq 0$ is called a Clarke stationary point. For convenience, the notation $B_\epsilon(x)$ is used to denote the open ball of radius ϵ centred at the point x.

Proposition 3.4. *If Assumption 3.1 holds at x^* and $f^\circ(x^*; v^*) < 0$, then for all $\mu < 1$, there exists $\delta > 0$ such that $y \in B_{2\beta\delta}(x^*)$, $v \in B_\delta(v^*)$ and $h \in (0, \delta)$ imply that*

$$\frac{f(y + hv) - f(y)}{h} < \mu f^\circ(x^*; v^*).$$

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

Proof. Let f have a Lipschitz constant L in a neighbourhood of x^*. Then

$$\frac{f(y+hv)-f(y)}{h} = \frac{f(y+hv)-f(y+hv^*)+f(y+hv^*)-f(y)}{h}$$
$$\leq \frac{L\|h(v-v^*)\|}{h} + \frac{f(y+hv^*)-f(y)}{h}.$$

Examining the right-hand term for sufficiently small $\delta > 0$,

$$\frac{f(y+hv^*)-f(y)}{h} < \frac{1+\mu}{2} f^\circ(x^*; v^*)$$

from the definition of f°. The left-hand term has magnitude less than $L\delta$, so by choosing δ so that $L\delta < (\mu-1)f^\circ(x^*; v^*)/2$,

$$L\|v-v^*\| + \frac{f(y+hv^*)-f(y)}{h} < \frac{\mu-1}{2} f^\circ(x^*; v^*), + \frac{1+\mu}{2} f^\circ(x^*; v^*)$$

which gives the required result. □

Corollary 3.5. *Let $f^\circ(x^*, v^*) < 0$ with $\|v^*\| = 1$. Then, for all sufficiently small positive δ, a forward tracking ray search from $y \in B_\delta(x^*)$ along $v \in B_\delta(v^*)$ with $h < \delta$ will generate a value of α satisfying both $\alpha \geq 1$, and*

$$f(y+\alpha hv) \leq f(y) + \mu \frac{(2\beta-1)\delta}{\beta(\|v^*\|+\delta)} f^\circ(x^*; v^*). \tag{3.1}$$

Proof. Let

$$\alpha_{\max} = \frac{(2\beta-1)\delta}{h(\|v^*\|+\delta)}.$$

First, note that $\alpha_{\max} \geq \beta$ for sufficiently small δ. Now, for $\alpha \in [0, \alpha_{\max})$,

$$\|y - x^* + \alpha hv\| < \delta + \frac{(2\beta-1)\delta}{(\|v^*\|+\delta)}\|v\| \leq 2\beta\delta,$$

which implies that $y + \alpha hv \in B_{2\beta\delta}(x^*)$.

Proposition 3.4 guarantees that the function $f(y+\alpha hv)$ is strictly decreasing for $0 \leq \alpha < \alpha_{\max}$. Hence the value of α chosen by the forward ray search will satisfy $\alpha \geq \alpha_{\max}/\beta$. Hence Proposition 3.4 yields (3.1), as required. □

Theorem 3.6. *Let $(x^*, v^*, 0)$ be a cluster point of the combined sequence $\{(x_k, v_k, h_k)\}_{k\in\mathcal{K}}$, where a forward tracking ray search is performed along the ray $x_k + \alpha v_k$, $\alpha > 0$, for all $k \in \mathcal{K}$. Then $f^\circ(x^*; v^*) \geq 0$.*

Proof. The proof is by contradiction. Assume that $f^\circ(x^*; v^*) < 0$. By replacing $\{(x_k, v_k, h_k)\}$ with a subsequence of itself, if necessary, let $x_k \to x^*$, $v_k \to v^*$ and $h_k \to 0$ as $k \to \infty$.

Corollary 3.5 yields

$$f(x_k + \alpha_k h_k v_k) \leq f^* + |f(x_k) - f(x^*)| + \mu \frac{(2\beta-1)\delta}{\beta(\|v^*\|+\delta)} f^\circ(x^*; v^*) \tag{3.2}$$

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

for all sufficiently large k. The term on the far right in (3.2) is strictly negative and independent of k. Continuity of f implies that $|f(x_k) - f(x^*)| \to 0$ as $k \to \infty$. Together with (3.2), this implies that $f(x_k + \alpha_k h_k v_k) < f^*$ for all sufficiently large k. The sequence of function values $\{f(x_k)\}$ is monotonically decreasing, and so the continuity of f yields the contradiction. □

Theorem 3.6 is applicable to any convergent subsequence $\{(x_k, v_k)\}$ of forward searched rays, including those along quasi-Newton directions. Unfortunately, the quasi-Newton directions are unpredictable, because the finite difference gradients can be arbitrarily inaccurate. Consequently, the convergence properties of Algorithm 1 are developed using the ray searches in Steps 5 and 6 only. These steps are not performed in all iterations, so we must identify the various subsequences corresponding to these iterations. To this end, we define the set $\mathcal{F}$ such that $k \in \mathcal{F}$ if and only if Step 5 of Algorithm 1 is executed in iteration k. Similarly, $k \in \mathcal{G}$ if and only if Step 6 of Algorithm 1 is executed in iteration k. Proposition 3.3 implies that $k \in \mathcal{F}$ and $k \in \mathcal{G}$ for all sufficiently large k.

The main convergence result is along the following lines. It assumes that a cluster point x^* of the sequence of iterates is not a Clarke stationary point. It then shows that for any sufficiently large k, the NSQN method has a nonzero chance of locating a better point than x^*, provided x_k is sufficiently near x^*. These probabilities are bounded away from zero for large k, and the global direction searches in different iterations are independent of one another. Hence it can be shown that the NSQN method will find a better point than x^*, almost surely. Continuity of f means that convergence to a point which is not Clarke stationary is a probability zero event.

COROLLARY 3.7. *Let x^* be a cluster point of the subsequence of iterates $\{x_k\}_{k\in\mathcal{G}}$. Then x^* is a Clarke stationary point, almost surely.*

PROOF. The proof is by contradiction. Let v^* minimize $f^\circ(x^*, v)$ over $v \in \{v \in \mathbb{R}^n : \|v\|_2 = 1\}$. Assume that $f^\circ(x^*; v^*) < 0$. By replacing $\{x_k\}$ and $\{h_k\}$ with subsequences of themselves if necessary, let $x_k \to x^*$ and $h_k \to 0$ as $k \to \infty$. The Clarke derivative $f^\circ(x^*; v)$ is a subadditive and positively homogeneous function of its second argument v [9, Proposition 2.1.1]. Hence, there is a region D of positive measure on the unit hypersphere such that $f^\circ(x^*, v) \le f^\circ(x^*, v^*)/2$ if and only if $v \in D$.

Corollary 3.5 shows that, for sufficiently large k,

$$f(x_k + \alpha_k h_k v_k) - f(x_k) \le \frac{\mu(2\beta - 1)\delta}{\beta(1 + \delta)} f^\circ(x^*, v_k) \le \frac{\mu(2\beta - 1)\delta}{\beta(1 + \delta)} \cdot \frac{f^\circ(x^*, v^*)}{2},$$

when $v_k \in D$ and $\|v^*\| = 1$. The set $\mathcal{G}$ of iterations in which the HARS is performed is infinite by Proposition 3.3. At each iteration indexed by $\mathcal{G}$, the HARS generates at least one random sample on the unit hypersphere. Hence the probability that $v_k \in D$ is at least a for all $k \in \mathcal{G}$; here a is the probability that a random point on the unit hypersphere lies in D. Hence, almost surely we can select $k \in \mathcal{G}$ sufficiently large such that $v_k \in D$, and

$$|f(x_k) - f(x^*)| < \left| \frac{\mu(2\beta - 1)\delta}{\beta(1 + \delta)} \frac{f^\circ(x^*, v^*)}{4} \right|$$

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

by continuity of f. For this value of k, $f(x_{k+1}) \le f(x_k + \alpha_k h_k v_k) < f(x^*)$. Now x^* is a cluster point of $\{x_k\}_{k=1}^{\infty}$. Noting that the sequence $\{f(x_k)\}_{k=1}^{\infty}$ is monotonically decreasing, this contradicts the continuity of f. □

It is possible that the Clarke derivative is nonnegative in all directions, but descent directions still exist (see, for example, [23]). However, when f is strictly differentiable or locally convex at x^*, stronger statements can be made. Strict differentiability [9] of f at x^* means that there exists a $g_* \in \mathbb{R}^n$ such that $f^\circ(x^*; v) = g_*^T v$ for all $v \in \mathbb{R}^n$. When $g_* = 0$, x^* is a stationary point of f. The next result shows that if f is strictly differentiable at a cluster point x^* of $\{x_k\}$, then x^* is a stationary point of f. This result does not require a dense set of search directions, and holds even if Step 6 of Algorithm 1 is omitted.

THEOREM 3.8. *If f is strictly differentiable at a cluster point x^* of the subsequence $\{x_k\}_{k\in\mathcal{F}}$, then x^* is a stationary point.*

PROOF. Let g^* be the gradient of f at x^*. Step 2 of the NSQN method in Algorithm 1 specifies w_k such that $x_k + h_k w_k$ is the best point in the frame Φ_k. Choose $\mathcal{K} \subseteq \mathcal{F}$ such that $(x_k, w_k, h_k) \to (x^*, v^*, 0)$ as $k \to \infty$ with $k \in \mathcal{K}$. Theorem 3.6 implies that $f^\circ(x^*, v^*) \ge 0$. Now $f(x_k + h_k v) \ge f(x_k + h_k w_k)$ for all frame directions $v \in \mathcal{V}_+$, and hence

$$\frac{f(x_k + h_k v) - f(x_k)}{h_k} \ge \frac{f(x_k + h_k w_k) - f(x_k)}{h_k} \quad \text{for all } v \in \mathcal{V}_+.$$

Strict differentiability at x^* implies that the right-hand side converges to $f^\circ(x^*, v^*)$ as $k \to \infty$. Hence $f^\circ(x^*, v) \ge f^\circ(x^*, v^*)$ for all v in $\mathcal{V}_+$. Now $f^\circ(x^*, v^*) \ge 0$, which implies that $v^T g^* \ge 0$ for all v in $\mathcal{V}_+$; hence $g^* = 0$ [10]. □

The final result shows that a Clarke stationary point in an open neighbourhood over which f is convex is a local minimizer of f. A proof of this known result [13] is provided here for convenience.

THEOREM 3.9. *If x^* is a Clarke stationary point of f, and if f is locally convex at x^*, then x^* is a local minimizer of f.*

PROOF. [13]. Let v be an arbitrary unit vector and let f be convex on the open ball $B_{2r}(x^*)$. Then $f^\circ(x^*; v) \ge 0$ implies that there exist sequences $\{y_j\}_{j=1}^{\infty}$ and $\{t_j\}_{j=1}^{\infty}$ converging to x^* and to zero from above such that

$$\frac{f(y_j + t_j v) - f(y_j)}{t_j} \ge \frac{-1}{j}.$$

For fixed $a \in (0, r)$, convexity implies that

$$\frac{f(y_j + a v) - f(y_j)}{a} \ge \frac{-1}{j},$$

provided j is sufficiently large such that $y_j \in B_r(x^*)$ and $t_j \le a$. In the limit as $j \to \infty$, $f(x^* + av) \ge f(x^*)$. Since v and a are arbitrary, x^* must be a nonstrict local minimizer. □

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

TABLE 1. Results of test set A [18] . Here "H" and "no H" refer to results with and without HARS, respectively. On each problem, both versions of Algorithm 1 found the same listed final function value. All results are 30-run averages. † On problem 18 both methods found a stationary point.

Problem	Function	n	f^*	Final f	no. fcn evals	
					H	no H
1	Rosenbrock	2	0	5.7e-19	255	199
2	Freudenstein & Roth	2	48.9842	48.984253	107	79
3	Powell badly scaled	2	0	2.5e-29	1075	906
4	Brown badly scaled	2	0	0	201	145
5	Beale	2	0	3.5e-22	160	104
6	Jennrich & Sampson	2	124.362	124.362	209	128
7	Helical Valley	3	0	8.0e-20	276	212
8	Bard	3	0.00821487	0.00821488	200	168
9	Gaussian	3	1.12793e-8	1.12793e-8	81	49
10	Meyer	3	87.9458	87.945855	5537	4024
11	Gulf Research	3	0	7.6e-18	402	370
12	Box	3	0	1.1e-16	366	334
13	Powell singular	4	0	3.4e-14	450	378
14	Wood's	4	0	1.1e-18	862	790
15	Kowalik & Osborne	4	3.07505e-4	3.07505e-4	223	187
16	Brown & Dennis	4	85822.2	85822.2	362	290
17	Osborne 1	5	5.46489e-5	5.46489e-5	873	753
18	Biggs exp6†	6	0	5.65e-3	388	388
19	Osborne 2	11	0.0401377	0.0401377	875	811
20	Penalty function I	4	2.24997e-5	2.24997e-5	1155	1192
21	Penalty function I	10	7.08765e-5	7.08765e-5	3741	2285
22	Broyden tridiagonal	10	0	1.6e-15	583	463
23	Variably dimensioned	10	0	1.6e-20	1931	1811
24	Trigonometric	5	0	2.5e-14	275	275

These convergence results do not rely on the presence or accuracy of the estimated gradient and Hessian; they remain valid even if the quasi-Newton step is omitted and neither g nor B are constructed.

4. Numerical results

Algorithm 1 was tested on four sets of test problems, and results averaged over 30 runs are presented. Numerical testing used $\tau_{\text{acc}} = 10^{-5}$, $\tau_{\text{h}} = 10^{-3}$, $\tau_{\text{min}} = 10^{-10}$, $\beta = 4$, $\eta = 0.5$, $\rho = 10^{-5}$, $h_1 = 10^{-6}$ and $h_{\text{min}} = 10^{-10}$. Tabulated results use scientific e-notation; for example 2.6e-7 specifies the number 2.6×10^{-7}.

Test set A contains 24 problems from the work of Moré et al. [18]. It is used to show that Algorithm 1 behaves as an effective quasi-Newton method when f is smooth. Results are presented in Table 1. Columns 2, 3 and 4 list the name, dimension

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

and optimal function value of each test problem, respectively. Columns 5 and 6 list the final function value found by Algorithm 1, along with the number of function evaluations needed to find it. Convergence on these smooth problems is guaranteed by Theorem 3.8, with or without the use of the global direction search. Hence, for comparison, column 7 lists the number of function evaluations needed when the global direction search (Step 6 of NSQN) is disabled. Algorithm 1 found the same optimal function value on each problem with or without Step 6. A comparison shows that including Step 6 increases the number of function evaluations taken by about 25% on average.

On problem 18, both versions of Algorithm 1 located a stationary point rather than a minimizer. On all problems except problem 10, Algorithm 1 halted after satisfying the estimated gradient condition $\|g_k\| < \tau_{\text{acc}}$. On problem 10, Algorithm 1 halted by reaching $h_k = h_{\min}$ without being able to make further progress. The solution to problem 10 was located to high accuracy, but $\|g_k\|_2 \approx 100$ in the final iterations. For all problems except problems 3, 16, 20 and 21 $\tau_{\text{acc}} = 10^{-5}$ was used. Problem 20 used $\tau_{\text{acc}} = 10^{-6}$ and problems 3 and 21 used $\tau_{\text{acc}} = 10^{-8}$ in order to prevent Algorithm 1 halting early. Problem 16 has a large optimal function value, and hence $\tau_{\text{acc}} = 10^{-5}$ was found to be overly stringent, with errors in the gradient estimate being of a similar size. Results are presented using $\tau_{\text{acc}} = 10^{-4}$ on this problem.

Eleven of the problems in test set A are solved by Wu and Sun [25], nine of which are also solved in [30]. Algorithm 1 (with HARS) was faster on four of the eleven problems solved by Sun [25], and three of the nine problems solved by Wu and Sun [30]. Algorithm 1 contains two features which increase the number of function evaluations taken on smooth problems: the exclusive use of maximal frames and the global direction search. On smooth functions, the global direction search can be omitted without compromising convergence properties. Additionally, gradients can often be estimated using minimal rather than maximal frames, which can reduce the total function count significantly. Wu and Sun [30] do precisely this. Nevertheless, the HARS and maximal frames are important on nonsmooth problems, and so are permanent features of the NSQN method.

Test set B contains nonsmooth variants of selected problems of Moré et al. [18], all of which appear in previous literature [22–24]. The original version of each problem is a sum of squares $\sum_{i=1}^{m}(f_i(x))^2$ with an optimal function value of zero, giving $f_i(x^*) = 0$ for all i at the solution x^*. Each nonsmooth version is a sum of absolute values $f(x) = \sum_{i=1}^{m} |f_i(x)|$, which has the same solution x^* with $f(x^*) = 0$. This can make the final objective function values of any method look deceptively poor. For example, a final function value of 10^{-7} on the nonsmooth version roughly corresponds to a final function value of 10^{-14} on the original smooth version.

Results for test set B are listed in Table 2, where a comparison with the classification and regression trees optimization method CARTopt [24] is made. CARTopt was compared with two other methods on test set B [24], and shown to be the best overall. No comparison is made on Wood's function in Table 2, as the two methods found different local minimizers.

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

TABLE 2. A comparison of test results for set B with CARTopt [24].

				NSQN		CARTOPT	
Problem	Function	n	m	f	nf	f	nf
25	Rosenbrock	2	2	6.9e-7	3605	3e-9	1184
26	Brown	2	3	4.3e-13	853	2e-3	50000
27	Beale	2	3	2.8e-11	3227	1e-9	1083
28	Helical	3	3	5.6e-3	9594	5e-9	1891
29	Gulf	3	99	4.5e-9	4140	5e-6	16405
30	Powell	4	4	1.2e-7	4703	7e-9	2756
31	Trigonometric	5	5	2.0e-7	5056	2e-8	4105
32	Variably dimensioned	8	10	2.2e-6	7223	4e-8	16182

CARTopt is quite different to the NSQN method. At each iteration, it uses a training set T which consists of a set of points in $\mathbb{R}^n$ together with objective function values at those points. Low points in T are identified and one or more box-shaped regions around them are constructed. The boxes are not necessarily aligned with the coordinate axes. Random samples are drawn from these boxes and are used to update the training set T, which completes an iteration. Some strengths of CARTopt are clear: for example, the random search can allow it to step directly into the vicinity of a minimizer, avoiding difficult intervening terrain. On the other hand, if a long step is required in a particular direction, then CARTopt might not be able to make that step as efficiently as a line search method. The results concur with these observations. The NSQN method is least competitive on the helical valley problem, where it travels from the initial point to the solution by following a "V" shaped curved valley. The NSQN method is clearly more effective on problems 26, 29 and 32. It tends to become more competitive as the dimension of the problem increases. This trend continues in test set C, where CARTopt fails on almost all problems in 20 and 50 dimensions.

Test set C contains nonsmooth problems listed by Bagirov and Ugon [7] and elsewhere. Following them, we used randomly generated starting points. For all problems except generalized Brown and ppsf Brown, the initial points were drawn randomly from $[0, 10]^n$. For the two Brown problems, initial points were drawn randomly from $[0, 1]^n$, so that the function values at the initial points are similar to those listed by Bagirov and Ugon [7]. Algorithm 1 solved all problems to the accuracy given by them on all runs. These results are listed in Table 3. Direct comparison with those listed by Bagirov and Ugon [7] is not possible because their algorithm uses gradient information where it exists.

On test set C, with $n = 10$, about 30% of the function evaluations are used by the HARS, including its line search. For test set B, this is about 31%. This drops to 18% for test set C with $n = 50$. This indicates that the global search process gets comparatively cheaper as the problem dimension increases.

Test set D consists of nonsmooth problems which were given by Lukšan and Vlček

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

TABLE 3. Results for test set C. The columns headed "$f - f^*$" and "nf" list 30 run averages of the error between the final and optimal functions values, and the number of function evaluations used to minimize f.

	$n = 10$		$n = 20$		$n = 50$	
Function	$f - f^*$	nf	$f - f^*$	nf	$f - f^*$	nf
chained LQ	5.5e-11	8092	1.2e-10	17479	5.7e-10	63890
chained CB3 I	3.3e-10	7772	5.6e-10	16843	1.3e-9	63634
chained CB3 II	1.5e-4	9188	1.4e-4	17213	1.6e-4	36280
generalized Brown	7.2e-11	6488	2.0e-10	13464	1.7e-10	45488
crescent I	1.8e-7	7731	4.8e-7	10654	2.2e-7	21776
crescent II	6.7e-7	11673	5.5e-7	21373	2.3e-6	53968
ppsf Brown	6.5e-11	6147	2.0e-10	14840	7.1e-10	45461
ppsf Broyden	4.6e-11	2376	8.5e-11	8765	8.4e-11	40770
ppsf CB3 I	4.1e-10	7433	6.1e-10	17032	1.9e-9	62432
ppsf CB3 II	5.1e-7	11034	4.7e-6	20040	4.4e-6	56304

TABLE 4. Results for test set D. The column "$f_{\text{at cdv}}$" lists the average function value for NSQN at the same number of function evaluations used in [11].

			CDV			NSQN	
Function	n	f^*	nf	f_{cdv}	$f_{\text{at cdv}}$	final f	nf
El-Attar	6	0.56	569	0.691	4.081	0.590	18853
EVD61	6	3.49e-2	335	9.07e-2	6.41e-2	3.81e-2	24329
Filter	9	6.19e-3	333	9.50e-3	8.51e-3	8.28e-3	7950
Goffin	50	0	17038	0	16.6	1.17e-9	92729
HS78	5	−2.92	212	2.07e-4	−2.75	−2.91053	6179
L1 Hilbert	50	0	7660	0.220	8.81e-5	6.52e-5	32148
max Hilbert	50	0	3164	1.24	4.05e-5	1.74e-6	27811
Osborne 2	11	4.803e-2	761	0.101	0.144	5.693e-2	54694
PBC1	5	2.234e-2	264	0.434	5.53e-2	2.766e-2	7070
Polak2	10	54.5982	1739	54.6	54.6	54.5982	3367
Shor	5	22.6002	257	23.4	22.6	22.6002	5161
Wong1	7	680.630	366	685	699	680.715	18356
Wong2	10	24.3062	763	25.8	42.0	24.6708	11542

[16] and which were solved by Custódio et al. [11]. Results for test set D are listed in Table 4. The column headed $f_{\text{at cdv}}$ lists the average function value for NSQN after the same number of function evaluations at which [11] halted. NSQN was ahead on seven of the thirteen problems at that point, and tied on one. When allowed to run until its stopping conditions were triggered, the NSQN method found better solutions on eleven of the thirteen problems, with a tie on one problem, at the expense of many more function evaluations than those done by Custódio et al. [11].

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

A different selection of problems from Lukšan and Vlček [16] was solved by Bagirov et al. [6] using randomly generated initial points. NSQN was compared with their method [6] using the standard initial points listed by Lukšan and Vlček [16] in place of randomly generated ones. Also, NSQN was limited to the same number of function evaluations as [6]. The results showed that the NSQN method did better on eleven problems, worse on five and tied with [6] on two. Two problems solved by Bagirov et al. [6] were not used, as they clearly located a different local minimizer to the solution listed by Lukšan and Vlček [16].

The behaviour of the NSQN method on nonsmooth problems was examined via test sets B–D. The convergence theory for nonsmooth problems is established via Corollary 3.7, and it requires the global direction search. Disabling the global direction search led to worse solutions being found on five of the nonsmooth problems in test sets B–D, and a total failure on two problems. Interestingly, this shows that a frame-based quasi-Newton method is capable of solving a significant number of nonsmooth problems, even though such problems are outside the scope of its convergence theory. On the other hand, if the quasi-Newton direction is replaced by the steepest descent direction, the performance of the algorithm deteriorates; it solves half of the problems in test set A, three problems in test set B and two from test set D. On most of these problems it is much slower than the NSQN method. If the quasi-Newton step is simply removed, the performance is even worse than with the replacement of the steepest descent direction.

5. Conclusion

A discrete quasi-Newton algorithm for minimizing black-box Lipschitz continuous functions has been proposed. Algorithm 1 performs as a discrete quasi-Newton method as a first choice, but resorts to global optimization techniques when progress is not forthcoming. The global direction search provides almost sure convergence to one or more Clarke stationary points under mild conditions. In addition, if local convexity or strict differentiability holds at such a point, then that point is stationary. Without the global direction search, it is only possible to show that the Clarke derivative is nonnegative in directions parallel to the coordinate axes. In this case, local convexity no longer guarantees stationarity.

The NSQN method has been numerically tested on a wide range of smooth and nonsmooth functions in up to 50 dimensions. Comparisons with other methods show that NSQN is respectable in terms of speed and robustness across this range of problems. This paper demonstrates the theoretical and practical viability of tackling Lipschitz continuous problems via direct-search-based quasi-Newton techniques.

Acknowledgement

The author would like to thank the anonymous referee for some very wise comments leading to a much improved paper.

https://doi.org/10.1017/S1446181117000323 Published online by Cambridge University Press

References

[1] Z. Akbari, R. Yousefpour and M. Reza Peyghami, "A new nonsmooth trust region algorithm for locally Lipschitz unconstrained optimization problems", *J. Optim. Theory Appl.* **164** (2015) 733–754; doi:10.1007/s10957-014-0534-6.

[2] M. J. Appel, R. Labarre and D. Radulović, "On accelerated random search", *SIAM J. Optim.* **14** (2003) 708–731; doi:10.1137/S105262340240063X.

[3] C. Audet and J. E. Dennis Jr., "Analysis of generalized pattern searches", *SIAM J. Optim.* **13** (2003) 889–903; doi:10.1137/S1052623400378742.

[4] 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.

[5] C. Audet and J. E. Dennis Jr., "OrthoMADS: a deterministic MADS instance with orthogonal directions", *SIAM J. Optim.* **20** (2009) 948–966; doi:10.1137/080716980.

[6] A. M. Bagirov, B. Karasözen and M. Sezer, "Discrete gradient method: derivative free method for nonsmooth optimization", *J. Optim. Theory Appl.* **137** (2008) 317–334; doi:10.1007/s10957-007-9335-5.

[7] A. M. Bagirov and J. Ugon, "Piecewise partially separable functions and a derivative-free algorithm for large scale nonsmooth optimization", *J. Global Optim.* **35** (2006) 163–195; doi:10.1007/s10898-005-3834-4.

[8] J. V. Burke, A. S. Lewis and M. L. Overton, "A robust gradient sampling algorithm for nonsmooth nonconvex optimization", *SIAM J. Optim.* **15** (2005) 751–779; doi:10.1137/030601296.

[9] F. H. Clarke, *Optimization and nonsmooth analysis*, Volume 5 of *SIAM Classics in Applied Mathematics* (SIAM, Philadelphia, 1990).

[10] I. D. Coope and C. J. Price, "Frame based methods for unconstrained optimization", *J. Optim. Theory Appl.* **107** (2000) 261–274; doi:10.1023/A:1026429319405.

[11] A. L. Custódio, J. E. Dennis Jr. and L. N. Vicente, "Using simplex gradients of nonsmooth functions in direct search methods", *IMA J. Numer. Anal.* **28** (2008) 770–784; doi:10.1093/imanum/drn045.

[12] C. Davis, "Theory of positive linear dependence", *Amer. J. Math.* **76** (1954) 733–746; doi:10.2307/2372648.

[13] J. E. Dennis Jr., Private communication (University of Canterbury, NZ, 2004).

[14] P. E. Gill, W. Murray and M. H. Wright, *Practical optimization* (Academic Press, London, 1981).

[15] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems", *Assoc. Computing Machinery J.* **8** (1960) 212–229; doi:10.1145/321062.321069.

[16] L. Lukšan and J. Vlček, "Test problems for nonsmooth unconstrained and linearly constrained optimization", Technical Report 798, Prague: Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000, http://www.apmath.spbu.ru/cnsa/pdf/obzor/TestProblemsforNonsmoothOptimization.pdf.

[17] M. Makela, "Survey of bundle methods for nonsmooth optimization", *Optim. Methods Softw.* **17** (2002) 1–29; doi:10.1080/10556780290027828.

[18] J. J. Moré, B. S. Garbow and K.E. Hillstrom, "Testing unconstrained optimization software", *ACM Trans. Math. Software* **7** (1981) 17–41; doi:10.1145/355934.355936.

[19] J. A. Nelder and R. Mead, "A simplex method for function minimization", *Comput. J.* **7** (1965) 308–313; doi:10.1093/comjnl/7.4.308.

[20] E. Polak and J. O. Roysett, "Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques", *J. Optim. Theory Appl.* **119** (2003) 421–457; https://link.springer.com/article/10.1023/B:JOTA.0000006684.67437.c3.

[21] C. J. Price and I. D. Coope, "Frame based ray search algorithms in unconstrained optimization", *J. Optim. Theory Appl.* **116** (2003) 359–377; https://link.springer.com/article/10.1023/A:1022414105888.

[22] C. J. Price, M. Reale and B. L. Robertson, "A direct search method for smooth and nonsmooth unconstrained optimization", *ANZIAM J.* **48** (2008) C927–C948; http://www.math.canterbury.ac.nz/~m.reale/pub/priceetal08.pdf.

[23] C. J. Price, B. L. Robertson and M. Reale, "A hybrid Hooke and Jeeves – Direct method for non-smooth optimization", *Adv. Model. Optim.* **11** (2009) 43–61; http://www.math.canterbury.ac.nz/~m.reale/wp/hjdirect.pdf.

[24] B. L. Robertson, C. J. Price and M. Reale, "CARTopt: a random search method for nonsmooth unconstrained optimization", *Comput. Optim. Appl.* **56** (2013) 291–315; doi:10.1007/s10589-013-9560-9.

[25] L.-P. Sun, "A quasi-Newton algorithm without calculating derivatives for unconstrained optimization", *J. Comput. Math.* **12** (1994) 380–386; http://www.jstor.org/stable/43692595.

[26] C. M. Tang, S. Liu, J. B. Jian and J. L. Li, "A feasible SQP-GS algorithm for nonconvex, nonsmooth constrained optimization", *Numer. Algorithms* **65** (2014) 1–22; doi:10.1007/s11075-012-9692-5.

[27] V. Torczon, "On the convergence of pattern search algorithms", *SIAM J. Optim.* **7** (1997) 1–25; doi:10.1137/S1052623493250780.

[28] L. N. Vicente and A. L. Custodio, "Analysis of direct searches for discontinuous functions", *Math. Program. Ser. A* **133** (2012) 299–325; doi:10.1007/s10107-010-0429-8.

[29] M. H. Wright, "Direct search methods: once scorned, now respectable", in: *Numerical analysis 1995 (Dundee, 1995)*, Volume 344 of *Pitman Res. Notes Math. Ser.* (Longman, Harlow, 1996) 191–208.

[30] T. Wu and L.-P. Sun, "A new quasi-Newton pattern search method based on symmetric rank-one update for unconstrained optimization", *Comput. Math. Appl.* **55** (2008) 1201–1214; doi:10.1016/j.camwa.2007.06.012.