AMO — Advanced Modeling and Optimization, Volume 13, Number 3, 2011

A Hooke and Jeeves—CARTopt hybrid method for nonsmooth optimization

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

Abstract

A direct search method for unconstrained local optimization is described, where the objective f can be nonsmooth or discontinuous. The method is a combination of the Hooke and Jeeves algorithm, and the random search CARTopt algorithm. Iterations of an altered Hooke and Jeeves algorithm are applied until no further progress is forthcoming from the current iterate z_m. The CARTopt algorithm is then applied in a neighborhood of z_m until further progress is made. If progress is made the method reverts back to the Hooke and Jeeves algorithm once more, otherwise CARTopt confirms z_m is an essential local minimizer of f with probability one. Alternating between Hooke and Jeeves, and CARTopt provides an effective method for nonsmooth optimization. The sequence of iterates $\{z_m\}$ is shown to converge to an essential local minimizer of f with probability one under mild conditions. Numerical results are presented to show the method is effective and competitive in practice.

keywords: CART, CARTopt, direct search, Hooke and Jeeves, nonsmooth, numerical results.

1 Introduction

The unconstrained optimization problem is of the form

$$\min_x f(x) \text{ subject to } x \in \mathbb{R}^n, \tag{1}$$

where a local minimizer is sought. The objective function f maps $\mathbb{R}^n$ into $\mathbb{R} \cup \{+\infty\}$ and in this paper f is assumed to be nonsmooth and may be discontinuous. However, we do require that f is lower semi-continuous and Lebesgue measurable. The inclusion of $\{+\infty\}$ means the method can be applied to barrier functions and discontinuous penalty functions. That is, constrained problems can be dealt with directly by assigning the value $f = +\infty$ to infeasible points.

Since f is assumed to be nonsmooth or discontinuous the standard definition of a local minimizer is modified.

Definition 1. (Essential local minimizer). *A point $x_* \in \mathbb{R}^n$ for which the set*

$$\mathfrak{E}(x_*, \epsilon) = \{x \in \mathbb{R}^n : f(x) < f(x_*) \text{ and } \|x - x_*\| < \epsilon\}$$

has Lebesgue measure zero for all sufficiently small positive ϵ is called an essential local minimizer of f.

[1] email: {`blair.robertson, chrisj.price`, or `marco.reale`}`@canterbury.ac.nz`

If f is continuous at x_*, then x_* is a local minimizer in the classical sense. The objective value $f(x_*)$ is called an essential local minimum.

Direct search methods can be applied to (1) when f is assumed to be nonsmooth or discontinuous because no gradient information is required. However, when such assumptions are placed on f theoretical convergence properties of these methods can be compromised. Partial convergence results have been presented by others. Audet and Dennis provide a method which looks asymptotically in all directions at each limit point of their algorithm [1], for which the Clarke derivative [6] is non-negative. However, the non-negativity of the Clarke derivative in all directions is only a partial result and does not preclude the existence of descent directions [13, 14, 16].

An alternative approach to solving nonsmooth problems was proposed by Price, Reale and Robertson in [13], where it was shown that the problem of finding a descent step on a nonsmooth function is closely related to a global optimization problem. Algorithms exploiting this connection have been presented in [13, 14, 16, 17]. These algorithms use global optimization techniques to exhaustively search for points of descent in a neighborhood of potential cluster points. The measure of this neighborhood is bounded away from zero for all iterations, a property crucial for establishing convergence on nonsmooth problems. It is this localized global phase that gives convergence on nonsmooth problems.

In this paper we investigate a hybrid algorithm consisting of local and localized global optimization phases in a similar way to [14]. The local phase of the algorithm uses an altered Hooke and Jeeves algorithm [10]. The localized global phase performs pure random search on a subset of $\mathbb{R}^n$. This subset is defined and sampled using the CARTopt algorithm [16, 17]. This random search method is provably convergent on nonsmooth or discontinuous problems.

Firstly, the hybrid algorithm's structure is introduced, where sections 2 – 5 detail the local phase and Section 6 describes the localized global phase. A method for generating a new search grid for the Hooke and Jeeves algorithm is proposed in Section 7. Convergence results for both smooth and nonsmooth objective functions are given in Section 8. The paper concludes with numerical results on a selection of nonsmooth optimization problems and concluding remarks.

2 The Algorithm

The algorithm proposed here is a variant of Hooke and Jeeves [10] which searches over a nested sequence of increasingly finer grids. The new algorithm differs from the classical version in three major ways. The algorithm may perturb and rotate the successive grids; limited increases in f are allowed; and a localized global search strategy is applied between grids.

For second, the sequence of objective function values $\{f(x_k)\}_{k=1}^{\infty}$ generated by the algorithm is not necessarily decreasing so that our algorithm approximates the motion of a rolling ball. In [20], Snyman presents an analogous physical problem to function minimization where the motion of a ball of unit mass in a n-dimensional conservative force field is considered. In such an approach, the objective f represents the ball's potential energy and total energy of the system, consisting of kinetic and potential energy, is conserved. The system,

$$\ddot{x}(t) = -\nabla f(x(t)), \qquad x(0) = x_0, \qquad \dot{x}(0) = \mathbf{0}, \tag{2}$$

describes the motion of such a ball [20], where x_0 is the initial point (position of ball) and the parameter $t \geq 0$ defines points on the trajectory. Approximating the trajectory defined by

(2) using various integration techniques gives a variety optimization methods, see for example [3, 18, 20]. However, while operating in a conservative force field and with no damping in (2), the ball is in continual motion. To ensure motion towards a local minimizer of f some form of damping must be added to the system. Snyman, for example, monitors the kinetic energy of the ball to ensure f is systematically reduced at each iteration. If a step increases f, the trajectory is abandoned and a new one is started from the penultimate point.

In this paper we a primarily interested in nonsmooth optimization and thus, can not rely on having explicit derivatives at each iteration as Snyman does [20]. In [16] it is shown that the classical Hooke and Jeeves algorithm provides an iterative approximation to the trajectory defined by (2) and requires no explicit derivative calculations. Rather than forcing the iterates to be strictly decreasing with respect to f, as Snyman does, uphill steps are taken so our sequence of iterates $\{x_k\}$ approximates the natural path of (2). Allowing for uphill steps means our method can use momentum to potentially *roll* over nonsmooth regions of f. The relative merits of such an approach is illustrated in Figure 3. However, to ensure the trajectory terminates (if $\{x_k\}$ is bounded) and is not in continual motion, an artificial damping is introduced to the system (see Section 4).

Thirdly, our algorithm differs from the classical Hooke and Jeeves algorithm when a point x_k has been located, for which no descent is forthcoming on the current grid. At this stage the classical algorithm reduces the mesh size and tries to reduce f on the finer grid. In our algorithm, a random search phase in a neighborhood of x_k is conducted to locate a point of descent, before a mesh reduction is considered. It is this localized global optimization phase that gives convergence on nonsmooth optimization problems by exhaustively searching for a point of descent in a neighborhood of x_k.

2.1 Grid Structure

The Hooke and Jeeves phase of the algorithm proposed here reduces f by searching on a succession of finer grids, indexed by m. Each grid $\mathcal{G}_m$ is defined by a mesh size $h_m > 0$, a grid center $\mathcal{O}_m$ (on the grid) and a set of n vectors $\mathcal{V}_m$ spanning $\mathbb{R}^n$, where

$$\mathcal{V}_m = \left\{ \nu_m^{(i)} = H_m e_i : i = 1, \ldots, n \right\}.$$

Here H_m is a Householder matrix and e_i is the i^{th} column of the identity matrix. Points on the grid $\mathcal{G}_m$ are given by

$$\mathcal{G}_m = \left\{ x \in \mathbb{R}^n : x = \mathcal{O}_m + h_m \sum_{i=1}^{n} \lambda_i \nu_m^{(i)} \right\},$$

where each λ_i is an integer. The vectors in $\mathcal{V}_m$ are parallel to the axes of the grid and steps between adjacent grid points are given by the vectors $h_m \nu_m^{(1)}, \ldots, h_m \nu_m^{(n)}$. The sequence of centers $\mathcal{O}_m$ means each grid can be offset relative to one another and the sequence of matrices H_m means each grid can be transformed relative to one another. Hence, a succession of finer grids is not necessarily nested. In the classical algorithm of Hooke and Jeeves, $\mathcal{O}$ is fixed and H is the identity matrix for all iterations.

The set of vectors

$$\mathcal{V}^+ = \{\mathcal{V}_m, -\mathcal{V}_m\} \tag{3}$$

form a positive basis for $\mathbb{R}^n$ [7] and hence, any vector in $\mathbb{R}^n$ can be written as a non-negative linear combination of vectors in $\mathcal{V}^+$. Using (3) a grid local minimizer is defined [5].

Definition 2. (Grid Local Minimizer). *A point $x \in \mathcal{G}(\mathcal{O}, h, \mathcal{V})$ is a grid local minimizer of an objective function f with respect to a positive basis $\mathcal{V}^+$ if*

$$f(x + h\nu) \geq f(x) \text{ for all } \nu \in \mathcal{V}^+.$$

2.2 Algorithm Details

```
1. Initialize:  Set $k = m = 0$, $v_0 = \mathbf{0}$, $H_0$ to the identity matrix and $h_{\min} > 0$.
   Choose $h_0 > h_{\min}$ and $x_0 = \mathcal{O}_0 \in \mathbb{R}^n$ such that $f(x_0)$ is finite.
2. Evaluate the objective function $f_0$, and set $\mathcal{U}_0 = f_0$.
3. while stopping conditions do not hold, do
   (a) Exploratory phase:  Calculate the exploratory vector $E_k$ at $x_k + v_k$.
   (b) Sinking lid:  If $f(x_k + v_k + E_k) \geq f_k$ and $\mathcal{U}_k \neq f_k$, then choose $\mathcal{U}_{k+1} \in [f_k, \mathcal{U}_k)$, else $\mathcal{U}_{k+1} = \mathcal{U}_k$.
   (c) Pattern move:  If $f(x_k + v_k + E_k) < \mathcal{U}_{k+1}$ then set
           $x_{k+1} = x_k + v_k + E_k$   and   $v_{k+1} = v_k + E_k$,
       increment $k$ and goto (a).
   (d) If $v_k \neq \mathbf{0}$ set $v_{k+1} = \mathbf{0}$, and goto (a).
   (e) Localized global optimization phase:  Set $z_m = x_k$.  Choose $\Omega_m \subset \mathbb{R}^n$
       such that $z_m \in \Omega_m$ and $m(\Omega_m) > 0$.  Execute CARTopt in $\Omega_m$ until a
       lower point $x_{k+1}$ is found or stopping conditions are satisfied.
   (f) Choose $h_{m+1}$, form $H_{m+1}$ and set $\mathcal{O}_{m+1} = x_{k+1}$.  Increment $k$ and $m$
       and goto (a).
   end
```

Figure 1: The Hooke and Jeeves—CARTopt hybrid algorithm

A precise statement of the main algorithm is given in Figure 1. The algorithm consists of an initialization phase and three nested loops. An exploratory phase (Step 3(a)) and a localized global phase (Step 3(e)) are listed as separate subroutines in sections 3 and 6 respectively. The dependence on $\mathcal{O}_m$ is suppressed in the following as $\mathcal{O}_m$ is the origin of $\mathcal{G}_m$. The notation $f_k = f(x_k)$ has been used for convenience.

The initialization phase sets the iteration counters k and m to zero, the Householder matrix H_0 to the identity matrix, a lower bound $h_{\min} > 0$ on the mesh size and v_0 to the zero vector. The user chooses an initial mesh size $h_0 > h_{\min}$ and an initial point $x_0 \in \mathbb{R}^n$ such that $f(x_0)$ is finite, which is also set to the grid center $\mathcal{O}_0$. It is not necessary to set v_0 to the zero vector. Setting a non-zero v_0 gives an initial *velocity* to the system and usually creates a different trajectory. This allows the user to explore different regions a possibly locate other essential local minimizers of f.

The two inner loops (steps 3(a) - 3(c) and steps 3(a) - 3(d), indexed by k) search on the grid $\mathcal{G}_m$ until a grid local minimizer is found. Step 3(d) implements a Hooke and Jeeves automatic restart, removing unsuccessful pattern moves by setting $v_{k+1} = \mathbf{0}$. This forces an exploratory phase from $x_{k+1} = x_k$, a successful iterate. The outer loop (steps 3(a) - 3(f), indexed by m) defines the new grid $\mathcal{G}_{m+1}$ for the inner loops to search over. The grid is aligned with a promising direction of descent, $x_{k+1} - x_k$, using a Householder matrix H_k, where x_{k+1} is generated in Step 3(e). This alignment potentially reduces the number of grid local minimizers and can increase the computational efficiency of the method. Each component of Step 3 is discussed in the sections which follow.

The algorithm terminates when either $h_m \leq h_{\min}$ or if the stopping conditions of CARTopt are satisfied. The interested reader is referred to [16] for details on the stopping rule of CARTopt. Once terminated, z_m is the candidate solution for an essential local minimizer of f.

3 Exploratory Phase

Each exploratory phase tries to reduce f by altering each grid coordinate, $\nu^{(i)} \in \mathcal{V}_m$, of the current exploratory search point $z = x_k + v_k$ in turn. This requires at least n, but no more than $2n$, f evaluations. Each step $h\nu^{(i)}$ is tried in turn. Firstly z is increased by $h\nu^{(i)}$. If this yields a reduction in f then this increase is retained and the method moves on to the next grid step $h\nu^{(i)}$. Otherwise the increase is removed, and z is decreased by $h\nu^{(i)}$ from its original value. Again, if this reduces f the decrease is retained, otherwise it is abandoned. In either case the process moves on to the next coordinate of z. The exploratory steps are retained in the vector E_k, where initially $E_k = \mathbf{0}$, as demonstrated in Figure 2 on the canonical grid.

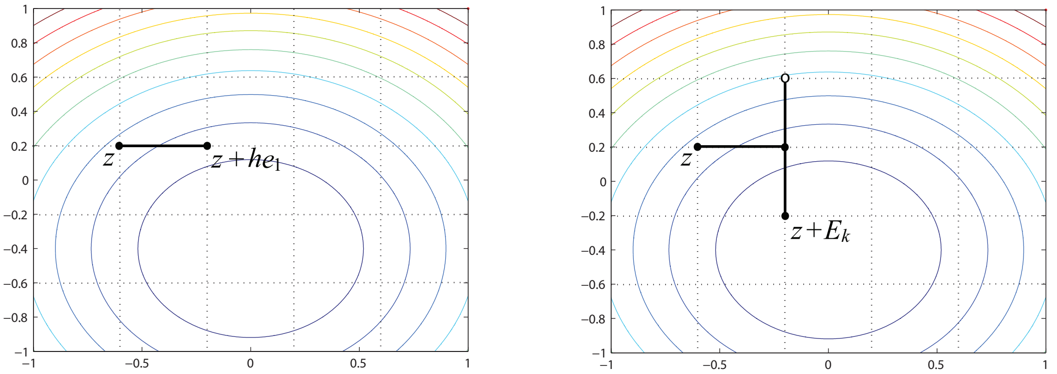

Figure 2: Hooke and Jeeves exploratory steps: Increasing z by he_1 gives descent and the step is retained. For the next coordinate $z + he_1$ is increased by he_2. This step increases f and the step is removed. $z + he_1$ is then decreased by he_2, which gives descent, and the step is retained. Thus, $E_k = [h, -h]$ in this example.

While operating on the grid $\mathcal{G}_m$, if the step $-h\nu^{(i)}$ was successful during the current exploratory phase, $-h\nu^{(i)}$ is tried first in the next exploratory phase. That is, x_2 would be decremented first for the next exploratory phase in our example in Figure 2. The classical exploratory phase increments each decision variable first, without reference to where the trajectory is heading. Including this additional heuristic has the potential to save n function evaluations at each exploratory phase.

4 Sinking Lid

The sinking lid (Step 3(b)) is a strategy used to ensure the trajectory does not oscillate or cycle endlessly if the sequence $\{x_k\}$ is bounded. This behavior can occur when multiple uphill steps are taken. Recall that the sequence $\{x_k\}$ approximates a ball rolling in a conservative force field and without damping can be in continual motion. The sinking lid systematically reduces an upper bound $\mathcal{U}$ on allowable function values each time ascent is made, creating a sequence of non-increasing upper bounds $\{\mathcal{U}_k\}$. Thus at iteration k the trajectory cannot surmount ridges where $f(x) > \mathcal{U}_k$. Provided $\{x_k\}$ is bounded, the lid eventually traps the ball in the neighborhood of an essential local minimizer, or at least terminates the trajectory in reasonable time.

Initially $\mathcal{U}_0 = f_0$ and thus, f cannot be increased above the initial function value. If descent is made $(f(x_k + v_k + E_k) < f_k)$ the upper bound remains unchanged with

$$\mathcal{U}_{k+1} = \mathcal{U}_k. \tag{4}$$

Otherwise, $f(x_k + v_k + E_k) \geq f_k$ and the upper bound is systematically reduced as follows. If $f(x_k + v_k + E_k) < \mathcal{U}_k$, then

$$\mathcal{U}_{k+1} = f_k + \frac{1}{2}\Big[\mathcal{U}_k + f(x_k + v_k + E_k) - 2(f_k + \tau)\Big]_+, \tag{5}$$

where $[.]_+ = \max\{., 0\}$ and $\tau > 0$ is a small positive constant. Otherwise,

$$\mathcal{U}_{k+1} = f_k. \tag{6}$$

If $[.]_+$ in equation (5) is positive, $\mathcal{U}_{k+1}$ is approximately halfway between the previous upper bound and f evaluated at the point of ascent given by

$$\mathcal{U}_{k+1} = \frac{\mathcal{U}_k + f(x_k + v_k + E_k)}{2} - \tau.$$

Thus the lid is reduced by at least τ, which prevents a series of arbitrarily small bound reductions. The value $\tau =$ 1e-10 is used hereafter. Otherwise $[.]_+$ is zero and $\mathcal{U}_k = f_k$ by (5). If the point of ascent has a f value greater than $\mathcal{U}_k$, the bound equals the previous successful iterate, $U_{k+1} = f_k$ by (6).

The following proposition shows that only a finite number of consecutive uphill steps can be taken in our approach before a point of descent is required. This ensures the existence of an infinite subsequence of $\{x_k\}$ with strictly monotonically decreasing f values. This subsequence is used to establish convergence in Section 8.

Proposition 3. *If f_k and $\mathcal{U}_k$ are finite, only a finite number of consecutive uphill steps are possible before a point of descent is required in the hybrid algorithm.*

Proof. Consider a point x_{k_*} on the trajectory such that f_{k_*} and $\mathcal{U}_{k_*}$ are finite. If $\mathcal{U}_{k_*+1} = f_{k_*}$, a point of descent is required from x_{k_*}. There are two possible ways this can happen. Either $f(x_{k_*} + v_{k_*} + E_{k_*}) \geq \mathcal{U}_{k_*}$ or $\mathcal{U}_{k_*} + f(x_{k_*} + v_{k_*} + E_{k_*}) \leq 2(f_{k_*} + \tau)$. In either case $\mathcal{U}_{k_*+1} = f_{k_*}$ (by (6) and (5)) and no uphill steps are taken from x_{k_*}.

Otherwise

$$\mathcal{U}_{k_*+1} = \frac{\mathcal{U}_{k_*} + f(x_{k_*} + v_{k_*} + E_{k_*})}{2} - \tau \tag{7}$$

and an uphill step is possible. Assume that infinitely many consecutive uphill steps are possible from x_{k_*}. Then there exists a sequence of f values $\{f_k\}_{k=k_*}^{\infty}$ such that $f_{k+1} > f_k$, and sequence of $\mathcal{U}$ values $\{\mathcal{U}_k\}_{k=k_*}^{\infty}$ such that $\mathcal{U}_{k+1} < \mathcal{U}_k$. Furthermore, $f_k < \mathcal{U}_k$ for each term in the sequences, otherwise descent is required by (6). After $m > 0$ consecutive uphill steps are taken

$$\mathcal{U}_{k_*+m} < \mathcal{U}_{k_*} - m\tau, \tag{8}$$

by (7). Choosing $m > (\mathcal{U}_{k_*} - f_{k_*})/\tau$ in (8) we have $\mathcal{U}_{k_*+m} < f_{k_*}$, which contradicts (5). Therefore, only a finite number of consecutive uphill steps can be taken by the algorithm before a point of descent is required. □

The trajectory produced using this method can define a path which leaves locally optimal points. This also reduces the tendency of the Hooke and Jeeves phase getting trapped by a discontinuity of f. However, the primary interest here is in local optimization and hence locating any optimal point is sufficient. Setting $\mathcal{U}_{k+1} = f_k$ for all k gives a strict descent (downhill) version of the altered Hooke and Jeeves algorithm.

5 Pattern Move

The Hooke and Jeeves pattern move may be expressed by the pair of equations

$$x_{k+1} = x_k + v_k + E_k, \quad \text{and} \quad v_{k+1} = \theta(v_k + E_k), \tag{9}$$

where E_k is the exploratory vector, $v_0 = \mathbf{0}$ and θ is a positive integer. The pattern move used here differs from the classical move in two ways. Firstly, by choosing $\theta > 1$ a more aggressive pattern move is achieved, allowing larger steps to be taken. Secondly, uphill steps are accepted provided f is not increased above the current upper bound $\mathcal{U}_k$. This allows f to temporarily increase along the trajectory approximating the path of a rolling ball given by (2). The relative merits of taking uphill steps on a nonsmooth function are illustrated in the following example.

5.1 Potential Benefits of taking Uphill Steps

Consider the nonsmooth version of Rosenbrock's function, given by

$$f(x) = |10(x_2 - x_1^2)| + |1 - x_1|, \tag{10}$$

where $x_* = (1, 1)$ is the unique essential local minimizer. This nonsmooth version of Rosenbrock's function shares all the difficulties of its smooth counterpart with an added sharp valley floor where f is non-differentiable. The reader is referred to Figure 3 where a contour plot of (10) is shown.

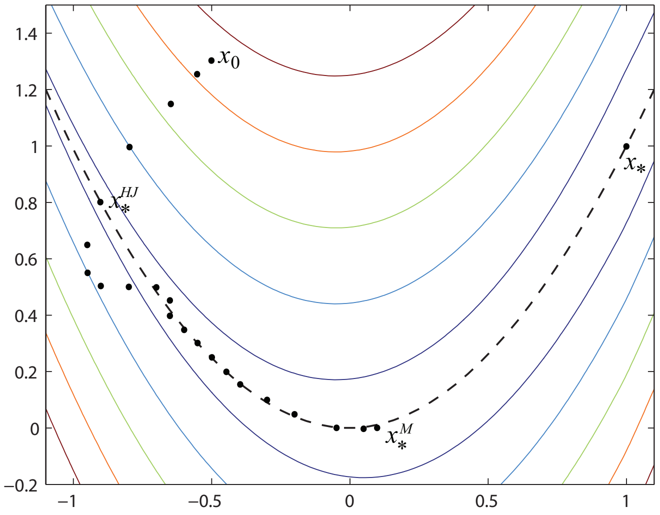

Figure 3: Trajectory generated by one iteration of the inner loop of the hybrid algorithm (dotted curve) for minimizing (10), terminating at x_*^M. The classical Hooke and Jeeves algorithm terminates at x_*^{HJ}. The dashed curve indicates the nonsmooth valley floor. Here $\theta = 1$, $x_0 = [-0.5, 1.3]$, $h_0 = 0.05$ and $\mathcal{G}$ is a grid with canonical basis vectors.

The benefits of allowing uphill steps are clear from Figure 3. Initially, both the classical and altered Hooke and Jeeves algorithms share the same trajectory. However, upon first contact with the nonsmooth valley floor the classical algorithm terminates at x_*^{HJ}. The classical algorithm then experiences a period of stagnation, requiring a series of mesh reductions before any progress is made toward x_*. Whereas, the altered algorithm *rolls* over the nonsmooth valley floor multiple times making progress toward x_* while operating on the same grid. Although the trajectory terminates at x_*^M, still some distance from x_*, much more progress is made. Our algorithm then uses a series of random search phases and grid transformations to drive the iterates to the solution. In addition, Figure 3 illustrates the potential for our algorithm to escape local nonsmooth wedges in f, potentially reducing the risk of stagnation, when f is largely smooth.

6 Localized Global Phase

The localized global optimization phase is used to exhaustively search for a point of descent in the neighborhood of $z_m = x_k$. In this paper we use the random search algorithm CARTopt [17] to identify and randomly sample promising sub-regions in the neighborhood of z_m. If CARTopt locates a point of descent the hybrid algorithm reverts back to the Hooke and Jeeves phase. Otherwise no descent is found and as we will show, z_m is an essential local minimizer of f with probability one.

CARTopt is a partitioning random search algorithm that alternates between partition and sampling phases. Initially a batch of points are drawn randomly from an optimization region, Ω. If a set of points $x \in \Omega$ with known f values exists, they can be used in the initial batch. The batch of points are classified into two sets, points with relatively high f values, ω_H, and points with relatively low f values, ω_L. It is necessary that ω_H and $\omega_L \neq \emptyset$. The set of classified points, $T = \omega_L \cup \omega_H$, is called the training data set. The training data set is used to form a partition on Ω using classification and regression trees (CART) [2, 8].

The CART partition defines desirable subsets of Ω where f is presumed to be relatively low based on T. The partition itself consists of a set of non-empty hyperrectanglar sub-regions, A_i, such that $\cup_i A_i = \Omega$ and $A_i^{\circ} \cap A_j^{\circ} = \emptyset$ for all $i \neq j$, where A_i° denotes the interior of A_i. Each A_i contains an element(s) from ω_L or ω_H, but not elements from both. The hyperrectangular partition structure is aligned with the coordinate axes which makes it simple to draw points directly from the union of sub-regions containing low points. The union of these low sub-regions is called an approximate level set, $\mathcal{L}$ [16, 17].

For the next iteration a batch of points X_k is drawn from $\mathcal{L}$. The best points from $T_k \cup X_k$ are retained in T_{k+1} and the training data is reclassified into new sets, ω_L and ω_H. A new CART partition on Ω is formed with T_{k+1} and the method repeats. The method continues alternating between these sampling and partition phases, defining new subsets of Ω to explore, until stopping conditions are satisfied. The interested reader is referred to [16] for full details on a bound constrained version of CARTopt and to [17] for an unconstrained instance. In this paper we consider using the bound constrained and unconstrained CARTopt methods in our hybrid algorithm.

6.1 Bound Constrained CARTopt Algorithm

The bound constrained CARTopt algorithm is applied in a hypercube neighborhood of z_m defined by

$$\Omega_m = \{x \in \mathbb{R}^n : \|H_m(x - z_m)\|_\infty \leq \rho\}, \tag{11}$$

where

$$\rho = \max\{3h_m/2, h_\Omega\} \tag{12}$$

is the hypercube radius and $h_\Omega > 0$ is a minimum radius imposed on each Ω_m. The minimum radius ensures $m(\Omega)$ is bounded away from zero for all iterations, which is crucial for establishing convergence on nonsmooth problems. The notation $m(.)$ is used to denote Lebesgue measure. If the objective function is known to be smooth, then $\rho = 3h_m/2$ is used. This allows $m(\Omega)$ to tend to zero as h tends to zero, potentially increasing the rate of convergence on smooth problems.

Points with known f values in Ω_m are included in an input training data set for CARTopt. These include the points from the sequence of iterates $\{x_k\}$ and points generated during the previous CARTopt phase. Let T_m^* denote the training data set from the m^{th} $(m \geq 1)$ CARTopt phase with $T_0^* = \emptyset$, then the input training data set is defined by

$$T_m = \{T_{m-1}^* \cup \{x_k\}\} \cap \Omega_m. \tag{13}$$

Recycling points with known function values makes computational sense, but care must be taken when reusing points in the CART partition. CART partitions Ω_m into a set of non-empty hyperrectangular sub-regions using a set of hyperplanes, each parallel to a coordinate axis [16]. Therefore, if two or more points in T_m share the same coordinate value, there exists a potential hyperplane orientation for which no split exists. This is not a problem when the elements of T_m are randomly drawn from Ω_m because the probability that two points share the same coordinate value is zero. However, some elements of $\{x_k\}$ may exist on a common grid and potentially share the same coordinate value.

To ensure no partitioning problems, two points $x, y \in T_m$ will not be split in the i^{th} dimension if

$$|x_i - y_i| < \epsilon, \tag{14}$$

where $\epsilon > 0$ and sufficiently small. Numerical results are generated with ϵ = 1e-15.

A direct consequence of (14) is that two points $x \in \omega_L$ and $y \in \omega_H$ will not be split if $\|x - y\| < \epsilon$. Therefore, it is possible that a sub-region of the partition can contain both a low and high point. We still consider any sub-region containing a low point, a low sub-region of the partition. This ensures there exists a non-zero probability of sampling a neighborhood of each low point during CARTopt's next iteration because low sub-regions are sampled further [17]. This property is needed to establish convergence on nonsmooth problems.

Very few sub-regions containing elements from both ω_L and ω_H were found whilst generating our numerical results. Including T_m in CARTopt's training data greatly increased the numerical performance of our hybrid algorithm. Few (or no) function evaluations were required to form the initial partition on Ω_m. Thus, CARTopt could identify and sample promising subsets of Ω_m at minimal cost. Setting $T_m = x_k$ is sufficient to establish convergence, but including the additional points makes computational sense.

The CARTopt phase can terminate in two ways. Firstly, if a point $x \in \Omega_m$ is found such that $f(x) < f(z_m)$, then x is set to the next iterate x_{k+1} and a new Hooke and Jeeves phase begins. Secondly, if the stopping conditions of CARTopt are satisfied [16]. The latter terminating hybrid algorithm and z_m is an approximation to an essential local minimizer of f.

6.2 Unconstrained CARTopt Algorithm

Rather than applying CARTopt in a hypercube search region, the unconstrained CARTopt algorithm performs random search on $\mathbb{R}^n$. At each iteration, a bounded subset of $\mathbb{R}^n$ is defined by a CART partition from which points are drawn randomly [17]. The partition is formed using a non-empty training data set T such that

$$|T| \leq \max\{2N, (n-1)N\}, \tag{15}$$

where N is CARTopt's batch size. If (15) holds with equality, T is called full size [17].

The unconstrained CARTopt algorithm is implemented in our hybrid algorithm using a training data set consisting of the best points generated by the hybrid algorithm. These include points from the sequence of iterates $\{x_k\}$, and points generated in CARTopt phases. Specifically, all points are retained in the hybrid algorithm until a full size training data set exists. Then a number of points with the largest f values are discarded to stop $|T|$ increasing above full size. This set of points with known f values is used as an input training data set for CARTopt.

Before a CART partition is formed, CARTopt requires a training data set $T = \omega_L \cup \omega_H$ such that $|T| \geq 2N$. If the input training data set has less than $2N$ elements, points are drawn randomly from

$$z_m + \frac{3h_m}{2}[-1, 1]^n \tag{16}$$

until $|T| = 2N$ so a CART partition on $\mathbb{R}^n$ can be formed.

The CART partition on $\mathbb{R}^n$ defines desirable subsets to explore further based on the best points generated by the hybrid algorithm. Although we sample the neighborhood of z_m, there may exist other low points found along the trajectory some distance from x_k that are also sampled. This is in contrast to the bound constrained CARTopt algorithm which is forced to focus its search in the neighborhood of z_m only (Ω_m). It may be advantageous to sample neighborhoods of other low points, rather than just the lowest, to reduce f. Using the unconstrained CARTopt algorithm in the hybrid algorithm achieves this.

If a point $x \in \mathbb{R}^n$ is found in the CARTopt phase such that $f(x) < f(z_m)$, $x = x_{k+1}$ and the method reverts back to the Hooke and Jeeves phase. Otherwise, when the stopping conditions of CARTopt are satisfied, z_m is taken as the estimate of an essential local minimizer of f.

7 Generating the Next Grid

The outer loop (steps 3(a) - 3(f) indexed by m) of the hybrid algorithm generates a new grid for the Hooke and Jeeves phase to search over. In this section we assume a point of descent, x_{k+1}, is located in Step 3(e) so a new grid is generated. Rather than simply having all grids nested, as in the classical Hooke and Jeeves algorithm, each grid is perturbed, rotated and scaled. Each is discussed in the subsections which follow.

7.1 Perturbation

Setting the grid center $\mathcal{O}_{m+1} = x_{k+1}$ — the newly generated point of descent from the localized global phase — the grid is perturbed in a straightforward manner. Since x_{k+1} is generated from randomly sampling subsets of Ω_m and $\mathcal{G}_m$ is a set of measure zero, $\mathcal{G}_{m+1}$ is offset relative to $\mathcal{G}_m$ with probability one. Hence, the altered Hooke and Jeeves algorithm is not confined to a sequence of nested grids.

7.2 Rotation

Each time a grid local minimizer is located, the computationally expensive localized global phase is initiated. Therefore, if transforming the grid has the potential to reduce the number of grid local minimizers, the number of localized global phases can also be reduced. In [4] it is shown that conjugate grids greatly reduce the number of grid local minimizers on a strictly convex quadratic function in two dimensions. Although these ideas could have potential here, a new approach is developed which potentially yields the same desired result without assuming a locally quadratic approximation.

The iterate x_{k+1} not only gives a point of descent, but also a promising direction of descent, given by $d = (x_{k+1} - x_k)/\|x_{k+1} - x_k\|$. Using the Householder matrix,

$$H_m = I - 2uu^\top \text{ with } u = (e_1 - d)/\|e_1 - d\|, \tag{17}$$

the x_1 axis of the grid $\mathcal{G}_{m+1}$ is set parallel to d, i.e. $H_m e_1 = d$. This transformation can dramatically reduce the number of grid local minimizers and allow the Hooke and Jeeves phase to make more progress.

To illustrate the relative merits of our transformation consider, for example, the function

$$f = \max\{a^\top x, b^\top x\} \text{ such that } x \in \mathbb{R}^2, \tag{18}$$

where $a = [0.7, -0.7]$ and $b = [-0.9, 0.5]$. The reader is referred to Figure 4 where the contours of (18) are shown. With x_k a grid local minimizer of f, a localized global optimization phase is performed in the neighborhood of x_k, locating a point of descent x_{k+1}. If only

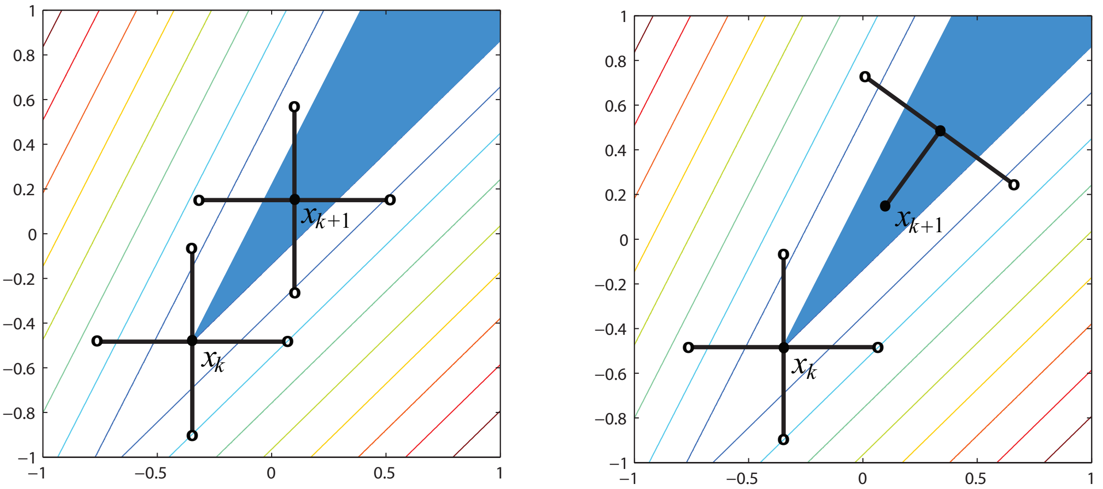

Figure 4: Grid perturbation with no rotation and grid perturbation with rotation. Points of descent and ascent are denoted •, o respectively and the shaded region indicates points of descent from x_k.

grid perturbations are performed, x_{k+1} remains a grid local minimizer of f and another localized global phase is conducted. This can repeat for many iterations, making our method computationally expensive. However, x_{k+1} is no longer a grid local minimizer if the grid is transformed using (17). The exploratory phase has successfully located a point of descent and a new Hooke and Jeeves trajectory is started. Although only a heuristic, this grid transformation dramatically increased the numerical performance of our algorithm.

7.3 Scaling

The mesh scaling heuristic used here is made with respect to the iterate x_{k+1} generated in the localized global phase. Firstly, consider the case when f is assumed to be nonsmooth. If

$$\|x_{k+1} - x_k\|_2 \geq h_m, \tag{19}$$

then $h_{m+1} = h_m$, otherwise

$$h_{m+1} = \max\{h_m/\tau_h, \|x_{k+1} - x_k\|\}, \tag{20}$$

where τ_h is a positive mesh reduction coefficient. Hereafter $\tau_h = 2$ is chosen, although any $\tau_h > 1$ can be used. If (19) is satisfied the mesh size remains unchanged. This may result in h_m not tending to zero as m tends to infinity, but convergence is still obtained through CARTopt with probability one. Otherwise the mesh is reduced using (20). The term h_m/τ_h is used as a bound on mesh reduction per iteration to prevent an unjustifiably small value for h_{m+1}, which may cause the algorithm to terminate early.

If f is known to be smooth, the mesh size is updated using

$$h_{m+1} = \max\left\{h_m/\tau_h, \min\{\|x_{k+1} - x_k\|, h_m/(1+\epsilon)\}\right\}, \tag{21}$$

where $0 < \epsilon \leq \tau_h - 1$ is a small positive constant. This choice of h ensures a mesh reduction is made if the CARTopt phase locates a point of descent, a property needed to establish convergence on smooth problems.

Proposition 4. *If f is assumed to be smooth and the sequence of grid local minimizers $\{z_m\}$ is infinite, then $h_m \to 0$ as $m \to \infty$ for all $1 < \epsilon + 1 \leq \tau_h$.*

Proof. From the form of (21), we have

$$\frac{h_m}{\tau_h} \leq h_{m+1} \leq \frac{h_m}{1+\epsilon},$$

where $1 < \epsilon+1 \leq \tau_h$. Therefore, h is reduced at each pass and with $\{z_m\}$ an infinite sequence the result follows. □

8 Convergence

In this section it is assumed that the stopping rules for both CARTopt and the altered Hooke and Jeeves algorithm are never invoked. This allows us to examine the asymptotic properties of the full sequence of iterates generated by the hybrid algorithm. The stopping conditions are included in the algorithms from a practical point of view.

8.1 Smooth Results

Firstly, let us consider the case when the objective function f is known to be smooth. These results are of interest here because the smooth version of the algorithm potentially converges faster on smooth problems than the nonsmooth version would. In order to establish the smooth convergence results the following assumptions are required.

Assumption 5. *The following conditions hold:*

(a) The points at which f is evaluated at lie in a compact subset of $\mathbb{R}^n$; and

(b) the sequence of function values $\{f(x_k)\}$ is bounded below.

These assumptions ensure the existence of cluster points in $\{z_m\}$ and exclude the case where $f(x_k) \to -\infty$ as $k \to \infty$.

The next theorem establishes the basic convergence result and follows closely from the results in [13]. This result uses Clarke's generalized derivative [6], which is, the *generalized directional derivative* of f in the direction d, defined by

$$f^o(x; d) = \limsup_{\substack{z \to x \\ \lambda \downarrow 0}} \frac{f(z + \lambda d) - f(z)}{\lambda},$$

where $z \in \mathbb{R}^n$ and λ is a positive scalar.

First we show that there exists a dense set of points in a neighborhood of z_m with larger f values if the m^{th} execution of Step 3(e) is infinite.

Proposition 6. *If Step 3(e) of the Hooke and Jeeves—CARTopt hybrid algorithm is an infinite process, then there exists a dense set of points with larger f values in the neighborhood of z_m with probability one.*

Proof. Step 3(e) is an infinite process and so CARTopt fails to generate a point with a function value less than $f(z_m)$. Noting that z_m is an element of CARTopt's input training data and no lower points are generated, the sequence of iterates $\{z_k\}$ generated by CARTopt is an infinite sequence with final value z_m (see Figure 1, Step 3(g) in [17]). It follows directly from Theorem 14 in [17] that z_m is an essential local minimizer of f with probability one. From the definition of an essential local minimizer (see Definition 1), the set

$$\mathfrak{E}(z_m, \epsilon) = \{x \in \Omega_m : f(x) < f(z_m) \text{ and } \|x - z_m\| < \epsilon\} \tag{22}$$

has Lebesgue measure zero for all sufficiently small $\epsilon > 0$. That is, there exists a dense set of points with larger f values in the neighborhood of z_m with probability one. □

The convergence result for the smooth version can now be given.

Theorem 7.

(a) Assume the sequences $\{z_m\}$ and $\{x_k\}$ are finite. If f is strictly differentiable at the final value z_{m*} of $\{z_m\}$, then $\nabla f(z_{m*}) = \mathbf{0}$.

(b) Let f be locally Lipschitz at z_*. If z_* is a cluster point of the sequence of grid local minimizers $\{z_m\}$ and if f is strictly differentiable at z_*, then $\nabla f(z_*) = \mathbf{0}$.

Proof. For part (a) the only way that the sequences $\{z_m\}$ and $\{x_k\}$ are finite is if Step 3(e) is an infinite process. Proposition 6 implies that there exists a dense set of points in $\|x - z_{m*}\| < \epsilon$ such that $f(z_{m*}) \leq f(x)$. The continuity (C^1) of f implies that $f(z_{m*}) \leq f(x)$ for all $\|x - z_{m*}\| < \epsilon$. The only possibility is $\nabla f(z_{m*}) = \mathbf{0}$.

For part (b) we restrict our attention to a subsequence $\{z_j\}$ for which the corresponding subsequence $\{z_j, \mathcal{V}_j^+\}$ converges uniquely to $(z_*, \mathcal{V})$. In what follows, ν_j is an element of the positive basis $\mathcal{V}_j^+$ (see (3)). Using the definition of a grid local minimizer we have,

$$f(z_j + h_j \nu_j) - f(z_j) \geq 0 \text{ for all } \nu_j. \tag{23}$$

Rewriting (23), dividing by h_j and taking the limit we have

$$\limsup_{j \to \infty} \frac{f(z_j + h_j(w_j + \nu)) - f(z_j + h_j w_j) + f(z_j + h_j w_j) - f(z_j)}{h_j} \geq 0,$$

where $w_j = \nu_j - \nu$. Noting that $w_j \to 0$, $z_j \to z_*$ and $h_j \downarrow 0$ (Proposition 4) as $j \to \infty$, the first two terms give Clarke's generalized derivative at z_* and so we have

$$f^o(z_*; \nu) + \limsup_{j \to \infty} \frac{f(z_j + h_j w_j) - f(z_j)}{h_j} \geq 0. \tag{24}$$

With f locally Lipschitz at z_* we have

$$|f(z_j + h_j w_j) - f(z_j)| \leq K \|z_j + h_j w_j - z_j\|,$$

for a positive scalar K. Thus, after applying the Lipschitz condition and evaluating the limit, the second term of (24) vanishes leaving

$$f^o(z_*; \nu) \geq 0. \tag{25}$$

All smooth functions have an open half space of descent directions centered on x if $\nabla f(x) \neq \mathbf{0}$. Strict differentiability at z_* implies

$$\exists w \in \mathbb{R}^n \text{ such that } f^o(z_*; \nu) = w^\mathsf{T}\nu \text{ for all } \nu \in \mathbb{R}^n$$

and so if w is non-zero, there exists an open half space for which $w^\mathsf{T}\nu < 0$. However, every positive basis has at least one vector probing any open half space (see Corollary 8 in [16]) and (25) states that all such directions have $w^\mathsf{T}\nu \geq 0$. Therefore, w must be the zero vector and hence, $\nabla f(z_*) = \mathbf{0}$ as required. □

8.2 Nonsmooth Result

The nonsmooth convergence result is now given. To establish convergence to an essential local minimizer of f the following assumption are required.

Assumption 8. *The objective function f is lower semi-continuous and bounded below.*

Assumption 8 ensures that $\liminf_{z \to z_*} f(z) \geq f(z_*)$, where z_* is a cluster point of $\{z_k\}$. Also $f(x_k) \not\to -\infty$ as $k \to \infty$ because f is bounded below.

Theorem 9. *Let Assumption 8 hold. Exactly one of the following possibilities holds:*

(a) $\{z_m\}$ is an infinite sequence and each cluster point z_* of the sequence is an essential local minimizer of f with probability one; or

(b) both $\{z_m\}$ and $\{x_k\}$ are finite sequences and the final z_m is an essential local minimizer of f with probability one; or

(c) $\{z_m\}$ is finite and $\{x_k\}$ is an infinite unbounded sequence.

Proof. Noting that a sequence cannot be both infinite and finite only one of these possibilities can hold.

Case (a) is a direct consequence of Theorem 14 in [17].

Let $\{z_m\}$ be a finite sequence and let m_* be the final value of m. There are two possible ways this can happen; either the inner loop is an infinite process or the localized global phase is an infinite process.

For the former we consider two cases. Firstly, consider the case when $\mathcal{U}_k$ is eventually constant such that $\mathcal{U}_k = \mathcal{U}_{k_*}$ for all $k > k_*$. In this case we have $f(x_k) < f(x_{k-1})$ for all $k > k_*$ by (4) and all $x_k \in \mathcal{G}_{m_*}$ for all k sufficiently large. Thus $\{x_k\}$ must be an infinite unbounded sequence, which is case (c).

Secondly, if $\mathcal{U}_k$ is not eventually constant, there exists a strictly decreasing subsequence $\{\mathcal{U}_i\}_{i=1}^{\infty}$ of $\{\mathcal{U}_k\}$ such that $\mathcal{U}_{i+1} < \mathcal{U}_i$ for all i by (5) and (6). The only way this can happen is if there are an infinite number of uphill steps taken by the algorithm because $\mathcal{U}$ is only reduced by (5) or (6) when ascent is made. With f bounded below and $f(x_0)$ finite, f_k and $\mathcal{U}_k$ are finite for all k. Therefore, Proposition 3 shows that only a finite number of consecutive uphill steps are possible before a point of descent is required. That is, there exists an infinite subsequence $\{y_i\}_{i=1}^{\infty}$ of $\{x_k\}$ for which $f(y_i) < f(y_{i+1})$ and all $y_i \in \mathcal{G}_{m_*}$ for all i sufficiently large. Thus $\{x_k\}$ must be an infinite unbounded sequence, which is case (c).

For the latter we have CARTopt being an infinite process. Theorem 14 in [17] implies that there does not exist a set

$$\mathfrak{E} = \{x \in \Omega_{m_*} : f(x) < f(z_{m_*}) \text{ and } \|x - z_{m_*}\| < \epsilon\}$$

with positive measure for all sufficiently small $\epsilon > 0$. Therefore, z_{m_*} is an essential local minimizer of f with probability one, which is case (b). □

Corollary 10. *If the sequence $\{x_k\}$ is bounded and $\{z_m\}$ is an infinite sequence, then every cluster point of the sequence $\{z_m\}$ is an essential local minimizer of f with probability one.*

Proof. With $\{x_k\}$ a bounded sequence, case (c) is removed from Theorem 9 and the result follows. □

9 Numerical Results and Discussion

The hybrid algorithm was tested on a number of problems from Hock *et al.* [9], Moré *et al.* [12] and Schittkowski [19]. The test problems selected were expressed as a sum of squares $\sum_i f_i^2$ with a global minimum of zero. These problems are easily made nonsmooth by replacing the sum of squares with absolute values $\sum_i |f_i|$. The nonsmooth versions share the same global minimizer(s) as their smooth counterparts because $f_i = 0$ for all i at the solution. However, these modified functions can make the results seem deceptively poor. A final function value of 1e-5 on the nonsmooth function, for example, corresponds to a function value of approximately 1e-10 on the original problem. Thus, we consider a final function value less than 1e-3 acceptable.

The Hooke and Jeeves—CARTopt hybrid algorithm was implemented with an initial mesh size $h_0 = e/2$ and a minimum mesh size $h_{\min} =$ 1e-8. This apparently strange h_0 value was chosen over the popular choice of $h_0 = 1$ because the latter allowed the algorithm to step exactly to the solution on some of the test problems considered. This gave a misleading impression on the performance of the algorithm. The standard optimization starting point x_0 (see Appendix A.1 in [16]) for each problem was used as the initial point for each problem. The localized global optimization phase uses either the bound constrained or unconstrained CARTopt algorithm, with the parameters defined as in [16, 17]. The minimum infinity norm radius $h_\Omega =$ 1e-4 on Ω_m was used when implementing the bound constrained CARTopt algorithm.

The unconstrained CARTopt instance was also implemented for comparison purposes with the parameters defined as in [17]. The initial hypercube search region size was chosen relative to Ω_1 (see (11)), given by

$$x_0 + \frac{3e}{4}[-1, 1]^n.$$

This choice allows us to make fair comparisons between the two methods.

The stopping rule from [16] was used to terminate the bound constrained and unconstrained CARTopt algorithms. This rule terminates CARTopt at iteration k if the distribution of the γ least function values follows a power law distribution and the probability of reducing f below $f(z_k) - \epsilon$ is less than β. Here the values $\epsilon =$ 1e-8, $\beta =$ 1e-6 and $\gamma = 40$ were used. The interested reader is referred to [16] for full details on the stopping rule.

Table 1 lists the results for the nonsmooth problems considered when the bound constrained CARTopt method was used in the hybrid algorithm and Table 2 lists the results using the unconstrained CARTopt method in the hybrid algorithm. The legends for these tables is defined as follows. The first three columns list the problem name, dimension and the final f value. Columns headed with ‘nf(%HJ)’ list the total number of f evaluations and values in parentheses list the percentage of f evaluations in the Hooke and Jeeves phase of

Table 1: Nonsmooth results using the bound constrained CARTopt method in the hybrid algorithm.

		Uphill			Downhill		
Problem	n	f_*	nf (%HJ)	term	f_*	nf (%HJ)	term
Beale	2	2e-9	1180 (14)	HJ	2e-9	1192 (15)	HJ
CB2	2	5e-9	1232 (11)	CART	5e-9	1031 (11)	CART
QL	2	6e-10	1357 (12)	CART	1e-9	1605 (12)	CART
Rosenbrock	2	2e-9	1684 (22)	HJ	5e-9	1409 (13)	HJ
Wolfe	2	4e-10	1293 (20)	HJ	1e-9	1090 (14)	HJ
Gulf	3	1e-4	26438 (83)	CART	3e-6	24368 (75)	CART
240	3	1e-8	1879 (27)	HJ	1e-8	1781 (25)	HJ
Helical Valley	3	1e-8	2510 (19)	HJ	1e-8	2285 (21)	HJ
Powell	4	3e-8	3828 (25)	HJ	3e-8	3514 (21)	HJ
261	4	2e-8	4171 (09)	HJ	1e-8	3891 (14)	HJ
Rosen-Suzuki	4	5e-5	9688 (24)	CART	4e-5	12310 (23)	CART
Trigonometric	5	6e-8	4429 (15)	HJ	6e-8	4505 (18)	HJ
Variably Dim.	8	4e-8	13936 (13)	CART	3e-8	14059 (12)	CART
291 (Quartic)	10	4e-9	6742 (02)	CART	4e-9	6632 (02)	CART

the algorithm. Columns headed with 'term' indicate the method of termination. This can be either through the CARTopt algorithm 'CART' or if the minimum mesh size is reached 'HJ'. The multicolumns headed with 'Uphill' and 'Downhill' list the results for the altered Hooke and Jeeves algorithm with uphill steps possible, and the version which requires descent at each iteration, respectively. Because the algorithm is stochastic, all values are averaged over ten runs.

The Hooke and Jeeves—CARTopt hybrid algorithms solved all the problems considered to the desired standard. The results show little difference between the downhill and uphill versions of the Hooke and Jeeves algorithm. The hybrid algorithm terminated when the minimum mesh size was reached on approximately half the problems. Otherwise the algorithm terminated through the CARTopt algorithm. Most of the computational effort was conducted in the CARTopt phase of the hybrid algorithm. The only exception was the Gulf problem, where a huge number of Hooke and Jeeves iterations were conducted between locating grid local minima.

Using the unconstrained CARTopt method in the hybrid algorithm was more effective than using the bound constrained CARTopt algorithm, giving the nine least f values on the fourteen problems considered. Furthermore, nine problems required fewer f evaluations to obtain similar approximate solutions. Most notably the bound constrained CARTopt hybrid algorithm required more than twice the number of f evaluations to solve the Gulf problem. This was largely due to the fact that the bound constrained CARTopt algorithm searched in Ω_m, whereas the unconstrained CARTopt algorithm searches in $\mathbb{R}^n$. This meant the unconstrained CARTopt algorithm could take larger steps and the bound constrained version was forced to take a series of little steps.

The previous results show that the hybrid method is effective, but is it competitive in practice? To show the method is competitive a comparison between the unconstrained CAR-

Table 2: Nonsmooth results using the unconstrained CARTopt method in the hybrid algorithm.

		Uphill			Downhill		
Problem	n	f_*	nf (%HJ)	term	f_*	nf (%HJ)	term
Beale	2	4e-9	1162 (15)	HJ	4e-9	1104 (15)	HJ
CB2	2	5e-9	882 (15)	CART	6e-9	928 (14)	CART
QL	2	7e-10	974 (14)	CART	8e-10	993 (14)	CART
Rosenbrock	2	4e-9	1194 (14)	HJ	2e-9	1279 (12)	HJ
Wolfe	2	1e-9	1001 (15)	HJ	7e-10	1012 (14)	HJ
Gulf	3	1e-5	9018 (64)	CART	2e-7	6712 (38)	CART
240	3	7e-9	2031 (26)	HJ	9e-9	1858 (23)	HJ
Helical Valley	3	1e-8	1979 (17)	HJ	8e-9	1983 (15)	HJ
Powell	4	3e-8	2962 (17)	HJ	2e-8	3158 (17)	HJ
261	4	2e-8	4092 (13)	HJ	1e-8	4191 (13)	HJ
Rosen-Suzuki	4	7e-4	5873 (09)	CART	9e-4	5086 (09)	CART
Trigonometric	5	5e-8	4934 (17)	HJ	6e-8	4852 (17)	HJ
Variably Dim.	8	2e-8	14671 (13)	CART	4e-8	14652 (13)	CART
291 (Quartic)	10	7e-9	7175 (22)	CART	9e-9	6938 (21)	CART

Table 3: Comparison with CARTopt and two other direct search methods for nonsmooth unconstrained optimization

		Results from [13]		Results from [14]		CARTopt (Unconstrained)		Hybrid (Downhill)	
Problem	n	f_*	nf	f_*	nf	f_*	nf	f_*	nf
Beale	2	4e-8	3638	2e-8	1119	1e-9	1091	4e-9	1104
Rosenbrock	2	5e-8	4438	2e-8	1154	2e-9	1248	2e-9	1279
Gulf	3	1e-5	15583	6e-6	31306	9e-5	20101	2e-7	6712
Helical Valley	3	7e-8	8406	1e-9	2773	3e-9	1837	8e-9	1983
Powell	4	4e-7	11074	3e-3	3659	1e-8	2744	2e-8	3158
Trigonometric	5	5e-8	14209	4e-8	6678	2e-8	4562	6e-8	4852
Variably Dim.	8	2e-7	34679	5e-7	55647	4e-8	13048	4e-8	14652

Topt instance and two direct search methods for nonsmooth unconstrained optimization from Price, Reale and Robertson [13, 14] is given in Table 3. The hybrid algorithm using the unconstrained CARTopt method is used in the comparison because it performed best on the seven comparison problems. The algorithm in [13] is a frame based method which performs a ray search along either a direct search quasi-Newton direction, or along a ray through the best frame point at each iteration. Random perturbations of the frames from time to time gives convergence on nonsmooth problems. The algorithm in [14] is similar to the Hooke and Jeeves—CARTopt hybrid algorithm presented here, using a series of local and localized global optimization phases. The classical Hooke and Jeeves algorithm is used in the local phase and the DIRECT algorithm of Jones, Perttunen and Stuckman [11] in the localized global optimization phase. This algorithm is deterministic and is provably convergent on nonsmooth problems.

The hybrid algorithm was superior to the algorithm from [13], producing more accurate approximate solutions on all problems except the Trigonometric problem, where the f_* values were similar. Approximately one third of the function evaluations were required to solve each problem.

In comparison to the algorithm from [14], slightly more accurate approximate solutions the to Helical Valley and Trigonometric problem were obtained, otherwise the hybrid algorithm produced more accurate approximate solutions. Only the Rosenbrock function required more f evaluations although a more accurate approximate solution was produced. The hybrid algorithm required noticeably fewer f evaluations on the relatively high dimensional problems, particulary the Variably Dimensional problem. It is the authors' opinion that this is largely due to the fact that the algorithm in [14] is deterministic, whereas our hybrid algorithm is stochastic, exploring higher dimensions more efficiently.

The unconstrained CARTopt algorithm and the hybrid algorithm performed similar on the problems considered. This is expected because the hybrid algorithm uses CARTopt in the localized global phase. However, the Hooke and Jeeves phase of the hybrid algorithm was extremely effective on the Gulf problem, giving an answer two orders of magnitude better and required less than one third the f evaluations. Because the hybrid algorithm performs at least as good as CARTopt, and potentially better on some problems, it is the authors' preferred choice.

10 Conclusion

A direct search method for nonsmooth unconstrained optimization has been presented. The method is a combination of an altered Hooke and Jeeves algorithm for which grid transformations, perturbations and uphill steps are permitted, and the CARTopt algorithm. Alternating between both algorithms is an effective method for nonsmooth optimization. Convergence to an essential local minimizer of f with probability one has been demonstrated under mild conditions. Numerical results have been presented which verify that theoretical convergence is achieved in practice on a number of nonsmooth problems. Comparison with two other direct search methods shows that the hybrid algorithm is better than both. The hybrid algorithm performed similar to the unconstrained CARTopt algorithm.

References

[1] C. Audet and J. E. Dennis, *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optimization, **17** (2006), pp.188-217.

[2] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and regression trees*, Monterey (CA), Wadsworth and Brooks, 1984.

[3] A. A. Brown and M. C. Bartholomew-Biggs, *Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations*, Journal of Global Optimization, **62** 1989, pp. 211–224.

[4] D. Byatt, I. D. Coope and C. J. Price, *Conjugate grids for unconstrained optimization*, Computational Optimization and Applications, **29** (2003), pp.49–68.

[5] I. D. Coope and C. J. Price, *On the convergence of grid-based methods for unconstrained optimization*, SIAM J. Optimization, **11** (2001), pp. 859–869.

[6] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classics in Applied Mathematics, SIAM, Philadelpha, 1990.

[7] C. Davis, *Theory of positive linear dependence*, American Journal of Mathematics, **76** (1954), pp. 733–746.

[8] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern classification*, Wiley-Interscience, New York, 2001.

[9] W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems (**187**), Springer-Verlag, 1981.

[10] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the Association for Computing Machinery, **8** (1961), pp. 212–229.

[11] D. Jones, C. D. Perttunen and B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, Journal of Optimization Theory and Applications, **79** (1993), pp.157–181.

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

[13] C. J. Price, M. Reale and B. L. Robertson, *A direct search method for smooth and nonsmooth unconstrained optimization*, ANZIAM J., **48** (2006), pp.927–948.

[14] C. J. Price, B. L. Robertson and M. Reale, *A hybrid Hooke and Jeeves—Direct method for nonsmooth optimization*, Advanced Modeling and Optimization, **11** (2009), pp.43–61.

[15] B. L. Robertson, C. J. Price and M. Reale, *Nonsmooth optimization using classification and regression trees*, Cairns, Australia: 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, 13-17 Jul 2009. In Proceedings of the 18th IMACS World Congress and MODSIM09 International Congress on Modelling and Simulation, 2009, pp.1195-1201.

[16] B. L. Robertson, *Direct search methods for nonsmooth problems using global optimization techniques*, PhD thesis, University of Canterbury, Christchurch, New Zealand, 2010.

[17] B. L. Robertson, C. J. Price and M. Reale, *CARTopt: A random search method for nonsmooth unconstrained optimization*, Manuscript in preperation, 2010.

[18] S. Schäffler and H. Warsitz, *A trajectory-following method for unconstrained optimization*, Journal of Optimization Theory and Applications, **67** 1990, pp.133–140.

[19] K. Schittkowski, *More test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems (**282**), Springer-Verlag, 1987.

[20] J. A. Snyman, *A new and dynamic method for unconstrained minimization*, Applied Mathematical Modelling, **6** 1982, pp.449–462.