A DIRECT SEARCH ALGORITHM FOR OPTIMIZATION WITH NOISY FUNCTION EVALUATIONS

EDWARD J. ANDERSON [*] AND MICHAEL C. FERRIS [†]

Abstract. We consider the unconstrained optimization of a function when each function evaluation is subject to a random noise. We assume that there is some control over the variance of the noise term, in the sense that additional computational effort will reduce the amount of noise. This situation may occur when function evaluations involve simulation or the approximate solution of a numerical problem. It also occurs in an experimental setting when averaging repeated observations at the same point can lead to a better estimate of the underlying function value. We describe a new direct search algorithm for this type of problem. We prove convergence of the new algorithm when the noise is controlled so that the standard deviation of the noise approaches zero faster than the step size. We also report some numerical results on the performance of the new algorithm.

1. Introduction. In the last few years there has been an increasing interest in direct search methods (Hooke & Jeeves 1961, Nelder & Mead 1965, Spendley, Hext & Himsworth 1962) for unconstrained optimization. These methods do not make gradient estimates, and involve relatively few function evaluations at each iteration. The most commonly used method in this class is due to Nelder & Mead (1965): this is a simplicial method which works using the repeated operations of reflection, expansion and contraction applied to a simplex of $n+1$ points in $\Re^n$. Though the Nelder-Mead method was invented more than 30 years ago, some version of this approach is probably still the most common way to carry out optimization when each function evaluation requires a separate experiment; this is despite the apparent superiority of other direct search methods, such as that due to Powell (Brent 1973, Powell 1964, del Valle, Poch, Alonso & Bartroli 1990).

Methods based on a simplicial approach have the disadvantage that for poorly behaved problems, they can fail to converge. (There are even examples of well behaved problems for which Nelder-Mead converges in theory to a non-stationary point (McKinnon 1998).) This has been recognized for some time and has led to a variety of suggestions for modifications which can help the convergence behavior in practice (e.g. Parker, Cave & Barnes (1985) and Hedlund & Gustavsson (1992)). At the same time researchers have developed versions of the basic simplicial algorithm which have provable convergence for certain classes of objective function. Examples include the work of Yu (1979), Rykov (1980), Torczon (1991), Torczon (1997), Lagarias, Reeds, Wright & Wright (1998) and Tseng (2000).

Another important restriction on methods in this class is that they have computation times that are very dependent on the dimension of the problem; they are not usually suitable for problems with more than a small number of variables. Indeed, for Nelder-Mead, there are even difficulties with convergence when the dimension of the problems becomes reasonably large. However this disadvantage may be partially offset by the fact that some direct search algorithms are capable of easy parallelization (Dennis & Torczon 1991).

A major factor in the continuing popularity of simplicial methods amongst users of optimization software is their ability to deal effectively with situations in which

[*] Australian Graduate School of Management, University of New South Wales Sydney 2052 Australia

[†] University of Wisconsin–Madison, Computer Sciences Department, Madison, WI 53706, USA. The research of this author was partially funded by the National Science Foundation and the Air Force Office of Scientific Research.

function evaluations are inaccurate. In this case more complex methods which approximate the function with some polynomial based on recent function evaluations (Conn & Toint 1996, Powell 1994) may be led seriously astray: even the estimation of gradient information by finite differencing must be carried out carefully (Gill, Murray, Saunders & Wright 1983).

In this paper we deal explicitly with the optimization of functions where the accuracy of the function evaluation depends on the time devoted to it. An example occurs when the function evaluation involves the solution of a PDE, where the accuracy depends on the grid size used. A second example occurs when attempting to optimize settings to achieve the maximum yield from a chemical reaction. Here the objective function is evaluated by carrying out a chemical experiment which is subject to random errors. For a given set of parameter values the experiment can be carried out many times over and then the average yield over the whole set of experiments gives an improved estimate of the objective function. Finally the same situation occurs in the design of a facility, for example a new warehouse, using a simulation model. The performance of the facility for some particular set of parameter values can be estimated using the simulation and the longer the simulation is run the more accurate will be the result. In each of these examples finding the best choice of parameter values requires the judicious balancing of time spent on improving the estimation of the objective function at a single point against time spent in making function evaluations at different parameter settings.

There are a variety of approaches to the problems of optimization with noise in the function evaluations. One approach is called the *Response Surface Methodology* (Khuri & Cornell 1987). This is straightforward: an estimation of the behavior of the objective function around the current point x is obtained by making some kind of factorial experiment using points in the neighborhood of x. A regression fit of a low order polynomial (usually linear) is then made to these points. Then a line search is carried out in the negative gradient direction before the whole process is repeated.

A related method, which uses a quadratic function which best fits the function values and chooses a descent direction based on this quadratic approximation has been proposed by Glad & Goldstein (1977). They establish a form of convergence result in the case that the noise is bounded. A similar approach has been used by Elster & Neumaier (1995) whose grid algorithm can be shown to be superior to Nelder-Mead on a variety of test problems when noise is included.

Another approach from the area of *Stochastic Approximation*, is the Keifer-Wolkowitz method (see for example (Kushner & Clark 1978) and (Polyak 1987)). This method estimates the gradient by evaluating the function at points $x \pm \alpha_k e_i$ where e_i are the n unit vectors and α_k is a constant depending on the iteration, and then takes a step of length γ_k in the negative gradient direction (rather than carrying out a line search). In order to obtain convergence when there is noise it is necessary to let γ_k and, especially, α_k tend to zero very slowly. There are various conditions but the crucial ones are that the infinite sum $\sum \gamma_k$ diverges, but that the infinite sum $\sum(\gamma_k/\alpha_k)^2$ converges. There are other stochastic approximation techniques which use increased sampling of $f(x)$, rather than decreasing step lengths to ensure convergence, see for example (Dupuis & Simha 1991). There are also some stochastic approximation techniques that involve a line search (Wardi 1988, Wardi 1990).

Barton & Ivey (1996) consider variations of the Nelder-Mead algorithm designed to cope with noisy function evaluations. These authors test alternative Nelder-Mead variants on a suite of test problems, using a stochastic noise term sampled from a truncated Normal distribution which is added to the underlying function.

In this paper we introduce a new simplicial direct search method which is designed for use with function evaluations subject to noise. We will prove convergence of this method subject to some assumptions on the behavior of the objective function. We believe that this is the first time that an analysis of the convergence behavior of a direct search algorithm with unbounded random noise has been carried out. We also present some preliminary computational results which demonstrate that the new method can be effective in practice.

There are a number of points of interest. First the algorithm includes a stochastic element. This is found to be advantageous in practice, and is easily incorporated into the stochastic framework of our analysis. The reader should note that the new method is not a pattern search method in the usual sense: there is no reason why the points at which the function is evaluated should be drawn from any kind of regular grid of points in $\Re^n$.

Second the stochastic nature of the function noise actually acts as an advantage in establishing convergence of the new algorithm. Paradoxically we do not have a proof of convergence for this algorithm in the case that there is no function noise. The reason for this is that, in the absence of noise, our algorithm might make an infinite sequence of iterations without contracting the size of the structure, and with function improvements tending to zero, without this implying that the gradient is close to zero. Our algorithm contains nothing to stop it cycling back to points closer and closer to points that have been visited before.

Finally the balance between function accuracy and step size is of interest. We show that convergence can be obtained when the standard deviation of the error of the function estimate decreases to zero faster than the step length size. There are reasons for thinking that convergence is unlikely if the standard deviation of the error becomes large in comparison to the step length so this may be the best possible result.

2. Description of the algorithm.

The algorithm we propose operates with a set of m points in $\Re^n$ at each iteration (with $m \geq n+1$). This set of points is called a *structure.*

In specifying the algorithm we need to ensure that each structure is of full dimension. For our purposes, a convenient way to measure the extent to which the structure is "flat" is to define, for any structure $S = \{x_1, x_2, \ldots x_m\}$,

$$d(S) = \min_{j=1,2,\ldots,m,\, u \in \Re^n,\, |u|=1} \{ \max_{k=1,2,\ldots,m} |(x_j - x_k)^T u| \}.$$

We also define the size of a structure S as

$$D(S) = \max_{j,k=1,2,\ldots,m} |x_j - x_k|.$$

We assume that there is an underlying objective function f defined on $\Re^n$ which we wish to minimize, and that each function evaluation we make is subject to some random noise. Thus the apparent function value at a point x is $\hat{f}(x) = f(x) + \xi$ where ξ is drawn from a distribution with zero mean and finite variance, σ^2. We will assume that the size of σ is under our control. For example we might take N function evaluations at a single point x and estimate the underlying function value $f(x)$ by averaging the results, in which case varying N allows us to control σ. The noise on successive function evaluations is independent. We will require increasing accuracy in our function evaluations as the structure size decreases.

We can generate new structures from a given structure S with the operations of reflection around a point x, or expansion around a point x (in each case x is one of the points of S):

$$\mathbf{reflect}(S, x) = \{2x - x_i | x_i \in S\},$$

$$\mathbf{expand}(S, x) = \{2x_i - x | x_i \in S\}.$$

We also need to define a structure $\mathbf{contract}(S, x)$ which is the result of a contraction operation. Just as expansion will double the size of a structure, contraction is defined in a way which essentially halves the size of a structure. This enables us to define the *level*, $l(S)$, of a structure S, such that the size of a structure S is a multiple $2^{-l(S)}$ of the size of the initial structure. A contraction operation increases the level by 1 and an expansion operation decreases the level by 1. More precisely, we assume that there are constants z_1 and z_2 such that for a structure S at level $l(S)$,

$$\frac{z_1}{2^{l(S)}} < d(S) < D(S) < \frac{z_2}{2^{l(S)}}. \tag{2.1}$$

We allow considerable freedom in the contraction operation. We require the structure $\mathbf{contract}(S, x)$ to contain the point x (which again is a point in S) and be such that the inequalities (2.1) will hold at all stages. There are a number of ways in which this can be done. One straightforward option is to take

$$\mathbf{contract}(S, x) = \{0.5(x + x_i) | x_i \in S\}. \tag{2.2}$$

Another possibility is to apply a random rotation around x to the set $\{0.5(x+x_i)|x_i \in S\}$. A third option is to apply a random perturbation to the elements in a core structure.

We suppose that the accuracy of the function evaluations made at any point depends on the level of the structure. At a higher level, when structures are smaller, we will need greater accuracy. We assume that the points in a structure S at level l are each evaluated in such a way that the noise has standard deviation σ_l where

$$2k_1 2^{-l(1+k_2)} \geq \sigma_l < k_1 2^{-l(1+k_2)}$$

for (small) constants $k_1 > 0$, $1/(m-2) > k_2 > 0$. Essentially this halves the standard deviation of the noise for each halving of structure size. Ignoring the small constant k_2, the standard deviation of the errors decreases at essentially the same rate as the size of the structure, $D(S)$.

For a structure S we define its value function as

$$F(S) = \min\{\hat{f}(x_j) | x_j \in S\}$$

and its best point (that we call the pivot point) as

$$v(S) = \arg\min\{\hat{f}(x_j) | x_j \in S\}.$$

The algorithm operates at each stage by pivoting about the point $v(S)$ for the current structure. The basic operation of reflection in the pivot point is carried out until no further improvement can be made, when there is contraction around the pivot point and the whole process starts again. At each stage that reflection produces an

```
given $S^0$ of full dimension and a sequence $\eta_i > 0, i = 1, 2, \ldots$
set $l = l_0$, $i = 0$, $b = F(S^0)$
while not satisfying termination criteria
begin
    $v^i = v(S^i)$
    $T = \mathbf{reflect}(S^i, v^i)$
    if $F(T) < F(S^i)$ then
    begin
        if $F(T) < b$ then $b = F(T)$
        $U = \mathbf{expand}(T, v^i)$
        if $F(U) < b - \eta_l$ and $l > 0$ then
        begin
            $S^{i+1} = U$
            $b = F(U)$
            $l = l - 1$
        end
        else $S^{i+1} = T$
    end
    else
    begin
        $S^{i+1} = \mathbf{contract}(S^i, v^i)$
        if $F(S^{i+1}) < b$ then $b = F(S^{i+1})$
        $l = l + 1$
    end
    $i = i + 1$
end
```

FIG. 2.1. *Pseudo code description of the algorithm*

improvement, an expanded structure is tested and accepted if this produces sufficient improvement on the best value so far. We can give a more formal definition of the algorithm with the pseudo code of Figure 2.1.

We shall assume that the algorithm has no memory of the points it has already evaluated, so that previous function evaluations at any point are not re-used if that point is revisited. This is also true for the pivot point when contraction takes place, but previous function values for the pivot point are used again for the reflection and expansion operations.

3. Convergence of the algorithm.

In this section we establish convergence of the algorithm under the following assumptions:

A1 The function f is uniformly Lipschitz and continuously differentiable, and has compact lower level sets.

A2 The noise distribution is Normal.

A3 The sequence η_i is bounded away from zero.

The first assumption is stronger than the assumptions made by (Tseng 2000) and (Torczon 1991) in their proofs of convergence for direct search methods where there is no noise component. These authors require that the function f is continuously differentiable and that the lower level set $L_0 = \{x \in \Re^n | f(x) \leq F(S^{(0)}\}$ is compact. Because of the stochastic nature of our algorithm, the structures generated do not necessarily contain a point in L_0 and this is one reason for the stronger assumption. We believe that some form of convergence will occur with a weaker condition than assumption A1: as it stands this restriction rules out well-behaved functions such as $f(x) = x^T x$. The final assumption effectively ensures that the probability of expansion decreases to zero as the algorithm proceeds.

Our analysis here is a stochastic one and all the probability statements we make are to be understood with respect to realisations of the noise process. We shall write $\gamma(x)$ for the probability in the tail of the standardized Normal distribution, so $\gamma(x) = \text{Prob}(\xi \geq \sigma x)$ where the noise ξ has an $N(0, \sigma^2)$ distribution.

We need to make use of the following inequality on $\gamma(x)$ (see for example Feller (1968))

$$\frac{1}{\sqrt{2\pi}} \left(\frac{1}{x} - \frac{1}{x^3} \right) e^{-x^2/2} \leq \gamma(x) \leq \frac{1}{\sqrt{2\pi}} \frac{1}{x} e^{-x^2/2}. \tag{3.1}$$

We write $\gamma_l(x)$ for the probability in the tail of the distribution for the errors in evaluations at level l; thus $\gamma_l(x) = \gamma(x/\sigma_l)$.

Before beginning the detailed analysis of the behavior of the algorithm it may be helpful to make some general remarks. The difficulty of the proof we give below arises primarily because the apparent function values are likely to increase whenever there is a contraction. To see why this is so, observe that, during the operation of the algorithm, we continually pivot around the best point in the structure: especially when the function is relatively flat this is likely to be a point with an evaluation error which is negative. The more negative the error, and the lower the apparent function value the more likely it is that none of the points in the reflected structure has an apparent value as small, and a contraction takes place. The pivot point is then reevaluated and is equally likely to have positive or negative error, giving a high probability of an increase in the apparent function value. Most of the work in the proof (Lemma 3.2 and Lemma 3.3) is required to establish that there is only a small probability of a large increase in apparent function value arising from a series of contractions.

We begin by showing that the structure size decreases to zero, i.e. that the level increases indefinitely.

LEMMA 3.1. *Under assumptions A1 and A2, $l(S^i) \to \infty$ as $i \to \infty$ with probability 1. If, in addition, assumption A3 holds, then with probability 1 there are only a finite number of expansions.*

Proof. We will prove this by regarding the algorithm as defining a stochastic process moving on the set of levels. First observe that, from assumption A1, the function f has a lower bound, which we will denote f^*.

We first suppose that the stochastic process is recurrent, so that there are infinitely many returns to some level i.

Since the function is uniformly Lipschitz there is some number k such that the real function difference between two points in a structure S at level i is less than $kD(S) < kz_2(0.5)^i$. From the remarks we have made above we know that there is a probability of at least a half that the pivot point has a negative error. (In what follows we use the term "error" to refer to $\hat{f}(x) - f(x)$ at some point x, where the error is "signed", rather than being an absolute magnitude.) Thus the probability of a contraction is always at least $q_i/2$, where q_i is the probability that all the points in the reflected structure (other than the pivot point) have errors greater than $kz_2(0.5)^i$.

This means that the probability of remaining at level i indefinitely (without steps to other levels) is zero.

Suppose that there are infinitely many steps taken away from the level i. The implication is that there are infinitely many expansion steps either at level i or $i-1$. Without loss of generality we suppose that these occur at level i. At each expansion step at level i, b decreases at least by a fixed amount. Thus we may choose a number M, so that after $M+j$ expansions at level i, $b < f^* - Kj\sigma_{i-1}$ for some $K > 0$.

Expansion can only take place if the apparent function value at one of the points in the expanded structure is less than b. Thus the probability of an expansion step after $M+j$ returns to level i is less than the probability that one or more of the errors in the points used in the expanded structure is more negative than $-Kj\sigma_{i-1}$. Since we reuse the pivot point in expansion, there are $m-1$ of these points we need to consider. The probability that one or more errors are less than $-Kj\sigma_{i-1}$ is thus less than $m-1$ times the probability that a single error is less than $-Kj\sigma_{i-1}$. Hence, if we write p_j for the probability of an expansion at level i after the $M+j$'th return to that level, then

$$\begin{aligned} p_j &< (m-1)\gamma_{i-1}(Kj\sigma_{i-1}) \\ &< \frac{m-1}{Kj\sqrt{2\pi}}\, e^{-(Kj)^2/2} \\ &< L^{-j^2} \end{aligned}$$

for some constant $L > 1$, provided j is chosen large enough.

We write $\tilde{p}_j$ for the probability that after $M+j$ returns to level i the next step away from i is to $i-1$ (rather than $i+1$). Now $\tilde{p}_j$ is equal to p_j divided by the probability of an expansion or contraction. This is no greater than $p_j/(p_j + q_i/2) < 2p_j/q_i$. The expected value of the remaining number of jumps from i to $i-1$ after M returns to level i is given by $\sum_{j=1}^{\infty} \tilde{p}_j < \sum_{j=1}^{\infty} 2p_j/q_i < (2/q_i)\sum_{j=1}^{\infty} L^{-j^2}$ which converges. Since this expectation is finite we can deduce that with probability 1 there are only a finite number of such jumps.

Thus with probability 1, the algorithm produces a sequence of levels which is not recurrent. Since the algorithm statement precludes l from becoming negative, the level must tend to ∞.

Under assumption A3, b decreases by at least some fixed amount at each expansion step, no matter at what level. We suppose that a sufficient number of expansion steps have taken place so that $b < f^* - K$, $K > 0$. Now consider the expected number of levels at which there is an expansion before a contraction on the first visit to the level. This is less than

$$\begin{aligned} &\sum_{l=1}^{\infty} \text{Prob (expansion before contraction at level l)} \\ &< \sum_{l=1}^{\infty} \frac{2(m-1)\gamma_l(K)}{q_l} \\ &< 2(m-1)\sum_{l=1}^{\infty} \frac{\gamma_l(K)}{\gamma_l(kz_2(0.5)^l)^{m-1}} \\ &< 2(m-1)\sum_{l=1}^{\infty} \frac{\gamma(K/\sigma_l)}{\gamma(kz_2(0.5)^l/\sigma_l)^{m-1}} \end{aligned}$$

$$< C_1 \sum_{l=1}^{\infty} \frac{\exp(-C_2/\sigma_l^2)}{\exp(-C_3(0.5)^{2l}/\sigma_l^2)}$$
$$< C_1 \sum_{l=1}^{\infty} \exp(-C_2 k_1 4^{(1+k_2)l} + C_3 2k_1 4^{k_2 l})$$

for some positive constants C_1, C_2, C_3.

The constant C_1 here must be chosen greater than

$$Q(l) = 2(m-1)(\sqrt{2\pi})^{m-2}\frac{\sigma_l}{K}\left(\frac{1}{x} - \frac{1}{x^3}\right)^{-(m-1)}$$

for $x = kz_2(0.5)^l/\sigma_l > k\sqrt{k_1}z_2 2^{lk_2}/\sigma$. To show that this is possible observe that, when l is large, x will be large enough for $\left(\frac{1}{x} - \frac{1}{x^3}\right) > \frac{1}{2x}$ and so

$$Q(l) < 2(m-1)(\sqrt{2\pi})^{m-2}\frac{\sigma_l}{K}(2x)^{m-1}$$
$$< C_4 2^{l(mk_2 - 2k_2 - 1)}$$

for some constant C_4, due to the upper bound on $N(l)$. This expression is bounded because of the choice of k_2.

Since for large enough l the term involving C_2 dominates that involving C_3, it is easy to see that this series has a finite value. Consequently there is probability zero of there being an infinite number of levels at which there is an expansion before a contraction on the first visit to the level. As we have already shown that with probability one there is only a finite number of expansions at any given level, we are done. $\square$

The next step is to show that the probability of an increase in $\hat{f}$ of a fixed size is bounded in an appropriate way. Throughout our analysis we will continue to make assumptions A1, A2 and A3.

Note that Lemma 3.1 shows that with probability 1 there is an iteration I after which there are no further expansion steps. Let $L(j)$ be an iteration at which a contraction at level j takes place. We write f_j for the function value f at $v(S^{L(j)})$ and $\hat{f}_j$ for the apparent function value at $v(S^{L(j)})$ *after* contraction takes place. At this stage $v(S^{L(j)})$ is one of the points in $S^{L(j)+1}$ but need not be the pivot point. We will keep track of both the function values f_j and the apparent function values $\hat{f}_j$.

Lemma 3.2. *Suppose that at step i of the algorithm at level $l = l(S^i)$, θ is chosen large enough so that*

$$2t > u > 2\sigma_l\sqrt{2},$$

where $t = \theta - f(v(S^i))$ and $u = \theta - \hat{f}(v(S^i))$, then, if $i > I$,

$$\text{Prob}(\max(f_l, \hat{f}_l) > \theta) \le \gamma_l(t) + \alpha\gamma_l(u/2) \tag{3.2}$$

where $\alpha = m/\gamma(2\sqrt{2})^{m-1} - 1$.

Proof. The probability we require depends on the function values which occur in all the structures that are evaluated between step i of the algorithm and the point at which contraction takes place for this level. We establish the bound we require by maximizing this probability over choices of function values.

Let $V_n(t,u)$ be the maximum value for the probability that the maximum of f_l and $\hat{f}_l$ is greater than θ conditional on a contraction occurring before n iterations. This is the maximum value of the probability in (3.2) with the restriction that a contraction at level l occurs within n steps. The maximum is taken over the choice of function values at the points in the structure $\mathbf{reflect}(S^i, v^i)$, and succeeding structures.

There is a dynamic programming recursion which links V_n and V_{n-1} of the form

$$\begin{aligned} V_n(t,u) = \max_{\theta_1,\theta_2,\ldots\theta_{m-1}} & (\text{Prob}\,(\text{immediate contraction})\gamma_{l+1}(t) \\ & + \text{Prob}\,(\text{reflection})E[V_{n-1}(t',u')]) \end{aligned}$$

where the θ_j are the real function values in $\mathbf{reflect}(S^i, v^i)$. Here t' and u' are the new values of t and u after reflection. The $\gamma_{l+1}(t)$ in the first term is the probability that $\hat{f}_l$ is greater than θ if contraction occurs immediately (we have assumed $t > 0$ so $f(v^i) < \theta$).

Observe that $V_0(t,u)$ is maximized by taking each θ_j very large forcing contraction. Hence $V_0(t,u) = \gamma_{l+1}(t) < \gamma_l(t)$ which satisfies the bound in (3.2). We will prove the result by induction on V_n. Once the bound is established for all values of n we are done. So we assume that the bound holds for $V_{n-1}(t,u)$.

Let $w = \hat{f}(v^i) - 2\sigma_l\sqrt{2}$. We shall consider two cases:

Case 1: Each $\theta_k > w$, $k = 1, 2, \ldots, m-1$.

We consider three possible events:

A : contraction occurs at this step,

B : reflection occurs at this step and $f(v^{i+1}) > \theta - u/2$,

C : reflection occurs at this step and $f(v^{i+1}) \le \theta - u/2$

Let p_A, p_B and p_C be the probabilities of these events . Since there will be a contraction if all the errors are greater than $2\sigma_l\sqrt{2}$,

$$p_A > \gamma_l(2\sigma_l\sqrt{2})^{m-1} = \gamma(2\sqrt{2})^{m-1}$$

If B occurs, then, since the apparent pivot value is less than $\hat{f}(v^i)$, the error is less than $-u/2$. The probability of one of the errors in the reflected structure being less than $-u/2$ is less than $(m-1)\gamma_l(u/2)$ which is thus an upper bound on p_B.

Finally note that if C occurs, then $t' = \theta - f(v^{i+1}) \ge u/2$ and, as the apparent function value at the new pivot is smaller than at the old one, $u' = \theta - \hat{f}(v^{i+1}) \ge u$. So if C occurs then $V_{n-1}(t',u') \le V_{n-1}(u/2,u)$. Using the fact that $V_{n-1}(t,u) \le 1$, we have

$$V_n(t,u) \le p_A\gamma_{l+1}(t) + p_B + p_C V_{n-1}\left(\frac{u}{2}, u\right)$$

Since $p_A \le 1$ and $p_C \le 1 - p_A \le 1 - \gamma(2\sqrt{2})^{m-1}$ we obtain the bound

$$\begin{aligned} V_n(t,u) &\le \gamma_{l+1}(t) + (m-1)\gamma_l\left(\frac{u}{2}\right) + (1-\gamma(2\sqrt{2})^{m-1})V_{n-1}\left(\frac{u}{2}, u\right) \\ &\le \gamma_l(t) + (m-1)\gamma_l\left(\frac{u}{2}\right) + (1-\gamma(2\sqrt{2})^{m-1})(\alpha+1)\gamma_l\left(\frac{u}{2}\right) \\ &= \gamma_l(t) + (m+\alpha-\gamma(2\sqrt{2})^{m-1}(\alpha+1))\gamma_l\left(\frac{u}{2}\right) \\ &= \gamma_l(t) + \alpha\gamma_l\left(\frac{u}{2}\right) \end{aligned}$$

as required.

Case 2: At least one θ_k is less than w.

We now need to consider four possibilities:

A : Contraction occurs at this step;

B : Reflection occurs and $f(v^{i+1}) \geq \theta - u/2$;

C : Reflection occurs, $f(v^{i+1}) < \theta - u/2$ and $\hat{f}(v^{i+1}) \geq w$;

D : Reflection occurs, $f(v^{i+1}) < \theta - u/2$ and $\hat{f}(v^{i+1}) < w$.

As in Case 1, we write p_A, p_B, etc. for the probabilities of these events, and we have the same upper bound on p_B. Also, as before, if C occurs then $V_{n-1}(t', u') \leq V_{n-1}(u/2, u)$. If D occurs, u' is larger and we obtain the stronger bound $V_{n-1}(t', u') \leq V_{n-1}(u/2, u + 2\sigma_l\sqrt{2})$. Thus

$$V_n(t,u) \leq p_A\gamma_{l+1}(t) + p_B + p_C V_{n-1}\left(\frac{u}{2}, u\right) + p_D V_{n-1}\left(\frac{u}{2}, u + 2\sigma_l\sqrt{2}\right)$$

Since $p_D \leq 1 - p_C$ we have

$$\begin{aligned} V_n(t,u) \leq \gamma_{l+1}(t) + (m-1)\gamma_l\left(\frac{u}{2}\right) + p_C V_{n-1}\left(\frac{u}{2}, u\right) \\ + (1-p_C)V_{n-1}\left(\frac{u}{2}, u + 2\sigma_l\sqrt{2}\right). \end{aligned}$$

The event C can only occur if all θ_k with a value less than w have a positive error, so $p_C \leq 1/2$. The right hand side of the inequality above is maximized when $p_C = 1/2$, so we obtain

$$\begin{aligned} V_n(t,u) &\leq \gamma_l(t) + (m-1)\gamma_l\left(\frac{u}{2}\right) + \frac{\alpha+1}{2}\gamma_l\left(\frac{u}{2}\right) \\ &\quad + \frac{1}{2}\gamma_l\left(\frac{u}{2}\right) + \frac{\alpha}{2}\gamma_l\left(\frac{u}{2} + \sqrt{2}\sigma_l\right) \\ &= \gamma_l(t) + \left(m + \frac{\alpha}{2}\right)\gamma_l\left(\frac{u}{2}\right) + \frac{\alpha}{2}\gamma_l\left(\frac{u}{2} + \sqrt{2}\sigma_l\right). \end{aligned}$$

Now

$$\begin{aligned} \gamma_l\left(\frac{u}{2} + \sqrt{2}\sigma_l\right) &= \gamma\left(\frac{u}{2\sigma_l} + \sqrt{2}\right) \\ &< \frac{1}{\sqrt{2\pi}}\,\frac{2\sigma_l}{u + 2\sigma_l\sqrt{2}}\ \exp\left(\frac{-u^2}{8\sigma^2}\right)\ \exp(-1) \\ &< \frac{2}{e\sqrt{2\pi}}\left(\frac{2\sigma_l}{u} - \left[\frac{2\sigma_l}{u}\right]^3\right)\ \exp\left(\frac{-u^2}{8\sigma^2}\right), \end{aligned}$$

since $(2\sigma_l/u)^3 < \sigma_l/u$. Thus again using the inequality (3.1)

$$\gamma_l\left(\frac{u}{2} + \sqrt{2}\sigma_l\right) < \frac{2}{e}\gamma\left(\frac{u}{2\sigma_l}\right) = \frac{2}{e}\gamma_l\left(\frac{u}{2}\right).$$

So

$$V_n(t,u) \leq \gamma_l(t) + \left(m + \frac{\alpha}{2} + \frac{\alpha}{e}\right)\gamma_l\left(\frac{u}{2}\right).$$

It is now easy to check that α is large enough for $m + (\alpha/2) + (\alpha/e) < \alpha$ which is the inequality we require to establish the bound in this case.

$\square$

Lemma 3.2 relates to the probability of achieving a high function value (or apparent function value) immediately after the next contraction. The next result is concerned with the probability of achieving a high value at any point after the current iteration. We will prove this by stringing together applications of Lemma 3.2. Let $W(\delta, i)$ be the probability that there is some level j, $j \geq i$ with either $\hat{f}_j > \hat{f}_i + \delta$ or $f_j > \hat{f}_i + \delta$. Let $\bar{W}(\delta, i)$ be the probability $W(\delta, i)$ conditional on $i > I$.

LEMMA 3.3. *If i_0 is large enough then $\bar{W}(\delta, i_0) \to 0$ as $\delta \to \infty$, and for any $\delta > 0$, $W(\delta, i_0) \to 0$ as $i_0 \to \infty$.*

Proof. First note the identity $\sum_{k=1}^{\infty} k/2^k = 2$. Thus $\bar{W}(\delta, i_0)$ is less than the probability, conditional on $i > I$, that for some j, $j \geq i_0$ either $\hat{f}_j$ or f_j is greater than

$$\Delta_j = \hat{f}_{i_0} + \delta/2 + (\delta/4) \sum_{k=1}^{j-i_0} k/2^k.$$

We can bound $\bar{W}(\delta, i_0)$ by the sum of the probabilities $q_j, j \geq i_0$ where q_j is the probability that one of $\hat{f}_j$ or f_j is greater than Δ_j, but that $\hat{f}_h$ and f_h are both less than Δ_h for $i_0 \leq h < j$, i.e. q_j is the probability that one of the inequalities is broken for the first time at level j.

Now q_{i_0} is the probability that $f_{i_0} > \hat{f}_{i_0} + \delta/2$, so $q_{i_0} = \gamma_{i_0}(\delta/2)$.

In general, for $j > i_0$ we wish to apply Lemma 3.2 to bound q_j. We know that $\hat{f}_{j-1}$ and f_{j-1} are both less than Δ_{j-1}. If $v^{L(j-1)}$ is also the pivot point in $S^{L(j-1)+1}$ then we will apply Lemma 3.2 with $i = L(j-1)+1$. Otherwise we let $S^* = \mathbf{reflect}(S^{L(j-1)+1}, v^{L(j-1)})$ be an artificial predecessor for $S^{L(j-1)+1}$. The bound in Lemma 3.2 is independent of the apparent function values at the points in structure S^i other than the pivot point. Hence we can apply Lemma 2 with $i = L(j-1)$ and using S^* instead of $S^{L(j-1)}$. In either case we obtain

$$\begin{aligned} q_j &< \gamma_j(\Delta_j - f_{j-1}) + \alpha\gamma_j\left(\frac{\Delta_j - \hat{f}_{j-1}}{2}\right) \\ &< \gamma_j\left(\frac{\delta(j-i_0)}{2^{j-i_0+2}}\right) + \alpha\gamma_j\left(\frac{\delta(j-i_0)}{2^{j-i_0+3}}\right) \\ &< (1+\alpha)\gamma_j\left(\frac{\delta(j-i_0)}{2^{j-i_0+3}}\right). \end{aligned}$$

In order to apply Lemma 3.2 we require that $\delta(j-i_0)/(2^{j-i_0+2})$ is greater than $2\sigma_j\sqrt{2}$. This inequality will hold provided i_0, and hence j, is chosen large enough.

Hence

$$\bar{W}(\delta, i_0) \leq \gamma_{i_0}(\delta/2) + \sum_{j=i_0+1}^{\infty} (1+\alpha)\gamma_j\left(\frac{\delta(j-i_0)}{2^{j-i_0+3}}\right)$$

Now $\sigma^2_{i_0+j} \leq \sigma^2_{i_0}/4^{j(1+k_2)}$, so we can use Chebychev to show that

$$\gamma_{i_0+j}(x) \leq \frac{\sigma^2_{i_0}}{x^2 4^{j(1+k_2)}}$$

Thus, for $j \geq 1$,

$$\gamma_{i_0+j}\left(\frac{\delta j}{2^{3+j}}\right) \leq \frac{\sigma^2_{i_0}}{\delta^2 j^2 4^{jk_2-3}}.$$

It also follows from Chebychev that $\gamma_{i_0}(\delta/2) \leq 4\sigma_{i_0}^2/\delta^2$ and so

$$\bar{W}(\delta, i_0) \leq \frac{\sigma_{i_0}^2}{\delta^2}\left(4 + 64(1+\alpha)\sum_{j=1}^{\infty}\frac{1}{j^2 4^{jk_2}}\right).$$

Since the infinite sum converges the first part of the result follows.

The second part of the result follows from observing that

$$W(\delta, i_0) \leq \text{Prob}(I > i_0) + \bar{W}(\delta, i_0)$$

and noting that the first term tends to zero from Lemma 3.1 and the second term also tends to zero since $\sigma_{i_0} \to 0$ as $i_0 \to \infty$. $\square$

With these three lemmas established we can now go on to prove our main result. We will show that, starting from a point with non-zero gradient there is, for a high enough level, a high probability of a succession of reflection steps leading to a reduction in $\hat{f}_j$ of a certain size, where the amount of reduction is independent of the level. Since the probability of ever seeing the same increase in $\hat{f}_j$ decreases to zero from Lemma 3.3, the probability of an infinite number of returns to a neighborhood of the initial point is zero. The consequence is that any cluster point has zero gradient - though we cannot establish that the cluster point is a local minimum.

This theorem concerns the behavior of a single sequence $\{v^i\}$ generated from running the algorithm. This may have multiple cluster points, but we will show that with probability 1 they will all have the same function value. Notice, however, that a new sequence generated from running the algorithm again might converge to a point with a different function value.

THEOREM 3.4. *If $\{v^i\}$ is the sequence of pivot points occurring when the algorithm is run, and assumptions A1, A2 and A3 hold, then with probability 1 there is a cluster point v^* for the sequence v^i and with probability 1, $\nabla f(v^*) = 0$ for each such cluster point. Moreover with probability 1 each cluster point has the same function value $\tilde{f} = f(v^*)$ and $\hat{f}(v^i)$ converges in probability to $\tilde{f}$.*

Proof. It will be convenient in the proof below to assume that $z_2 \leq 1$. Since z_2 is an upper bound on the size of the structure at level 0, we can make this assumption without any loss of generality.

We begin by discarding the first iterations of the algorithm till we reach a position in which the first statement of Lemma 3.3 applies. We choose an arbitrary $H_0 > 0$, and write v^0 for the pivot point after H_0 further steps of the algorithm. Let $\epsilon_1 > 0$ be arbitrary. Using Lemma 3.3 we can choose M large enough so that, conditional on there being no more expansion steps, there is a probability of less than ϵ_1 that $f_j > \hat{f}(v^0) + M$. Let $S = \{x | f(x) \leq \hat{f}(v^0) + M\}$ which is compact by assumption A1. Hence there is a probability of at least $1 - \epsilon_1$ that all the points at which there is a contraction are in S. But in this case there must be a cluster point in S. Since ϵ_1 was arbitrary, this establishes that, conditional on $I < H_0$, there is a cluster point with probability 1. Now, since this holds for all choices of H_0, we deduce that there is a cluster point with probability 1.

If v^* is a cluster point for v^i and we choose a subsequence $v^{k(i)}$ which converges to v^*, then $f(v^{k(i)}) \to f(v^*)$. Now observe that the choice of pivot point is influenced by the errors in evaluating all the points of a structure, but the error is equal to one of the at most $2m-1$ independent evaluation errors which can be involved at each step. Using Lemma 3.1 all evaluation errors approach zero, and hence $\hat{f}(v^{k(i)})$ converges in probability to $f(v^*)$.

Suppose that there is a subsequence $v^{k(i)}$ with $\hat{f}(v^{k(i)})$ converging to $f' > f(v^*)$. Since the apparent function values of the pivots can only decrease when the step is not a contraction, this implies that there is an infinite subsequence of $\hat{f}_j$ which is greater than or equal to f'. But since the $\hat{f}_j$ values must also make infinitely many visits to a neighborhood of $f(v^*)$, Lemma 3.3 implies that this occurs with probability zero. The same argument can be used to show that there is probability zero of a subsequence with apparent values converging to $f' < f(v^*)$.

We continue under the assumption of arbitrary $\epsilon_1 > 0$ and hence of the compact set S. Since f is continuously differentiable ∇f is uniformly continuous on S. Let $\epsilon_2 > 0$ be arbitrary. Choose $\delta_1 > 0$ such that for any $y \in S$, $|\nabla f(x) - \nabla f(y)| < (z_1/z_2)(\epsilon_2/4)$, for all $|x - y| < \delta_1$. Let $\delta_2 = z_1\delta_1/(64 + z_1)$.

Since S is compact we can find a finite set $y_1, y_2, \ldots y_k$ with

$$S \subset \bigcup_{j=1,2,\ldots k} B_{\delta_2}(y_j).$$

Let $A = B_{\delta_2}(y^*)$ be chosen from amongst this cover of S. We suppose that $3\epsilon_2/2 > |\nabla f(x')| > \epsilon_2$ for some $x' \in A$. From the definition of δ_1, $2\epsilon_2 > |\nabla f(x)| > \epsilon_2/2$ for every $x \in B = B_{\delta_1}(y^*)$.

Consider the steps of the algorithm while within B. We suppose that the current pivot point is v^h, and $l = l(S^h)$. If ∇f was constant within B and there were no evaluation errors, then we would obtain an improvement of $\max_j |\nabla f^T(v^h - x_j)|$ where the x_j are the other points in the current structure. This happens because we can choose either $v^h + (v^h - x_j)$ or $v^h - (v^h - x_j)$ as the next pivot: since the function values at these two points bracket that at v^h, only the one with the lower value appears in $\mathbf{reflect}(S^h, v^h)$.

From the definition of $d(S)$ we see that this improvement is at least $|\nabla f|\, d(S^h) > d(S^h)\epsilon_2/2$. Now we need to consider the variation in ∇f over B. The actual function value at the new improved point x_j may not be as low as would be predicted on the basis of a constant gradient of $\nabla f(v^h)$. Using the mean value theorem we have

$$f(x_j) = f(v^h) + \nabla f(v^h)^T(x_j - v^h) + (\nabla f(x) - \nabla f(v^h))^T(x_j - v^h)$$

for some x on the line segment (v^h, x_j). Thus the overall improvement predicted could be reduced by at most $|\nabla f(x) - \nabla f(v^h)||x_j - v^h|$, and since $|x_j - v^h| < D(S^h)$, the reduction is less than $d(S^h)\epsilon_2/4$. Hence, unless at least one evaluation error amongst the $2m - 1$ errors in the current and reflected structure is greater than $d(S^h)\epsilon_2/8$ in magnitude, we will observe an improvement of at least $d(S^h)\epsilon_2/8$ in both the actual and apparent values of f.

Suppose that we start with the pivot point v^h in the smaller ball, A. Consider a sequence of $\kappa = (\delta_1 - \delta_2)/D(S^h)$ steps of the algorithm. Note that throughout these steps the pivot point will be within B. If there is an improvement of $d(S^h)\epsilon_2/8$ at each step, then the total improvement after these κ steps is at least

$$(\delta_1 - \delta_2)\frac{d(S^h)\epsilon_2}{8D(S^h)} > \frac{64\delta_1}{64 + z_1}\frac{\epsilon_2 z_1}{8z_2} > 8\epsilon_2\delta_2,$$

where we use the fact that $z_2 < 1$.

We will obtain this overall improvement, unless at one of these κ steps the improvement is less than $d(S^h)\epsilon_2/8$. At each step there is an improvement of less than

$d(S^h)\epsilon_2/8$ with probability that is less than $(2m-1)\gamma_l(d(S^h)\epsilon_2/8)$. Hence the probability of not seeing the overall improvement is less than

$$\kappa(2m-1)\gamma_l(d(S^h)\epsilon_2/8) = \frac{\delta_1 - \delta_2}{D(S^h)}(2m-1)\gamma(d(S^h)\epsilon_2/(8\sigma_l)).$$

But $d(S^h)/\sigma_l > z_1\sqrt{k_1}2^{lk_2}$, and so, using inequality (3.1), it is easy to see that there will be a term in $\exp(-2^{2lk_2})$ arising from the γ function which will dominate the multiplier $1/D(S^h)$. Thus the probability of not obtaining the overall improvement $8\epsilon_2\delta_2$ approaches zero as the level l approaches ∞. Hence we can choose an iteration, h, large enough that the probability of an improvement of this size is greater than 1/2.

The maximum difference in function values between points in A is $4\epsilon_2\delta_2$. We choose h large enough so that the probability of the error at v^h being greater than $\epsilon_2\delta_2$ is less than 1/2. Thus with probability at least 1/4 both the apparent and actual function value are at least $3\epsilon_2\delta_2$ less than the smallest function value in A by the end of the sequence of κ steps in B. We choose h large enough for $W(3\epsilon_2\delta_2, l) < 1/2$. So provided h is sufficiently large the probability of never returning to A is at least 1/8. Thus the probability of an infinite number of returns is zero.

This establishes that the probability of a cluster point occurring in A is zero. Hence with probability 1 the cluster points of the pivot sequence occur in sets $B_{\delta_2}(y_j)$ in which every point has either $|\nabla f| > 3\epsilon_2/2$ or $|\nabla f| < \epsilon_2$. Hence at each cluster point v^* either $|\nabla f(v^*)| > 3\epsilon_2/2$ or $|\nabla f(v^*)| < \epsilon_2$. Since ϵ_2 is arbitrary, the appropriate choice of ϵ_2 rules out any strictly positive value of $\nabla f(v^*)$. Thus $\nabla f(v^*) = 0$. □

4. Computational Results.

In this section we report the results of some limited computational testing of the algorithm. Our aim is to investigate the performance of the new algorithm on a small set of test functions and we will look at its behavior both when function evaluations are noisy and when they are not. The algorithm requires the user to set some parameters; to choose an initial structure; and to determine whether or not to apply some sort of perturbation on contraction. The experiments we report are enough to indicate a reasonable choice of structure and settings for the parameters, as well as demonstrating the value of allowing a perturbation when there is controlled noise.

The algorithm we use is as described in section 2. The contraction operation (in the absence of a perturbation) just uses the definition of (2.2).

We have chosen to use a test suite of 11 problems. Two of these are well known: *rosen*, and *powell*. The Rosenbrock function (Moré, Garbow & Hillstrom 1981) *rosen* is a two dimensional problem with a single banana-shaped valley and is known to provide difficulties, particularly for direct search algorithms. The test function *powell* (Powell 1970) of dimension 4 is also well known. The problems *camel*, *ill3a*, *ill3b*, *sincusp* and *smalla* are taken from the problem set given in Aluffi-Pentini, Parisi & Zirilli (1988), being problem numbers 6, 10, 13, 35 and 37 from this set. The other four problems are straightforward quadratic functions. *quad6* and *quad6bad* are of dimension 6 with *quad6bad* being poorly conditioned (the reciprocal condition number is approximately 1e-6). Similarly *quad10* and *quad10bad* are quadratics of dimension 10, with the second function being poorly conditioned. All of the problems have small dimension; as we have already observed, direct search methods do not perform well when problems have a large number of variables.

We shall investigate two versions of the initial structure. They each have the same basic 'cross' form, with points radiating from a central point along all the axes

Table 4.1
No noise, no perturbation, 200 starts for each case

Problem	Struct	Average				Failures
		Iters	Evals	Obj Error	Distance	(major)
rosen	S_A	409	6696	0.2	0.4	9(9)
	S_B	59	1934	0.14	0.17	4(4)
camel	S_A	20	337	2.9e-06	0.078	0(0)
	S_B	20	682	5.2e-06	0.086	0(0)
ill3a	S_A	20	341	1.3e-06	0.00037	0(0)
	S_B	20	672	8.8e-06	0.00099	0(0)
ill3b	S_A	19	325	0.0053	0.015	6(4)
	S_B	22	725	0.015	0.048	20(10)
powell	S_A	62	2024	0.00051	0.14	0(0)
	S_B	46	3021	0.00013	0.094	0(0)
sincusp	S_A	43	1739	0.041	0.0012	0(0)
	S_B	40	3295	0.065	0.003	0(0)
smalla	S_A	45	1845	1.1e-06	0.001	0(0)
	S_B	40	3254	6.5e-06	0.0025	0(0)
quad6	S_A	47	2293	2.7e-06	0.0012	0(0)
	S_B	46	4475	1.6e-05	0.003	0(0)
quad6bad	S_A	452	21888	0.00063	11	0(0)
	S_B	156	15091	0.00042	10	0(0)
quad10	S_A	75	6064	1.2e-05	0.002	0(0)
	S_B	71	11433	7.3e-05	0.0049	0(0)
quad10bad	S_A	2346	188487	0.0011	15	0(0)
	S_B	710	113956	0.00084	14	1(0)

directions. The smaller structure, S_A, uses points

$$x^0, x^0 \pm \xi e^i, x^0 \pm 2\xi e^i.$$

Here e^i represents the ith unit vector in $\Re^n$and ξ is a scaling parameter, initially set to 1. The larger structure, S_B, adds the points $x^0 \pm 3\xi e^i$ and $x^0 \pm 5\xi e^i$ to S_A.

We ran the algorithm from 200 randomly generated starting points whose coordinates were uniformly generated within $(-10, 10)$. In each case we continued until the size of the structure was reduced so that adjacent points in the structure were within a distance 0.0001 of each other. This roughly parallels the stopping criteria used by Barton & Ivey (1996).

The first set of runs are reported in Table 4.1. These describe the performance of the algorithm for each test problem when there is no noise. In the absence of noise η_l is the only parameter to set. We choose $\eta_l = 10^{-8}$ at every step. Notice that in this case the algorithm is a pattern search method and the results of Torczon (1997) will imply convergence of the algorithm.

In the table we report, for both structure S_A and structure S_B, the average number of iterations, the average number of function evaluations required, the mean error in the solution value and the average distance of the final point found to the solution. In addition we report the number of failures of the algorithm in the final

Table 4.2
No noise, with perturbation, 200 starts for each case

Problem	Struct	Average				Failures
		Iters	Evals	Obj Error	Distance	(major)
rosen	S_A	46	770	0.065	0.2	14(7)
	S_B	38	1238	0.037	0.11	6(4)
camel	S_A	20	350	1.6e-06	0.085	0(0)
	S_B	20	671	1.4e-06	0.097	0(0)
ill3a	S_A	21	361	6.3e-07	0.00028	0(0)
	S_B	20	674	6.9e-07	0.00024	0(0)
ill3b	S_A	24	403	0.0039	0.011	4(2)
	S_B	24	800	0.0023	0.0076	4(1)
powell	S_A	48	1558	0.00057	0.12	0(0)
	S_B	38	2462	0.00013	0.079	0(0)
sincusp	S_A	31	1284	0.048	0.0018	0(0)
	S_B	29	2358	0.04	0.0012	0(0)
smalla	S_A	31	1259	2e-06	0.0013	0(0)
	S_B	28	2320	1.1e-06	0.00097	0(0)
quad6	S_A	36	1754	1.1e-05	0.0024	0(0)
	S_B	32	3112	5e-06	0.0016	0(0)
quad6bad	S_A	57	2778	0.0011	12	1(0)
	S_B	74	7161	0.0011	12	4(0)
quad10	S_A	52	4234	7.4e-05	0.005	0(0)
	S_B	44	7200	3.8e-05	0.0035	0(0)
quad10bad	S_A	137	11045	0.0057	20	27(0)
	S_B	204	32792	0.0038	18	13(0)

column - the first figure is the number of runs (out of 200) in which both the error in the solution value and the distance to the known solution are greater than 10^{-2}. The second figure in parenthesis is the number of major failures; identified as solutions in which both the error in the solution value and the distance to the solution are more than 10^{-1}. The problems *camel*, *ill3a*, *ill3b* all have multiple local optima. For these problems we take the closest local optima as the correct solution - this seems appropriate since the algorithm has not been designed to find a global optimum (even though in the early part of a run the method may well avoid being trapped in a local optimum in circumstances when more sophisticated techniques could fail).

It can be observed that the use of the larger structure, S_B, does not produce very much better solution quality as measured by distance to the solution, or the error in the function value. For *rosen*, and the ill-conditioned problems *quad6bad* and *quad10bad* the larger structure produces faster convergence with fewer function evaluations, but for the other functions the average number of function evaluations is smaller using S_A. Overall there is no significant advantage from using a larger structure.

The next set of results relate to the performance of the algorithm with random perturbations to the structure whenever a contraction is carried out. Thus, instead

Table 4.3

Controlled noise, no perturbation, 200 starts for each case

Problem	Struct	Average				Failures
		Iters	Evals	Obj Error	Distance	(major)
rosen	S_A	407	6658	0.2	0.43	11(10)
	S_B	62	2030	0.14	0.18	4(4)
camel	S_A	21	358	5.6e-06	0.074	0(0)
	S_B	22	723	5.1e-06	0.082	0(0)
ill3a	S_A	21	360	4.2e-06	0.00087	0(0)
	S_B	22	731	5.7e-06	0.00088	0(0)
ill3b	S_A	19	327	0.0053	0.015	6(4)
	S_B	22	730	0.016	0.051	21(11)
powell	S_A	60	1964	0.00075	0.16	0(0)
	S_B	44	2862	0.00026	0.11	0(0)
sincusp	S_A	42	1720	0.041	0.0012	0(0)
	S_B	40	3283	0.064	0.0029	0(0)
smalla	S_A	43	1766	4e-05	0.006	0(0)
	S_B	43	3459	1.6e-05	0.0037	0(0)
quad6	S_A	46	2272	3.9e-05	0.0049	0(0)
	S_B	48	4699	2.1e-05	0.0035	0(0)
quad6bad	S_A	105	5105	0.028	12	167(0)
	S_B	53	5111	0.0042	12	14(0)
quad10	S_A	76	6134	5.4e-05	0.0043	0(0)
	S_B	75	12077	5.6e-05	0.0043	0(0)
quad10bad	S_A	131	10554	0.037	17	198(2)
	S_B	63	10249	0.014	20	85(1)

of using the structures outlined above, we use the following for S_A:

$$x^0, x^0 \pm \xi(e^i + p_1^i), x^0 \pm 2\xi(e^i + p_2^i), x^0 \pm 3\xi(e^i + p_3^i),$$

where each component of the perturbations p_j^i are selected from a uniform distribution on $(-0.5, 0.5)$. S_B is similar except that the points $x^0 \pm 5\xi(e^i + p_4^i)$ are added.

Whenever a contraction is carried out by the algorithm, the following steps are performed:

1. Remove the current perturbation.
2. Contract towards the pivot point.
3. Add in a new randomly generated perturbation.

Note that the perturbations may realign the structure and this can be advantageous if it enables the structures to follow the geometry of the particular problem. Table 4.2 was generated using precisely the same mechanism as outlined for Table 4.1, except that random perturbations were incorporated.

The algorithm with perturbation often reaches a good solution more quickly than when there is no perturbation. This happens in 8 out of the 11 problems considered here. It is interesting that the speedup occurs on those problems where the algorithm without perturbation takes a large number of iterations, however there are insufficient results here to draw strong statistical conclusions. Moreover the solution quality with

Table 4.4
Controlled noise, with perturbation, 200 starts for each case

Problem	Struct	Average				Failures
		Iters	Evals	Obj Error	Distance	(major)
rosen	S_A	45	744	0.1	0.3	25(10)
	S_B	37	1225	0.018	0.073	6(3)
camel	S_A	21	356	5.4e-06	0.086	0(0)
	S_B	20	670	3.9e-06	0.096	0(0)
ill3a	S_A	21	365	4.3e-06	0.00079	0(0)
	S_B	20	677	2.8e-06	0.00061	0(0)
ill3b	S_A	24	402	0.0039	0.011	4(2)
	S_B	24	812	0.0018	0.006	3(1)
powell	S_A	47	1538	0.00074	0.13	0(0)
	S_B	37	2405	0.00017	0.088	0(0)
sincusp	S_A	32	1315	0.046	0.0017	0(0)
	S_B	29	2368	0.04	0.0012	0(0)
smalla	S_A	31	1293	1.5e-05	0.0037	0(0)
	S_B	29	2402	1e-05	0.003	0(0)
quad6	S_A	36	1783	2.4e-05	0.0038	0(0)
	S_B	32	3134	1.5e-05	0.0029	0(0)
quad6bad	S_A	47	2277	0.0027	13	3(0)
	S_B	34	3357	0.0016	12	0(0)
quad10	S_A	52	4221	8.3e-05	0.0054	0(0)
	S_B	45	7234	4.9e-05	0.0041	0(0)
quad10bad	S_A	71	5723	0.0074	17	43(0)
	S_B	53	8581	0.0075	17	40(0)

perturbations is often a little worse, so the superiority of perturbation is not clearly established. Just as in the case without perturbations, the advantage of using a larger structure is not significant given the greater number of function evaluations required.

Tables 4.3 and 4.4 repeat Tables 4.1 and 4.2 in the presence of noise. We deal with the case where the standard deviation of the noise is controlled. This is the situation for which the convergence of the algorithm has been established theoretically. We set $k_1 = 1$ and $k_2 = 0.1$ and hold other elements of the experiments as they were for Tables 4.1 and 4.2. The mean error is calculated from the underlying function value f at the final pivot point.

Observe that in Table 4.3 there are significant numbers of failures for *quad6bad* and *quad10bad*, but the overall performance of the algorithm in Tables 4.3 and 4.4 is reasonable. If Tables 4.3 and 4.4 are compared, one can see that in almost all cases using perturbation produces better quality solutions with fewer function evaluations. For *rosen* the number of non-major failures is worse with perturbation. Nevertheless it seems that perturbation offers an overall advantage on this set of test problems. It is possible that a similar perturbation approach could be beneficially applied to other direct search algorithms.

Using a larger structure gives better results, but there is some penalty in function evaluations. With both perturbation and the larger structures there are only 4 major

Table 4.5
Uncontrolled noise, no perturbation, 200 starts for each case

Problem	Struct	Average				Failures
		Iters	Evals	Obj Error	Distance	(major)
rosen	S_A	20	342	2.7	2.5	190(140)
	S_B	20	682	1.9	2.1	192(124)
camel	S_A	21	362	0.04	0.13	156(15)
	S_B	21	710	0.038	0.13	161(8)
ill3a	S_A	21	363	0.05	0.086	161(25)
	S_B	21	705	0.049	0.084	151(17)
ill3b	S_A	20	350	0.08	0.047	31(9)
	S_B	22	743	0.082	0.093	51(21)
powell	S_A	29	974	0.13	0.44	192(83)
	S_B	29	1932	0.11	0.4	191(64)
sincusp	S_A	31	1267	0.7	0.41	200(169)
	S_B	30	2458	0.64	0.38	200(160)
smalla	S_A	31	1277	0.13	0.34	200(112)
	S_B	30	2444	0.12	0.33	200(103)
quad6	S_A	33	1637	0.17	0.31	199(141)
	S_B	33	3221	0.16	0.3	200(146)
quad6bad	S_A	25	1223	0.2	13	194(114)
	S_B	24	2377	0.18	13	195(108)
quad10	S_A	47	3834	0.32	0.33	200(193)
	S_B	47	7645	0.31	0.32	200(191)
quad10bad	S_A	27	2242	0.56	18	200(188)
	S_B	26	4256	0.37	17	200(177)

failures in the complete set of 2200 starts and in this case the performance of the algorithm is remarkably good. It is interesting that, with this setup, there is remarkably little degradation in performance due to the introduction of noise, as can be seen by a comparison of the relevant parts of Tables 4.2 and 4.4. Indeed, for some problems there appears to be an improvement in performance. Observe that for *rosen* there is one less major failure and smaller average errors than were achieved in any of the noiseless cases, and there are also substantially fewer function evaluations required for both *quad6bad* and *quad10bad*.

Finally in the last two tables, Tables 4.5 and 4.6, a fixed standard deviation of the noise is used, with σ set to 0.1 at all levels. As before the initial value of the scaling parameter is set to 1. These tables demonstrate how hard it is for direct search algorithms to perform well in the presence of uncontrolled noise. There are large numbers of major failures for all the functions tested.

5. Discussion and Conclusions.

We have presented an algorithm for the optimization of functions that are subject to stochastic error in their evaluation. The algorithm is shown to have a cluster point that is a stationary point of the function with probability 1. This is established using a novel proof technique.

The computational results show the method is effective in the presence of noise,

TABLE 4.6
Uncontrolled noise, with perturbation, 200 starts for each case

Problem	Struct	Average				Failures
		Iters	Evals	Obj Error	Distance	(major)
rosen	S_A	22	372	1.7	1.8	188(119)
	S_B	22	722	1.3	1.7	178(117)
camel	S_A	21	362	0.041	0.13	155(20)
	S_B	21	700	0.031	0.13	145(10)
ill3a	S_A	22	370	0.038	0.075	145(14)
	S_B	21	703	0.033	0.066	149(11)
ill3b	S_A	23	390	0.054	0.046	32(7)
	S_B	23	754	0.056	0.057	35(11)
powell	S_A	30	995	0.15	0.46	196(95)
	S_B	28	1827	0.075	0.33	180(43)
sincusp	S_A	28	1157	0.61	0.36	200(159)
	S_B	28	2282	0.46	0.2	199(106)
smalla	S_A	27	1108	0.11	0.31	198(92)
	S_B	27	2241	0.079	0.26	193(63)
quad6	S_A	29	1451	0.13	0.27	198(108)
	S_B	28	2789	0.1	0.24	199(87)
quad6bad	S_A	24	1210	0.14	13	189(98)
	S_B	24	2331	0.13	13	186(80)
quad10	S_A	38	3136	0.28	0.31	200(185)
	S_B	36	5869	0.19	0.26	200(172)
quad10bad	S_A	27	2190	0.32	18	200(159)
	S_B	25	4139	0.31	17	200(162)

if this is controlled in the appropriate way. Our proof of convergence holds in the case that we allow a perturbation of the structure on contraction. Using this perturbation approach is shown to improve the computational performance of the algorithm on the test set we have considered when there is controlled noise. It is possible that the introduction of a perturbation of this kind could improve the performance of other direct search methods. A significant contribution of this paper is the introduction of this algorithmic innovation, together with a proof that the method converges in this case.

However, the convergence result we establish here is weaker than would be desirable. In effect we have arranged the test for expansion to ensure that with probability one there are only a finite number of expansion steps, and then established convergence when no more expansions take place. Nevertheless there seems no good reason to suppose that convergence will fail when there are an infinite number of expansion steps, i.e. under weaker conditions than assumption A3. For example we might allow η_i to tend to zero but ask that $\sum \eta_i$ diverges. However a result of this kind seems hard to prove.

REFERENCES

Aluffi-Pentini, F., Parisi, V. & Zirilli, F. (1988), 'Algorithm 667: SIGMA- A stochastic-integration global minimization algorithm', *ACM Transactions on Mathematical Software* **14**, 366–380.

Barton, R. & Ivey, J. S. (1996), 'Nelder Mead simplex modifications for simulation optimization', *Management Science* **42**(7), 954–973.

Brent, R. P. (1973), *Algorithms for Minimization without Derivatives*, Prentice-Hall, Inc, Englewood Cliffs, New Jersey.

Conn, A. R. & Toint, P. L. (1996), An algorithm using quadratic interpolation for unconstrained derivative free optimization, *in* G. D. Pillo & F. Giannessi, eds, 'Nonlinear Optimization and Applications', Plenum Press, New York.

del Valle, M., Poch, M., Alonso, J. & Bartroli, J. (1990), 'Comparison of the Powell and simplex methods in the optimization of flow-injection systems', *Analytica Chimica Acta* **241**, 31–42.

Dennis, J. E. & Torczon, V. (1991), 'Direct search methods on parallel machines', *SIAM Journal on Optimization* **1**, 448–474.

Dupuis, P. & Simha, R. (1991), 'On sampling-controlled stochastic approximation', *IEEE Transactions on Automatic Control* **36**, 915–924.

Elster, C. & Neumaier, A. (1995), 'A grid algorithm for bound constrained optimization of noisy functions', *IMA Journal of Numerical Analysis* **15**, 585–608.

Feller, W. (1968), *An Introduction to Probability Theory and its Applications*, John Wiley & Sons, New York.

Gill, P. E., Murray, W., Saunders, M. A. & Wright, M. H. (1983), 'Computing forward-difference intervals for numerical optimization', *SIAM Journal on Scientific and Statistical Computing* **4**, 310–321.

Glad, T. & Goldstein, A. (1977), 'Optimization of functions whose values are subject to small errors', *BIT* **17**, 160–169.

Hedlund, P. & Gustavsson, A. (1992), 'Design and evaluation of modified simplex methods having enhanced convergence ability', *Analytica Chimica Acta* **259**, 243–256.

Hooke, R. & Jeeves, T. A. (1961), 'Direct search solution of numerical and statistical problems', *Journal of Associated Computing Machinery* **8**, 212–229.

Khuri, A. I. & Cornell, J. A. (1987), *Response Surfaces: Designs and Analyses*, Marcel Dekker, New York.

Kushner, H. J. & Clark, D. S. (1978), *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York.

Lagarias, J. C., Reeds, J. A., Wright, M. H. & Wright, P. E. (1998), 'Convergence of the Nelder-Mead simplex method in low dimensions', *SIAM Journal on Optimization* **9**, 112–147.

McKinnon, K. I. M. (1998), 'Convergence of the Nelder-Mead simplex method to a nonstationary point', *SIAM Journal on Optimization* **9**, 148–158.

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

Nelder, J. A. & Mead, R. (1965), 'A simplex method for function minimization', *Computer Journal* **7**, 308–313.

Parker, L. R., Cave, M. R. & Barnes, R. M. (1985), 'Comparison of simplex algorithms', *Analytica Chimica Acta* **175**, 231–237.

Polyak, B. T. (1987), *Introduction to Optimization*, Optimization Software, Inc., Publications Division, New York.

Powell, M. J. D. (1964), 'An efficient method for finding the minimum of a function of several variables without calculating derivatives', *Computer Journal* **17**, 155–162.

Powell, M. J. D. (1970), A hybrid method for nonlinear equations, *in* P. Rabinowitz, ed., 'Numerical Methods for Nonlinear Algebraic Equations', Gordon and Breach.

Powell, M. J. D. (1994), A direct search optimization method that models the objective and constraint functions by linear interpolation, *in* 'Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico', Vol. 275, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 51–67.

Rykov, A. S. (1980), 'Simplex direct search algorithms', *Automation and Remote Control* **41**, 784–793.

Spendley, W., Hext, G. R. & Himsworth, F. R. (1962), 'Sequential application of simplex designs in optimization and evolutionary operation', *Technometrics* **4**, 441–461.

Torczon, V. (1991), 'On the convergence of the multidirectional search algorithm', *SIAM Journal on Optimization* **1**, 123–145.

Torczon, V. (1997), 'On the convergence of pattern search algorithms', *SIAM Journal on Optimiza-*

tion **7**, 1–25.

Tseng, P. (2000), 'Fortified-descent simplicial search method: a general approach', *SIAM Journal on Optimization* **10**, 269–288.

Wardi, Y. (1988), 'A stochastic steepest-descent algorithm', *Journal of Optimization Theory and Applications* **59**, 307–323.

Wardi, Y. (1990), 'Stochastic algorithms for with Armijo stepsizes for minimization of functions', *Journal of Optimization Theory and Applications* **64**, 399–417.

Yu, W. C. (1979), 'The convergence property of the simplex evolutionary techniques', *Scientia Sinica, Special issue of Mathematics* **1**, 68–77.