Pattern Search Methods for Nonlinear Optimization

Virginia Torczon

CRPC-TR95552
April 1995

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Appeared in *SIAG/OPT Views-and-News: A Forum for the SIAM Activity Group on Optimization,* No. 6, Spring 1995, pp. 7 - 11.

PATTERN SEARCH METHODS FOR NONLINEAR OPTIMIZATION

VIRGINIA TORCZON

In 1961, Hooke and Jeeves [8] coined the term *direct search*

> ...to describe sequential examination of trial solutions involving comparison of each trial solution with the "best" obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be.

In the intervening years, the term *direct search* has come to refer to any method that does not use derivatives or approximations of derivatives to solve the problem

$$\min_x f(x)\,,$$

where $x \in \mathbf{R}^n$ and $f : \mathbf{R}^n \to \mathbf{R}$. Instead the search is directed using only function values. Thus direct search properly includes such disparate methods as the pattern search method proposed by Hooke and Jeeves [8], the wildly popular genetic algorithms [7], the Nelder-Mead simplex method [11] (a Science Citation Classic), and random search methods [2].

Few of the direct search methods were conceived with any sort of convergence analysis in mind. Notable exceptions include the conjugate directions method proposed by Powell [15, 22] and the class of deformed configuration methods proposed by Rykov and his colleagues [16, 17, 18, 9].

However, recent work [20] shows that a significant subset of the direct search algorithms—a class we call *pattern search* methods in deference to Hooke and Jeeves—share a structure that makes a unified convergence analysis possible for the algorithms as originally conceived. The surprising conclusion of this analysis is that global convergence results comparable to those for line search [12] and trust region [10] globalization strategies are possible, even though gradient information is neither explicitly calculated nor approximated.

The key to the convergence analysis is that we are able to relax the conditions on accepting a step by placing stronger conditions on the step itself.

1. Generalized Pattern Search

Pattern search methods follow the general form of most optimization methods: given an initial guess at a solution x_0 and an initial choice of a step length parameter $\Delta_0 > 0$,

```
ALGORITHM 1. General Pattern Search
For $k = 0, 1, \ldots,$
  a) Check for convergence.
  b) Compute $f(x_k)$.
  c) Determine a step $s_k$ using Exploratory Moves($\Delta_k, P_k$).
  d) If $f(x_k) > f(x_k + s_k)$, then $x_{k+1} = x_k + s_k$.
     Otherwise $x_{k+1} = x_k$.
  e) Update($\Delta_k, P_k$).
```

Pattern search methods only require *simple*, as opposed to *sufficient*, decrease on the objective function. This weaker condition is possible because we require that the step be defined by Δ_k and the *pattern* P_k and we place certain mild conditions on both the Exploratory Moves and the way in which we update Δ_k to guarantee global convergence. We do not need any derivative information because we do not need to enforce the classical sufficient decrease conditions, such as the Armijo–Goldstein–Wolfe conditions used for line search methods or the fraction of Cauchy decrease or fraction of optimal decrease conditions used for trust region methods.

A pattern P_k is defined by two components, a real nonsingular *basis matrix* $B \in \mathbf{R}^{n\times n}$ and an integer *generating matrix* $C_k \in \mathbf{Z}^{n\times p}$, where $p > 2n$. In addition, the columns of C_k must contain a *core pattern* represented by $M_k \in \mathbf{M} \subset \mathbf{Z}^{n\times n}$ and its negative $-M_k$, where $\mathbf{M}$ is a finite set of nonsingular matrices (thus ensuring that C_k has full row rank).

A pattern P_k is then defined by the columns of the matrix $P_k = BC_k$. Since both B and C_k have rank n, the columns of P_k span $\mathbf{R}^n$. The steps are of the form $s_k = \Delta_k B c_k$, where $c_k \in C_k$. (We adopt this convenient abuse of notation to indicate that c_k is a column of C_k.)

We require that the Exploratory Moves satisfy two hypotheses:

Hypotheses on Exploratory Moves.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k$.
2. If $\min\{f(x_k + y),\ y \in \Delta_k B[M_k, -M_k]\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

The second hypothesis is more interesting. It suggests that if descent can be found for any one of the $2n$ steps defined by the core pattern, then the Exploratory Moves must return a step that gives simple decrease. There is no requirement that such a step must be defined by the core pattern, nor that all $2n$

steps defined by the core pattern must be evaluated, or even that the step returned give the greatest decrease possible.

Thus, a legitimate Exploratory Moves algorithm would be one that somehow "guesses" which of the steps defined by $\Delta_k P_k$ will produce simple decrease and then evaluates only that single step. At the other extreme, a legitimate Exploratory Moves algorithm would be one that evaluates all p steps defined by $\Delta_k P_k$ and returns the step that produced the least function value.

The core pattern guarantees that at least one of the $2n$ directions defined by the columns of $[M_k, -M_k]$ is a descent direction when $\nabla f(x_k) \neq 0$. Thus the Exploratory Moves algorithm must contain a safeguard to ensure that these $2n$ directions are polled if the other strategies employed do not produce a step that gives simple decrease on $f(x_k)$.

To finish our specification we give a simplification of the technique for updating the step length control parameter Δ_k. (For a full specification of the update procedures for both Δ_k and $P_k = BC_k$, see [20].)

Algorithm 2. (Simplified) Update for Δ_k.
Given $\tau = 2$, $\theta = \tau^{-1}$ and $\lambda_k \in \{\tau^0, \tau^1\}$.
a) If $f(x_k + s_k) < f(x_k)$, then $\Delta_{k+1} = \lambda_k \Delta_k$.
b) Otherwise, $\Delta_{k+1} = \theta \Delta_k$.

Here Δ_k may be reduced if and only if simple decrease has not been realized.

The general specification for pattern search methods is rich enough to capture a variety of direct search algorithms. These include coordinate search with fixed step lengths (Davidon [4] describes its use by Fermi and Metropolis to set phase shift parameters), the evolutionary operations algorithm of G.E.P. Box [1], the pattern search method of Hooke and Jeeves [8], and the multidirectional search algorithm of Dennis and Torczon [6, 19].

The general specification also leads to global convergence results. The goal of the next section is to show that pattern search methods are as robust as their proponents have long claimed and to demonstrate that the convergence analysis is comparable to that for line search and global trust region strategies.

2. The Analysis

Critical to proving global convergence of pattern search methods is recognizing the algebraic structure of the iterates, a structure that is independent of the function to be optimized. This leads to the following theorem.

Theorem 2.1. *Any iterate x_N produced by a general pattern search (Algorithm 1) can be expressed in the following form:*

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where

- *$\beta/\alpha \equiv \tau$, with $\alpha, \beta \in \mathbf{N}$ and relatively prime, and τ is as defined in the algorithm for updating Δ_k (Algorithm 2),*
- *r_{LB} and r_{UB}* depend on *N,*
- *$z_k \in \mathbf{Z}^n$, $k = 0, \ldots, N-1$.*

The import of this theorem is that all the iterates lie on a scaled, translated integer lattice. The basis depends on the initial choice of Δ_0 and the basis matrix B, the translation depends on the initial choice of x_0, and the scaling is based solely on the sequence of updates that have been applied to Δ_k, for $k = 0, \ldots, N-1$. (See [20] for a proof.)

With this theorem in hand, it is then possible to prove the following theorem regarding the global convergence behavior of pattern search methods.

Theorem 2.2. *Assume that $L(x_0) = \{x : f(x) \leq f(x_0)\}$ is compact and that $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuously differentiable on $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the general pattern search (Algorithm 1),*

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0\,.$$

There are three key points to the proof [20]. First, it is straightforward to show that pattern search methods are descent methods. Second, it is possible to prove that pattern search methods are gradient-related methods (as defined in [13]). The third and final part of the argument involves a proof by contradiction to show that the algorithm cannot terminate prematurely due to inadequate step length control mechanisms.

The proof that pattern search methods are descent methods uses differentiability, the core pattern, the Hypotheses on the Exploratory Moves and the update rules for Δ_k. The n columns of BM_k form a set that spans $\mathbf{R}^n$ so that if $\nabla f(x_k) \neq 0$, then at least one of the $2n$ directions defined by the columns of $B[M_k, -M_k]$ (the core pattern) must be a descent direction from the current iterate. The Hypotheses on the Exploratory Moves require that in the worst case, if we have not already found a step that produces simple decrease, then we must look at all $2n$ steps defined by $\Delta_k B[M_k, -M_k]$. The update for Δ_k specifies that Δ_k must be reduced if the Exploratory Moves failed to produce a step giving simple decrease. Thus we have a backtracking line search along $2n$ search directions, at least one of which is a direction of descent. It is then a simple matter to show that this process must terminate in a finite number of iterations.

To show that pattern search methods are gradient-related methods, we prove that the following holds for any $x \neq 0$:

$$\max \left\{ \frac{|x^T s_k^i|}{\|x\| \|s_k^i\|}, \ i = 1, \ldots, p \right\} \geq \xi > 0,$$

with

$$\xi = \min_{M \in \mathbf{M}} \left\{ \frac{1}{\kappa(BM)\sqrt{n}} \right\},$$

where $\kappa(BM)$ denotes the condition number of the matrix BM. (All norms are the Euclidean vector norms.) Again we make use of the core pattern defined by $B[M_k, -M_k]$, of the fact that $M_k \in \mathbf{M}$ where $\mathbf{M}$ is a finite set, and of the fact that the Hypotheses on Exploratory Moves require that all steps satisfy $x_k \in \Delta_k P_k$.

The most delicate part of the analysis involves assuring that the common pathologies that require step length control mechanisms cannot occur because of the structure placed on the choice of iterates for pattern search methods. The problem of steps that are either too long, relative to the amount of decrease realized by the next iterate, or too short, relative to the amount of decrease predicted by the gradient at the current iterate [5], cannot occur. Because the iterates lie on a lattice, which depends on Δ_k, steps of arbitrary lengths along arbitrary search directions are not possible. Thus such pathologies cannot occur. Details of the proof can be found in [20].

Proofs of global convergence for individual pattern search methods have appeared in the literature over the years. The text by Céa [3] contains a proof of convergence for the pattern search method of Hooke and Jeeves while the text by Polak [14] contains a proof of convergence for coordinate search with fixed step length. These two proofs require that the step sizes be monotonically decreasing. Yu Wen-ci [21] gives a unified analysis for a class of direct search techniques, but it requires both that the step sizes be monotonically decreasing and that an "error-controlling" sequence, which plays the role of a sufficient decrease condition, be introduced into the algorithms considered—with no suggestions on how such a sequence could be constructed in practice. As we have demonstrated, neither restriction is necessary to obtain global convergence results for pattern search methods.

REFERENCES

[1] G. E. P. Box, *Evolutionary operation: A method for increasing industrial productivity*, Applied Statistics, 6 (1957), pp. 81–101.

[2] M. J. Box, D. Davies, and W. H. Swann, *Non-Linear Optimization Techniques*, ICI Monograph No. 5, Oliver & Boyd, Edinburgh, 1969.

[3] J. Céa, *Optimisation : théorie et algorithmes*, Dunod, Paris, 1971.

[4] W. C. Davidon, *A belated preface for ANL 5990*, SIAM J. Optimization, 1 (1991), p. 1.

[5] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] J. E. Dennis, Jr. and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optimization, 1 (1991), pp. 448–474.

[7] J. H. Holland, *Genetic algorithms*, Scientific American, (1992), pp. 66–72.

[8] R. Hooke and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, J. Assoc. Comput. Mach., 8 (1961), pp. 212–229.

[9] A. G. Kuznetsov, *Nonlinear optimization toolbox*, Tech. Report OUEL 1936/92, Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K., 1992.

[10] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming, The State of the Art, A. Bachem, M. Grotschel, and G. Korte, eds., Springer–Verlag, 1983, pp. 256–287.

[11] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[12] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numerica, 1 (1992), pp. 199–242.

[13] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[14] E. Polak, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, 1971.

[15] M. J. D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Comput. J., 7 (1964), pp. 155–162.

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

[17] ———, *Simplex methods of direct search*, Engineering Cybernetics, 18 (1980), pp. 12–18.

[18] ———, *Simplex algorithms for unconstrained minimization*, Problems of Control and Information Theory, 12 (1983), pp. 195–208.

[19] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM J. Optimization, 1 (1991), pp. 123–145.

[20] ———, *On the convergence of pattern search algorithms*, Tech. Report 93–10, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251–1892, 1993. In revision; submitted to *SIAM Journal on Optimization.*

[21] Y. Wen-ci, *Positive basis and a class of direct search techniques*, Scientia Sinica, Special Issue of Mathematics, 1 (1979), pp. 53–67.

[22] W. I. Zangwill, *Minimizing a function without calculating derivatives*, Comput. J., 10 (1967), pp. 293–296.