Short Notes

Randomized Pattern Search

J. P. LAWRENCE, III, MEMBER, IEEE, AND
KENNETH STEIGLITZ, MEMBER, IEEE

Abstract—A random search technique for function minimization is proposed that incorporates the step-size and direction adaptivity of Hooke and Jeeves' [1] pattern search. Experimental results for a variety of functions indicate that the random pattern search is more effective than the corresponding deterministic method for a class of problems with hard constraints.

Index Terms—Direct search, optimization algorithms, random search.

I. Introduction

In 1961, Hooke and Jeeves [1] proposed a direct search method for function minimization that has step-size and direction adaptivity. The algorithm is widely known [2]–[4], and has met with some success on difficult minimization problems. However, the method is known to run into difficulties at sharp corners and curving valleys or ridges because of the univariate character of its basic exploratory search strategy [3]. As will be seen, constrained problems that require movement along a boundary are especially troublesome, and cause the deterministic pattern search (DPS) of Hooke and Jeeves to get stuck on the boundary. The purpose of this note is to propose a randomized version of pattern search (RPS), which, to some extent, avoids these difficulties. Considered as a random search method, RPS incorporates both the direction and step-size adaptivity of DPS, and is of interest in this regard as an extension of previously proposed methods (see [5]–[10], for example).

II. Description of Deterministic Pattern Search (DPS)

DPS performs an exploratory univariate search at the base point ϕ as follows: the function F to be minimized is evaluated at the base point ϕ and at the point $\phi+\Delta e_1$, where e_1 is a unit vector in the first coordinate direction, and Δ is a step-size parameter that is reduced in an adaptive way as the search progresses. If $F(\phi+\Delta e_1)<F(\phi)$, the point $\phi+\Delta e_1$ is adopted as a new, improved point. Otherwise, $F(\phi-\Delta e_1)$ is examined, and if $F(\phi-\Delta e_1)<F(\phi)$, the point $\phi-\Delta e_1$ is adopted. This procedure is repeated sequentially for each coordinate, yielding, possibly, a new base point at which F has decreased. Acceleration steps are then performed on the basis of successful steps from base point to base point, with exploratory univariate searches performed at each projected tentative base. When no improvements can be found in this way, the step-size Δ is decreased to $\rho\Delta$, $\rho<1$; and the process is repeated. (See [1] for a complete description.)

Fig. 1 gives the definitions of the variables used in the program and Fig. 2 shows a flow chart[1] of the algorithm MAIN. This algorithm uses a subroutine EXPLOR (DETERMINISTIC), shown in Fig. 3, which performs the univariate search with base ϕ and step-size Δ. Algorithm MAIN terminates when the function falls below a minimum, FMIN, or when the number of function evaluations exceeds a maximum, LIMIT. Whenever the step-size falls below a minimum, δ, the step-size Δ is reset to its initial value Δ_0. This feature was introduced to give both DPS and RPS the opportunity to continue its search with a large step-size after the step-size has collapsed to below δ in a difficult situation.

III. Description of Random Pattern Search (RPS)

In the proposed algorithm the local exploration at each projected base ϕ is performed by generating a random vector, ξ, uniformly distributed on a hypersphere of radius Δ. The function is then evaluated at ϕ and at $\phi+\xi$. If $F(\phi+\xi)\geq F(\phi)$, then a reversal step is taken and $F(\phi-\xi)$ is evaluated. The new search base point is redefined as before. The procedure is repeated once for each dimension of the space during each exploratory search, so that the average number of function evaluations per base point will be the same as for the corresponding deterministic method.

The algorithm MAIN is unchanged for the random search method, and the subroutine EXPLOR (DETERMINISTIC) is replaced by the algorithm EXPLOR (RANDOM), shown in Fig. 4. The random vector ξ uniform on a hypersphere is generated by a method described by Knuth [11].

IV. Experimental Results

In all the examples described below, $\delta=1.\text{E}-7$, FMIN $=1.\text{E}-4$, LIMIT$=1500$, $\Delta_0=1$, and $\rho=\frac{1}{2}$.

Example 1: This problem illustrates clearly the difference in behavior between DPS and RPS at hard constraint boundaries. It is the following linear program in two variables:

$$\begin{aligned} \text{minimize} \quad & F = -4x_1 - 2x_2 + 3 \\ \text{subject to} \quad & x_1 - x_2 \leq 0 \\ & x_1 + x_2 \leq 1 \end{aligned}$$

with solution $x_1=x_2=0.5$, $F=0$. The following starting point was used (integer lattice points unfairly favor DPS when the step size is integral):

Manuscript received July 24, 1970; revised April, 1971. This work was supported by the U. S. Army Research Office, Durham, N. C. under Contract DAHCO4-69-C-0012 and NSF Grant GJ-965.

J. P. Lawrence, III, is a part-time student in the Department of Electrical Engineering, University of Maryland, College Park, Md.

K. Steiglitz is with the Department of Electrical Engineering, Princeton University, Princeton, N. J.

[1] The style of programming and flow charting allows only two control statements: an *if* (condition) *then A else B* branch indicated by a simple branching, and a *while* (condition) *do A* indicated by a junction point with a cross inside a circle. The code to the right of the junction is executed so long as the condition is true; the first time the condition is violated, control continues downward. All variables are global.

DESCRIPTION OF VARIABLES

N	dimension of parameter space
Δ	step-size
δ	minimum step-size (for terminating search)
ρ	step-size reduction ratio
FMIN	minimum function value (for terminating search)
LIMIT	maximum number of function evaluations (for terminating search)
ψ	current base value of parameter vector
φ	exploratory base value of parameter vector
θ	previous base value of parameter vector
F( )	function to be minimized
Fψ	F(ψ)
Fφ	F(φ)
KOUNT	current number of function evaluations
i	current number of successful steps
FNEW	F( ) at exploratory point
j	dimension index in EXPLOR (DETERMINISTIC)
k	counting index in EXPLOR (RANDOM)
ξ	random vector uniformly distributed on sphere of radius Δ
Δ_o	initial step-size
SAVE	temporary storage location for coordinate or vector

Fig. 1. Description of variables.

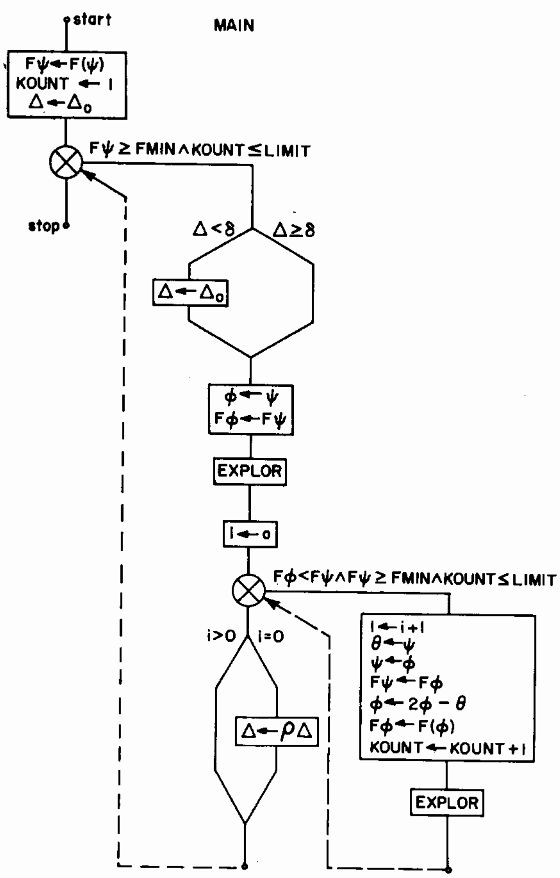

Fig. 2. MAIN.

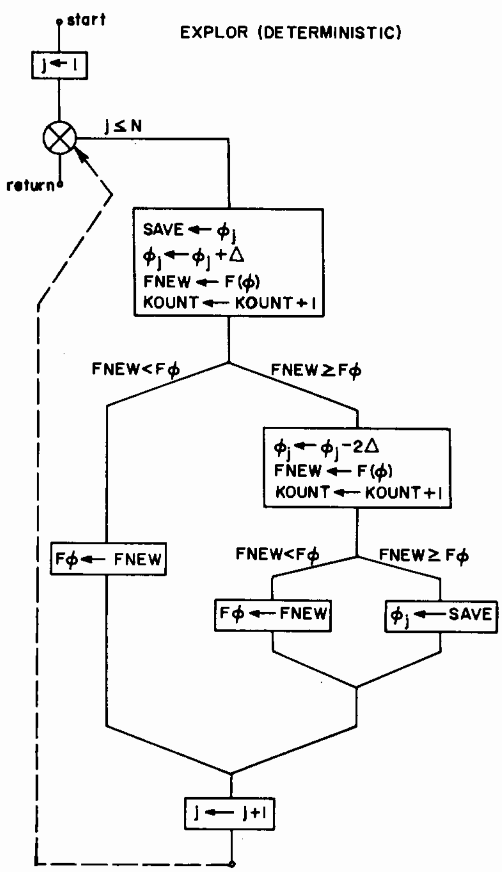

Fig. 3. EXPLOR (DETERMINISTIC).

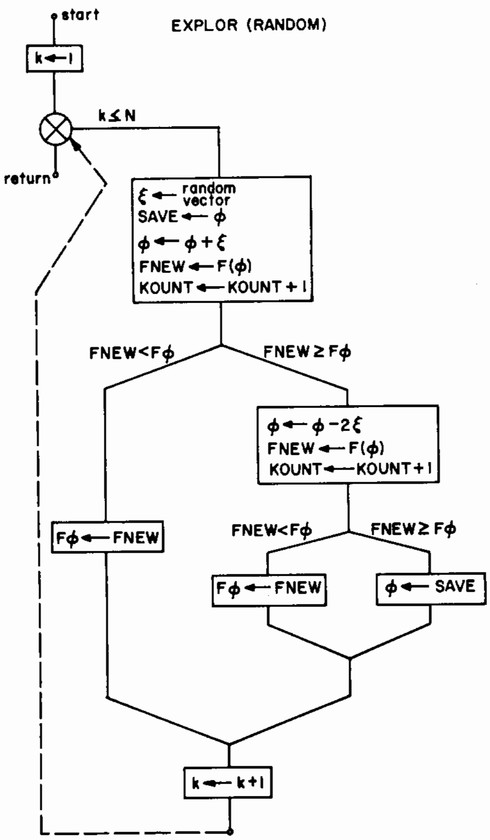

Fig. 4. EXPLOR (RANDOM).

$$x_1 = \frac{\sqrt{2}}{10}$$

$$x_2 = \frac{\sqrt{3}}{10}$$

and the constraint was enforced by defining $F=1.E+8$ outside the feasible region.

DPS, after 208 function evaluations, became frozen to the boundary at the point (0.3267959, 0.6732050), and $F \doteq 0.3464069$. RPS converged to within FMIN in from 192 to 1496 function evaluations in ten starts using different initializations of the random number generator. This wide range of evaluations-to-convergence is typical of RPS, and it is often worthwhile to start over if convergence becomes slowed down.

Example 2: This problem is identical to the previous, except for the fact that the function F is nonlinear:

$$F = -(x_1^2 + x_2^2) + 1$$

and the additional constraints x_1, $x_2 \geq 0$ have been added. The solution is $x_1=0$, $x_2=1$, $F=0$.

DPS behaved in exactly the same way, freezing to the same point after the same pattern moves (every feasible move up and to the right is an improvement in both cases). RPS converged to within FMIN in from 176 to 245 function evaluations in ten random starts.

Example 3: This problem is the following three-variable linear program:

$$\begin{aligned} &\text{minimize} && F = -x_1 + 1 \\ &\text{subject to} && x_1 + 2x_2 + 3x_3 \leq 1 \\ & && x_i \geq 0, \quad i = 1, \cdots, 3 \end{aligned}$$

and starting point

$$x_1 = \frac{\sqrt{2}}{100}$$

$$x_2 = \frac{\sqrt{3}}{100}$$

$$x_3 = \frac{\sqrt{5}}{100}.$$

The solution is at (1, 0, 0), $F=0$.

DPS froze at the boundary with $F=0.1017221$ after 288 function evaluations, while RPS converged in from 434 to 653 function evaluations in ten random starts.

Example 4: This problem is identical to the previous, except that the function to be minimized is

$$F = -(x_1^2 + x_2^2 + x_3^2) + 1$$

with the same solution.

As before, DPS went through the same sequence of decisions as in the linear program. RPS, on the other hand, behaved differently from the linear case; converging to the solution six times out of ten random starts in from 477 to 817 function evaluations. In one of the remaining four starts, it converged to the local minimum at $(0, \frac{1}{2}, 0)$, and the other three times it became stuck or started to creep very slowly. Again, this variance in behavior suggests restarting RPS with different random numbers if slow convergence is encountered.

Example 5: The remaining problems are unconstrained, and are included to show that for this class of problems RPS is effective but sometimes less efficient than DPS. Thus, the price paid for more robust behavior on constrained problems is slower convergence on some of the easier unconstrained problems. Example 5 is the helical valley of Fletcher and Powell [14], which is a three-dimensional function with a corkscrew type of valley leading to a minimum at (1, 0, 0). The starting point is

$$x_1 = \sqrt{2}$$
$$x_2 = \sqrt{3}$$
$$x_3 = \sqrt{5}.$$

DPS converged in 269 function evaluations, while RPS converged in from 285 to 552 function evaluations in ten random starts.

Example 6: This is Powell's four-dimensional quartic function [13], with starting point

$$x_i = \sqrt{(i+1) \text{ rst prime}}, \qquad i = 1, \cdots, 4.$$

DPS converged in 366 function evaluations, while RPS required between 303 and 670 in ten random starts.

Example 7: This is the ten-dimensional function

$$\sum_{i=1}^{10} x_i^4$$

and starting point given by the formula in the previous example. This function is highly symmetric, and requires no radical changes in direction. Hence, we would expect DPS to excel. In fact, DPS converged in only 271 function evaluations, while RPS required from 481 to 736.

Example 8: This is Rosenbrock's banana-shaped valley in two dimensions [12]. This function has a narrow, curved valley, and from some starting points may cause difficulty with either DPS or RPS. Then random starting points were chosen uniformly within the square $x_i \leq |10|$, $i=1, 2$. Both methods converged within LIMIT = 1500 function evaluations from the first nine starting points. However, the tenth starting point, given by

$$x_1 = -0.7241213E + 1$$
$$x_2 = 0.3670560E + 1,$$

caused both methods to exceed LIMIT, at which point the runs were terminated.

V. Discussion

The linear and nonlinear programming examples show how the deterministic pattern search of Hooke and Jeeves can become paralyzed at the boundary of a hard constraint because of the univariate character of the exploratory search. Randomizing the exploratory search strategy leads to marked improvements for these problems and results in a more robust technique.

The authors have been in communication with Prof. E. J. Beltrami about this problem, and further work along these lines will be reported by him and J. P. Indusi in another paper [15].

References

[1] R. Hooke and T. A. Jeeves, "A 'direct search' solution of numerical and statistical problems," *J. Ass. Comput. Mach.*, vol. 8, pp. 212–229, Apr. 1961.
[2] D. J. Wilde, *Optimum Seeking Methods.* Englewood Cliffs, N. J.: Prentice-Hall, 1964.
[3] M. Aoki, *Introduction to Optimization Techniques.* New York: Macmillan, 1971.
[4] G. S. G. Beveridge and R. S. Schechter, *Optimization: Theory and Practice.* New York: McGraw-Hill, 1970.
[5] L. A. Rastrigan, "The convergence of the random search method in the extremal control of a many parameter system," *Automat. Remote Contr.*, vol. 24, pp. 1337–1342, 1963.
[6] M. A. Schumer and K. Steiglitz, "Adaptive step size random search," *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 270–276, June 1968.
[7] N. V. Khovanov, "Stochastic optimization of parameters by the method of variation of the search region," *Eng. Cybern.*, vol. 5, pp. 33–38, July–Aug. 1967.
[8] J. Matyas, "Random optimization," *Automat. Remote Contr.*, vol. 26, pp. 244–251, 1965.
[9] G. A. Korn and H. Kosaka, "A proposed hybrid-computer method for functional optimization," *IEEE Trans. Comput.*, vol. C-19, pp. 149–153, Feb. 1970.
[10] R. C. White, Jr., "A survey of random methods for parameter optimization," Dep. Elec. Eng., Eindhoven Univ. Tech., Eindhoven, Netherlands, TH Rep. 70-E-16.
[11] D. E. Knuth, *The Art of Computer Programming*, vol. 2. Reading, Mass.: Addison-Wesley, 1967.
[12] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *Comput. J.*, vol. 3, pp. 175–184, Oct. 1960.
[13] M. J. D. Powell, "An iterative method for finding stationary values of a function of several variables," *Comput. J.*, vol. 5, pp. 147–151, 1962.
[14] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Comput. J.*, vol. 6, pp. 163–168, July 1963.
[15] E. J. Beltrami and J. P. Indusi, "An adaptive random search algorithm for constrained minimization," submitted for publication.

Improved Procedures for Determining Diagnostic Resolution

JAMES C. BASSETT AND CHARLES R. KIME, MEMBER, IEEE

Abstract—The definition of the generalized fault table [1] is expanded to cover a representation employing more than one fault q cube per fault pattern. On the basis of this expanded definition and simple concepts from a cover algebra, a new procedure is developed for finding first- and second-order diagnostic resolutions.

Index Terms—Cover algebras, diagnostic resolution, fault diagnosis, generalized fault table, switching theory.

Manuscript received April 28, 1971; revised November 11, 1971. This work was supported in part by National Science Foundation Grant GK-3844.

The authors are with the Department of Electrical Engineering, University of Wisconsin, Madison, Wis.