A controlled random search procedure for global optimisation

W. L. Price

Department of Engineering, The University, Leicester LE1 7RH

A new random search procedure is described which, while conceptually simple and easily programmed on a minicomputer, is effective in searching for global minima of a multimodal function, with or without constraints. The procedure is compared with a global optimisation algorithm devised by Becker and Lago and the results of trials, using a variety of test problems, are given.

(Received February 1976)

1. Introduction

There is no dearth of optimisation procedures which enable one to determine the position, in n-space, of the minimum of a given function (of n variables) that is unimodal within the domain of interest. Gradient methods, such as that of Fletcher and Powell (1963), involve the evaluation of the function and its derivatives at each iteration. Two part procedures, such as those used by Cutteridge (1974) in the solution of simultaneous nonlinear equations seek to achieve high efficiency by controlled interchange between two appropriately matched routines (e.g. conjugate gradient and damped Newton–Raphson) according to their relative merits at a given stage. Such methods are powerful but require the function to be differentiable over the relevant domain. Direct methods, involving only function evaluations, are generally less efficient than gradient methods but, because they do not require the calculation of derivatives, are simpler and are applicable to the optimisation of nondifferentiable functions. Typical of the direct methods are those of Rosenbrock (1960), Hooke and Jeeves (1969), and Nelder and Mead (1965).

The problem of global optimisation of a multimodal function has received much less attention. If sufficient is known of the properties of the function to enable the approximate positions of the minima to be estimated, then exploration of the individual modes by established methods should lead to the discovery of the global minimum. Without such prior knowledge of the behaviour of the function there would appear to be no alternative to a thorough search over the domain of interest. Having found, with sufficient confidence, the approximate position of the global minimum (or minima) the unimodal methods may be used for the final refinement. A systematic search so thorough as to guarantee the discovery of the global minimum would normally require so many function evaluations as to be quite impracticable. One must be content with a limited search, systematic or random, and accept the consequent uncertainty as to whether or not the global minimum has been found.

Brooks has reviewed several random search techniques (1958). The simple random method makes a specified number of trials at points randomly selected in V, the relevant domain in n-space, and accepts, as the optimum, the trial point with lowest function value. The stratified random sampling method divides V into a specified number of subdomains of equal size and selects, at random, a trial point within each. Such methods represent the best one can do with a limited number of trials if the function itself is quasi-random—i.e. if there is little or no correlation between the values of the function at neighbouring points. In practice however it is usual for correlation to exist—even when the function is discontinuous or the variables discrete some correlation will normally be expected (as in a histogram). It is reasonable, therefore, to suppose that the efficiency of the optimisation procedure can be improved by progressively focussing the search upon those subdomains which currently tend to contain the points with lowest function values.

Such an improvement on the simple random method is provided by the optimisation routine devised by Becker and Lago (1970). Their procedure begins with a simple random search over the chosen domain, V. Instead of retaining only the point with the lowest function value, as does the simple random method, Becker and Lago retain a predetermined number of points (those with the lowest function values). If the total number of trials is sufficient the retained points tend to cluster around minima. A mode-seeking algorithm, such as that developed by Bryan *et al.* (1969), is then used to group the points into discrete clusters and to define the boundaries of subregions each embracing a cluster. The clusters are graded, by searching in each for the retained point with the lowest function value, and then rated according to the relative values of the cluster minima. The entire procedure is then repeated using as the initial search region that subdomain, defined by the mode-seeking algorithm, around the 'best' cluster. The user may choose to examine also the second-best cluster (or indeed all clusters) according to the extent of his doubt as to whether or not the global minimum will be found in the subdomain defined by the best cluster.

2. A controlled random search optimisation procedure

The new global optimisation procedure about to be described is similar to that of Becker and Lago in that it is a direct, random method which does not require the function to be differentiable or the variables to be continuous, and which is applicable in the presence of constraints. The controlled random search (CRS) procedure differs from the Becker and Lago method in that it combines the random search and mode-seeking routines into a single, continuous process. By eliminating both the separate mode-seeking algorithm and the subjective decision making implicit in the choice of clusters for further investigation, CRS provides a simple procedure which, although best used interactively, will function automatically when interactive monitoring is not convenient.

The CRS algorithm

The essential features of the algorithm are indicated in the flow diagram of **Fig. 1.** An initial search domain, V, is defined by specifying limits to the domain of each of the n variables and a predetermined number, N, of trial points are chosen at random over V, consistent with the constraints (if any). The function is evaluated at each trial and the position and function value corresponding to each point are stored in an array, A. At each iteration a new trial point, P, is selected randomly from a set of possible trial points whose positions are related to the configuration of the N points currently held in store (the way in which the set of potential trial points is defined will be explained

later). Provided that the position of P satisfies the constraints the function is evaluated at P and the function value, f_p, is compared with f_m, M being the point which has the greatest function value of the N points presently stored. If $f_p < f_m$ then M is replaced, in A, by P. If either P fails to satisfy the constraints or $f_p > f_m$ then the trial is discarded and a fresh point chosen from the potential trial set. As the algorithm proceeds the current set of N stored points tend to cluster around minima which are lower than the current value of f_m. The probability that the points ultimately converge onto the global minimum (minima) depends on the value of N, the complexity of the function, the nature of the constraints and the way in which the set of potential trial points is chosen.

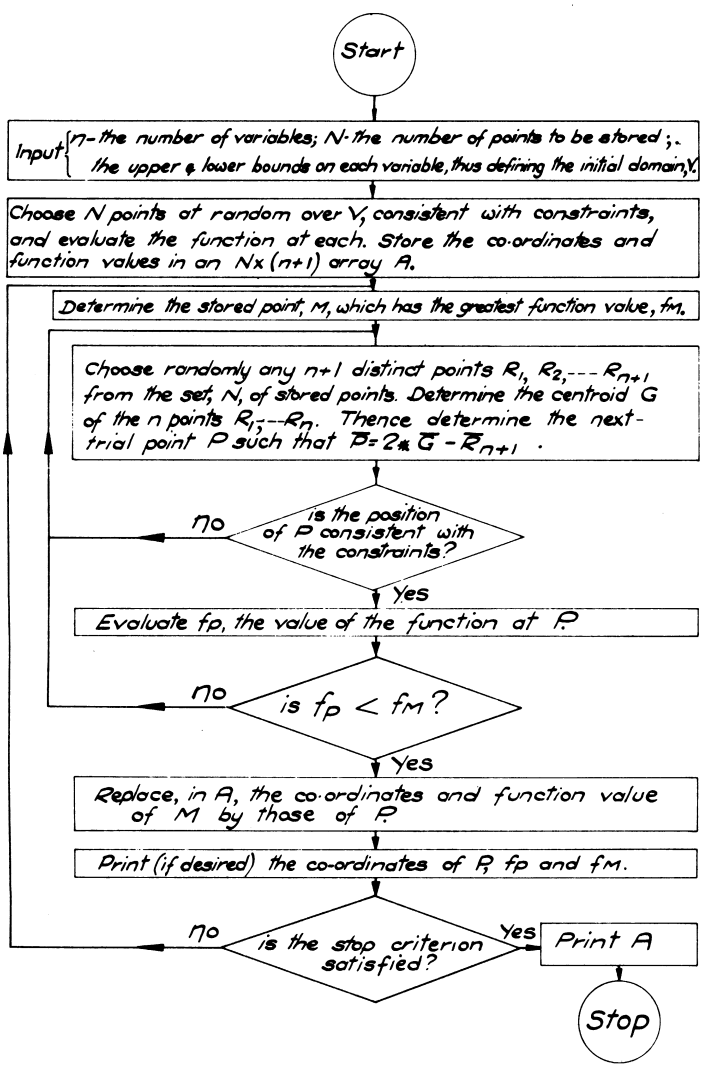

Fig. 1 Flow diagram .

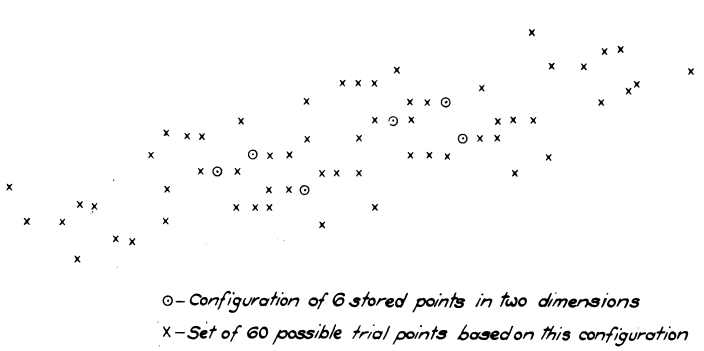

Fig. 2 An illustration of the algorithm for generating trial points

Because the procedure is intended to find global minima, thoroughness of search throughout the current subdomains is of greater importance than speed of convergence to a potential minimum. Nevertheless, if the procedure is to be more efficient than pure random search the probability of success ($f_p < f_m$) at each trial must be sufficiently high. The CRS procedure achieves a reasonable compromise between the conflicting requirements of search and convergence by defining the set of potential (next) trial points in terms of the configuration of the N points currently stored. At each iteration $n + 1$ distinct points, $R_1, R_2, \ldots, R_{n+1}$, are chosen at random from the N ($N \gg n$) in store and these constitute a simplex of points in n-space. The point R_{n+1} is taken (arbitrarily) as the pole (designated vertex) of the simplex and the next trial point, P, is defined as the image point of the pole with respect to the centroid, G, of the remaining n points. Thus

$$\bar{P} = 2 \times \bar{G} - \bar{R}_{n+1}$$

where $\bar{P}$, $\bar{G}$ and $\bar{R}_{n+1}$ represent the position vectors, in n-space, of the corresponding points. The number of different ways in which $n + 1$ points can be chosen from N is ${}^{N}C_{n+1}$ and because the choice of R_{n+1} as the pole is arbitrary (a given set of $n + 1$ points may be generated in any random order) the total number of (equiprobable) next trial points associated with the configuration of N stored points is $(n + 1) \times {}^{N}C_{n+1}$.

As a simple illustration of this principle **Fig. 2** shows an assumed configuration of six points at some stage in the search for global minima of a function of two variables. With $N = 6$ and $n = 2$ the number of possible trial points generated by this configuration is $3 \times {}^{6}C_3 = 60$. It will be observed that the disposition of this set of 60 points reflects a general trend in the current configuration and hence the random choice of any one from this set as the next trial point, P, is likely to result in a more efficient search than a procedure based on pure random search within V. On the other hand the domain of the set is not restricted to the immediate neighbourhood of the configuration. This is conducive to 'out-going' exploration, thus avoiding the dangers, for global search, of too rapid convergence.

The example also shows that while the macroscopic structure of the trial set reflects a general trend, the local variations of point density within the set correspond to detailed patterning of the current configuration. Thus the algorithm takes account of the possibility that the current configuration, regarded as two clusters each of three points, might indicate bi-modality. Further, the clusters of trial points remote from the centre of the configuration can be interpreted as 'images' of the two clusters within the configuration, each seen through the other, thus promoting the search for other possible modes which might result from periodicities of the function. The algorithm has the merits of simplicity and objectivity (it is not required to judge the modality of a given configuration) but yet demonstrates a 'reasonably intelligent' pattern recognition capability. A further attractive feature of the algorithm is that for a given N the number of potential trial points increases rapidly with n, provided that $N \gg n$. Thus high dimensionality does not, *per se*, demand inordinately large storage; it is sufficient that N should increase linearly (rather than exponentially) with n.

It might appear from the particular example of Fig. 2 that, while the algorithm is more efficient than pure random search, the probability of success ($f_p < f_m$) in a given trial is nevertheless likely to be quite small. This arises because, for simplicity of illustration, a configuration of only six points has been used. In practice one chooses $N \gg n$ and, typically, the author has taken $N = 50$ for a two dimensional search. The average success rate, i.e. the percentage of trials for which $f_p < f_m$, varies from problem to problem but in the trials conducted by the author this rate has rarely fallen below 30% and is often closer to 50% (irrespective of the number of dimensions).

Success rates of this order are, in the author's view, quite acceptable in global optimisation. It is possible to modify the algorithm in such a way as to tend to speed up the convergence, e.g. by selecting as the pole of a given simplex that point, from the set of $n + 1$, which has the largest function value (rather than choose the pole arbitrarily). But such improvements in speed of convergence are gained at the expense of thoroughness of search and thus tend to reduce the chances of finding a global minimum.

Constraints

The procedure regards all global optimisation problems as being constrained in the sense that the search is confined within the initially prescribed domain, V: if P falls outside V then the trial is discarded. When additional constraints are imposed then, dependent on the number and complexity of these constraints, a sufficiently large value of N must be chosen to ensure that a reasonable proportion of points from the potential trial set are valid (all N points of the current configuration necessarily satisfy the constraints).

The choice of N

Clearly the greater the value of N the more thorough the search and the higher the probability of discovering a global minimum. On the other hand the larger the value of N the greater is the demand on computer storage and the slower the convergence of the algorithm. In a given situation there is likely to be an optimum value of N—dependent on the complexity of the function to be optimised, the nature of the constraints and the volume of the initial search domain—such that the reduction in speed which results from a further increase in N is not justified by the gain in performance. The appropriate choice of N is therefore a matter of experience but when in doubt it is wise to make N large on the assumption that thoroughness of search takes precedence over speed of convergence.

The stop criterion

The flow diagram for the algorithm does not specify a particular stop criterion. The user is free to programme his own criterion, e.g. after a specified number of function evaluations or when the N points fall within a sufficiently small region of n-space.

3. Trials

The CRS procedure was programmed (in BASIC) and run either interactively on a PDP 11/20 minicomputer or in batch mode on a Cyber 72 computer. In order to compare the performance of the CRS procedure with the algorithm of Becker and Lago two of the examples given in their paper were used for initial trials.

Example 1

The function to be optimised is

$$f(x_1, x_2) = (|x_1| - 5)^2 + (|x_2| - 5)^2$$

which clearly has four minima, all with $f = 0$. To illustrate the global capacity of their algorithm Becker and Lago chose the large initial search domain V defined by $-10^7 < x_1 < +10^7$, $-10^7 < x_2 < +10^7$. For the first stage they selected 2,000 trial points but because, in effect, they used a logarithmic scaling of the variables, these points were not distributed uniformly over V. The best 25 of the 2,000 trials formed four clusters around the minima, principally within the $f \leqslant 50$ contours, and a second-stage run was performed on each of the four subdomains using 100 trial points.

The CRS algorithm was run, using $N = 50$, over the same initial domain V but *without* logarithmic scaling of the variables. In spite of this, after 4,000 function evaluations (counting failures as well as successes), the 50 current points were clustered around the four minima within the $f < 0{\cdot}1$ contours. After 5,000 evaluations function values of the order of 10^{-6} were achieved around each of the four minima.

Example 2

As an example of constrained optimisation Becker and Lago chose the following standard test problem first suggested by Beale (1967). The function to be minimised is

$$f(x_1, x_2, x_3) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 ,$$

subject to the constraints

$$x_1 \geqslant 0, x_2 \geqslant 0, x_3 \geqslant 0$$

and

$$x_1 + x_2 + 2x_3 \leqslant 3 .$$

From the nature of the constraints the solution must lie within

$$0 \leqslant x_1 \leqslant 3, 0 \leqslant x_2 \leqslant 3, 0 \leqslant x_3 \leqslant 1{\cdot}5$$

and these conditions were chosen to define the initial search domain V. (The minimum, at $x_1 = 4/3, x_2 = 7/9, x_3 = 4/9$, lies on the constraint boundary and has $f = 1/9 = 0{\cdot}1111\ldots$). Becker and Lago used a three-stage procedure and, restricting their trial points to a finite grid which corresponds to 2-decimal place accuracy in each variable, they obtained a minimum $f = 0{\cdot}111199$ in a total of 3,500 trials.

Taking the same initial search domain V, $N = 50$ and no truncation of coordinate values, the CRS algorithm achieved a 'best' point with $f = 0{\cdot}111112$ in 2,200 evaluations.

Example 3

This example was chosen to test the ability of the CRS procedure to cope with a periodic function having a large number of minima but only one global minimum:

$$f(x_1, x_2) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0{\cdot}1 \exp(-x_1^2 - x_2^2) .$$

The initial search domain, V, was chosen as $-10 < x_1, x_2 < 10$. Within V there are 49 minima, all having $f = 1$ except for the global minimum, with $f = 0{\cdot}9$, located at the origin (the exponential term has negligible influence on f in the neighbourhoods of the local minima). Clearly the choice of $N = 50$ does not permit of a thorough search over all the minima but the algorithm achieved its objective. After searching round various minima it began to focus its attention on the global minimum after 315 function evaluations and achieved $f = 0{\cdot}90022$ after 700 evaluations.

Example 4

This example (a modified form of Rosenbrock's function) was designed to assess the performance of the procedure when the minima lie within a steep-sided, nonlinear valley. The function to be minimised is

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + [6{\cdot}4(x_2 - 0{\cdot}5)^2 - x_1 - 0{\cdot}6]^2 .$$

This function has two global minima each with $f = 0$ (corresponding to the intersections of the two parabolas) and a local minimum with $f < 10^{-2}$ (where the parabolas approach without intersection). The initial search domain was defined as $-5 < x_1, x_2 < 5$, and again a value of $N = 50$ was chosen. The algorithm began by exploring all three minima until function values of the order of 10^{-2} were obtained in the neighbourhood of each but after 1,000 function evaluations the search began to concentrate on the global minima, finally abandoning the local minimum after 2,000 evaluations. By 4,000 function evaluations points having $f < 10^{-8}$ were located around each of the global minima.

The modification to the algorithm suggested in Section 2 was tested using this example. Again the procedure began by

exploring the neighbourhood of all three minima but the convergence then became too rapid and it abandoned not only the local minimum but also one of the global minima, concentrating on the other.

Example 5

The presence of exponential terms makes this 9-variable function a particularly difficult one to optimise.

$$f(x_1, x_2, \cdots, x_9) = \sum_{k=1}^{4} (\alpha_k^2 + \beta_k^2) + \gamma^2$$

where

$$\alpha_k = (1 - x_1 x_2)x_3 \{\exp [x_5(g_{1k} - g_{3k}x_7 \times 10^{-3} - g_{5k}x_8 \times 10^{-3})] - 1\} - g_{5k} + g_{4k}x_2 ,$$

$$\beta_k = (1 - x_1 x_2)x_4 \{\exp [x_6(g_{1k} - g_{2k} - g_{3k}x_7 \times 10^{-3} + g_{4k}x_9 \times 10^{-3})] - 1\} - g_{5k}x_1 + g_{4k} ,$$

$$\gamma = x_1 x_3 - x_2 x_4 ,$$

and the numerical constants g_{ik} are given by the matrix

0·485	0·752	0·869	0·982
0·369	1·254	0·703	1·455
5·2095	10·0677	22·9274	20·2153
23·3037	101·779	111·461	191·267
28·5132	111·8467	134·3884	211·4823

This function provides a 'least-sum-of-squares' approach to the solutions of a set of nine simultaneous nonlinear equations which arise in the context of transistor modelling and the primary concern, therefore, is to search for global minima for which $f = 0$. The function has been investigated by Cutteridge (1974) who reduces the dimensionality from nine to eight by using the equation $\gamma = 0$ to eliminate one variable.

The CRS procedure was run in batch mode on the Cyber 72 computer, a run being terminated if either the total number of function evaluations exceeds 30,000 or $(f_m - f_e)/(f_m + f_e) < 0{\cdot}01$, where f_m and f_e are respectively the greatest and least function values of the N points currently in store (this provides a simple, though arbitrary, measure of convergence). For the first run the initial search domain was chosen to be $0 \leqslant x_k \leqslant 10$ $k = 1, \ldots, 9$, and a storage value of $N = 200$ was taken. After 20,000 function evaluations the 200 points were clustered around the point.

$$x_1 = 0{\cdot}9,\ x_2 = 1{\cdot}0,\ x_3 = 3{\cdot}7,\ x_4 = 3{\cdot}9,\ x_5 = 7{\cdot}1,$$
$$x_6 = 8{\cdot}1,\ x_7 = 0{\cdot}4,\ x_8 = 1{\cdot}9,\ x_9 = 2{\cdot}8.$$

The minimum value of f being 22·6.

A second run, again using $N = 200$, took an initial search domain having the same volume as the first but centred on the above point. This resulted, after 30,000 function evaluations, in the 'best' point at

$$x_1 = 0{\cdot}9,\ x_2 = 0{\cdot}92,\ x_3 = 5{\cdot}2,\ x_4 = 5{\cdot}1,\ x_5 = 4{\cdot}8,$$
$$x_6 = 9{\cdot}8,\ x_7 = -1{\cdot}75,\ x_8 = 1{\cdot}1,\ x_9 = 0{\cdot}4.$$

having $f = 3{\cdot}8 \times 10^{-2}$.

The encouraging improvement shown by the second run suggested that this restart strategy (i.e. rerunning the program with no changes except to shift the centre of the initial search domain to the position of the best point of the previous run) be continued until no further significant improvement is obtained. Through a sequence of six such runs, each involving between 20,000 and 30,000 function evaluations, the function values corresponding to the best points were:

$22{\cdot}6,\ 3{\cdot}8 \times 10^{-2},\ 1{\cdot}8 \times 10^{-2},\ 6{\cdot}1 \times 10^{-3},\ 4{\cdot}7 \times 10^{-4},\ 3{\cdot}9 \times 10^{-4}$

The optimum point for the sixth run was

$$x_1 = 0{\cdot}8987,\ x_2 = 0{\cdot}9708,\ x_3 = 10{\cdot}95,\ x_4 = 10{\cdot}14,$$
$$x_5 = 3{\cdot}362,\ x_6 = 6{\cdot}952,\ x_7 = -8{\cdot}049,\ x_8 = 1{\cdot}236,$$
$$x_9 = -0{\cdot}4278.$$

Refinement, using a gradient method, has shown that this point is close to a global minimum having $f = 0$.

It is not known how many other global minima the function possesses but one occurs (Cutteridge, 1974) close to the point

$$x_1 = 0{\cdot}9,\ x_2 = 0{\cdot}45,\ x_3 = 1,\ x_4 = 2,\ x_5 = 8,$$
$$x_6 = 8,\ x_7 = 5,\ x_8 = 1,\ x_9 = 2.$$

Although this point lies within the initial search domain for the first run the procedure does not find it. By taking an initial search domain centred on this point and having a range of ± 2 in each variable, clear evidence of convergence towards the point was obtained ($f = 5{\cdot}7 \times 10^{-2}$ in 44,000 function evaluations) but it would appear that the sensitivity of the function is greater in the region of this global minimum than in the region of the minimum found by the CRS procedure. To have discovered this global minimum, with the initial search domain as defined for the first run, would have required a much larger value of N (beyond the storage capacity available to the author).

4. Conclusion

The CRS optimisation procedure, while simple and easily programmed on a small computer, is yet capable of global search with an efficiency which compares favourably with that of more sophisticated direct procedures. While high dimensionality does not, of itself, require an inordinate amount of storage it is desirable, when using CRS to optimise complicated functions of many variables, to use as large a value of N as is practicable. Where storage is at a premium one may perform several runs over various search domains either by dividing the given domain into discrete subdomains or by using a strategy such as that described. Because the CRS algorithm is designed for thoroughness of search rather than for speed of convergence it is desirable, where possible, to refine the discovered optima by recourse to faster, unimodal methods (e.g. by gradient methods if the function is differentiable).

References

BEALE, E. M. L. (1967). Numerical Methods, *Nonlinear Programming*, ed. J. Abadie, North Holland Publishing Company, Amsterdam, p. 150

BECKER, R. W. and LAGO, G. V. (1970). A Global Optimisation Algorithm, Proceedings of the Eighth Allerton Conference on Circuits and Systems Theory

BROOKS, S. H. (1958). A Discussion of Random Methods for Seeking Maxima, *Operations Research*, Vol. 6, pp. 244-251

BRYAN, J. K., DWYER, S. J., and LAGO, G. V. (1969). Non-Parametric Decision Schemes for EEG Classification, Sixth Annual Rocky Mountain Bio Engineering Symposium Conference Record, pp. 56-61, University of Wyoming, Laromie, Wyoming

CUTTERIDGE, O. P. D. (1974). Powerful 2—Part Program for Solution of Nonlinear Simultaneous Equations, *Electronics Letters*, Vol. 10, No. 10, pp. 182-184

FLETCHER, R. and POWELL, M. J. D. (1963). A Rapidly Convergent Descent Method for Minimisation, *The Computer Journal*, Vol. 6, pp. 163-168

HOOKE, R. and JEEVES, T. A. (1969). Direct Search Solution of Numerical and Statistical Problems, *JACM*, Vol. 8, pp. 212-229

NELDER, J. A. and MEAD, R. (1965). A Simplex Method for Function Minimization, *The Computer Journal*, Vol. 7, pp. 308-313

ROSENBROCK, H. H. (1960). An Automatic Method for Finding the Greatest or Least Value of a Function, *The Computer Journal*, Vol. 3, pp. 175-184