A survey of random methods for parameter optimization

Citation for published version (APA):

Document status and date:
Published: 01/01/1970

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
A SURVEY OF RANDOM METHODS FOR PARAMETER OPTIMIZATION

by

Dr. R.C. White, Jr.
A SURVEY OF RANDOM METHODS FOR PARAMETER OPTIMIZATION

by

Dr. R.C. White, Jr.

TH-Report 70 - E - 16
Summary
A class of algorithms known as random search methods has been developed for obtaining solutions to parameter optimization problems. This paper provides a guide to the literature in this area, while describing some of the theoretical results obtained as well as the development of practical algorithms. Included are brief descriptions of the problems associated with inequality constraints, noisy measurements, and the location of the global optimum. An attempt is made to indicate types of problems for which random search methods are especially attractive.

Contents
1. Introduction
2. Some theoretical results for random search
   a) pure random search
   b) creeping random search
3. Practical algorithms and applications
   a) some modifications of the basic creeping random search
   b) control of step size
   c) directional adaptation
4. The global optimum, noisy measurements, and constraints
   a) locating the global optimum
   b) noisy measurements
   c) inequality constraints
5. Discussion
1. Introduction

The fields of optimum system design, optimal control, and system identification have stimulated a great deal of research in the area of parameter optimization - the problem of finding a set of parameters, \( x = (x_1, x_2, \ldots, x_n)^t \), which minimizes (or maximizes) a function \( F(x) \). Many types of algorithms have been devised (e.g., steepest descent, conjugate-direction methods, pattern search), and the worth of an algorithm is judged in terms of its effectiveness in minimizing difficult functions and its economy in the use of evaluations of \( F(x) \) - usually the most time consuming operation of an algorithm. Although there are several recent books and review articles which discuss parameter optimization algorithms [1-9], they have, with some exceptions [8,9], largely neglected a group of techniques known as random search methods, which have proved effective in solving many optimization problems. This paper reviews the random search methods, indicates situations where they may be of special value, and provides a guide to the literature.

The early development of random search optimization was motivated mainly by the need for methods which were simple to program and effective in irregular parameter landscapes. Before the availability of true analog-digital hybrid computers, simple random search algorithms could be implemented by hard-wired optimizers attached to analog machines. Random search algorithms have still found use with modern hybrid computers. The complex, nonlinear dynamic systems which are most advantageously simulated on analog machines often have parameter landscapes with the sharp ridges, discontinuous first derivatives, etc., which can cause deterministic algorithms to become inefficient or to fail. Also the noisy environment of the analog machine can decrease the effectiveness of mathematically sophisticated algorithms. This is not to say that random search methods are limited to hybrid applications. There is evidence to suggest that random methods are superior in optimizing smooth functions of many variables.

Formal definitions of the parameter optimization problem and related mathematical concepts are given in References [1-7]. The notation to be used here is introduced in the following problem statement.
Determine the values of the ordered set of n parameters $x = (x_1, x_2, ..., x_n)^t$ which optimize (minimize or maximize) the criterion function

$$F(x)$$

subject to the m inequality constraints

$$g_i(x) \geq 0 \quad (i = 1, ..., m)$$

($F$ and $g_i$ are scalar functions). The set of all $x$ satisfying the constraints (2) defines the feasible region $R$. For some problems the constraints are not present or may effectively be eliminated (unconstrained optimization). The solution to the parameter optimization problem is denoted by $(x^*, F^*)$ where $x^*$ is the optimal $x$ and $F^* = F(x^*)$. For convenience all problems here are considered as minimization problems. Figure 1 illustrates the ideas introduced here.

For engineering purposes it is important to realize that the problem outlined above is only a formal framework by means of which a "real world" problem can be made amenable to solution. The engineer may be primarily interested in finding a value of $x$ such that $|F(x) - F^*|$ is small, and is not so concerned with knowing $x^*$ exactly (e.g., on-line adjustment of parameters in control system optimizations). On the other hand, in the estimation of the parameters of a system it is important that $|x_i - x_i^*|$ be as small as possible. Another consideration is whether or not the value $F^*$ is known a priori. In general the most difficult problem is that of minimizing $|x_i - x_i^*|$ as well as $F(x)$ when $F^*$ is not known a priori. These factors, which determine the goal of the optimization, must be considered in the design and/or evaluation of an algorithm.

Most of the techniques discussed here are designed to find a local minimum of $F(x)$ (a point $x^*$ such that $F(x^*) < F(x)$ for all $x$ in some neighborhood of $x^*$) for problems with no constraints on $x$ and where the measurements of $F(x)$ are noise-free. The problems of noisy measurements, inequality constraints, and the location of the global optimum are discussed briefly in Section 4.
2. Some theoretical results for random search

a) Pure random search

The pure random search method, proposed by Brooks [12] and discussed by other authors [13-16], consists of measuring $F(x)$ at $N$ random points selected from a probability distribution uniform over the entire parameter space and taking the point with the smallest value of $F$ as an approximation to the minimum. If we assume that each parameter can vary between zero and 100 per cent and that $x^*$ is to be located within 10 per cent for each parameter, then the probability of locating the optimum in $N$ trials is [15]

$$P = 1 - (1 - 10^{-n})^N \approx N10^{-n} \text{ for } 10^N >> N \quad (3)$$

Conversely, the number of trials required to have a probability 0.9 of locating the minimum is [14]

$$N = 2.3 \times 10^N$$

According to Korn [15] we are "looking for a needle in an n-dimensional haystack". Such a large number of trials obviates the use of pure random search for locating $x^*$, but in the absence of any information regarding the location of the optimum, it may be useful in choosing a starting point for a sequential search algorithm.

For the minimization of

$$F(x) = \sum_{i=1}^{n} x_i^2$$

where $|x| < \rho$, Schumer [16] found that if a total number $N$ of function evaluations may be expended on a pure random search and a subsequent local random search (Sec. III-2), five or six of these evaluations should be used for the pure random search in order to minimize the expected value of $F(x)$ obtained after the $N$ evaluations.

b. Creeping random search

Rastrigin [17] has studied the convergence of a simple creeping random search. Starting from a base point $x$ the criterion function is measured at $x + \Delta x$, where $\Delta x$ is a vector with fixed length (stepsize) and random direction. If $F(x + \Delta x) < F(x)$ (a "success") the base point is moved to $x + \Delta x$; otherwise the base point remains at $x$, and another random step is attempted.
Such an algorithm may be represented by

\[ x^{i+1} = x^i + \delta^i \Delta x^i \]  \quad (4)

where

\[ \delta^i = \begin{cases} 
1 & \text{if } F(x^i + \Delta x^i) < F(x^i) \quad \text{(success)} \\
0 & \text{if } F(x^i + \Delta x^i) \geq F(x^i) \quad \text{(failure)}
\end{cases} \]

Figure 2 shows typical progress of such a search in two dimensions. This algorithm was compared to a steepest descent method, where at each iteration a step of magnitude \(|\Delta x|\) is taken in the negative gradient direction. Rastrigin introduced the concept of search loss, defined as the number of criterion function evaluations required for a displacement in the negative gradient direction equal to the step length \(|\Delta x|\), or equivalently, the reciprocal of the average displacement in the negative gradient direction per function evaluation. The search loss was computed for both algorithms applied to a linear test function

\[ F(x) = \sum_{i=1}^{n} x_i \]

and a distance function

\[ F(x) = \left[ \sum_{i=1}^{n} x_i^2 \right]^{1/2} \]

For both functions it was found that as the number of parameters increased, the creeping random algorithm was superior to the steepest descent method on the basis of search loss. A similar result for the function

\[ F(x) = \sum_{i=1}^{n} x_i^2 \]

has been found [16,18].
The convergence of the creeping random method in the presence of noise has been studied by Gurin and Rastrigin [19]. For a linear criterion function, measurements were corrupted by Gaussian noise with zero mean and variance $\sigma^2$. The random search algorithm used a "testing step" of fixed length $\alpha$ and random direction. When such a testing step resulted in an improvement in the measured value of $F(x)$, a step of length $|\Delta x| > \alpha$ was taken in the same direction. The progress of this algorithm was compared to that of a steepest descent method, which used $2n$ perturbations of length $\alpha$ to determine the gradient and then took a working step of length $|\Delta x|$ in the estimated negative - gradient direction. Comparisons were made on the basis of search loss, and as a function of the number of parameters $n$ and a signal - to - noise ratio

$$\gamma = \frac{|\nabla F|}{\sigma \sqrt{2}}$$

where $\nabla F$ is the gradient of $F$.

For any fixed value of $\gamma$ search loss is a linear function of $n$ for the random method. For $\gamma = \infty$ (no noise) the gradient method has a search loss linear in $n$, but for $\gamma = 1$ the search loss is greater than $c \sqrt{n-1}$, where $c$ is a constant. For $\gamma = 1$ and $\gamma = \infty$ the random search method was superior for $n > 6$. It might be noted that a study by Brooks and Mickey [20] of a similar steepest descent algorithm in the presence of noise has shown that a minimum number of function evaluations $(n+1)$ should be expended on estimating the gradient. This alteration of the steepest descent algorithm would not change the nature of the results obtained by Gurin and Rastrigin, but would increase the value of $n$ above which the creeping random algorithms is superior.

It must be recognized that the results reviewed above were obtained for algorithms simplified so as to be amenable to analysis. In fact, a similar study [21] (without noise) using two different models of steepest - descent and random search algorithms has shown the steepest descent method to be superior for a class of criterion functions. Thus, the extension of the results to practical algorithms is unclear. But further results of Schumer and Steiglitz [16] (Sec. 3.b,) seem to indicate the superiority of creeping random search for problems with many parameters.
3. Practical Algorithms and Applications

Experiments with creeping random search on analog computers were reported as early as 1958-59 by Favreau and Franks [22] and Munson and Rubin [23]. A hard-wired creeping random optimizer, including provisions for expanding and reducing step size and correlating future trial-step directions with past successful directions, was built by Mitchell [24] and employed by Maybach [25] in the solution of optimal control problems on a fast repetitive hybrid computer. The development of true analog-digital hybrid computers has made it possible to employ more sophisticated random search strategies. In this section we describe some of the alterations to the basic creeping random algorithm and some schemes for adapting the step size and search directions to the function being minimized.

a) Some modifications of the basic creeping random search.

For the basic algorithm, Eq. (4), the steps $\Delta x$ are of fixed length and random direction. Although $\Delta x$ can be generated quickly by having each component $\Delta x_j$ of equal length and random sign, this results in only $2^N$ possible search directions, and the search may be forced to zig-zag toward the optimum. This can be avoided by choosing each $\Delta x_j$ from a probability distribution uniform on, say, $[-\Delta, \Delta]$ and normalizing the resulting $\Delta x$ to obtain the desired step size. The steps can be made random in length and direction by choosing each $\Delta x_j$ from a uniform [26,27] or a Gaussian distribution [28-30].

Another modification concerns the classification of a trial step as a success or failure. Stewart, Kavanaugh and Brocker [28] have used a creeping random search to solve a five-parameter two-point boundary value problem resulting from the Maximum Principle solution of an orbit transfer problem (For this problem $F(x) > 0$ and $F^* = 0$). Their algorithm included a threshold strategy, which requires a certain percentage change in $F(x)$ in order to have a success:

$$F(x^i) - F(x^{i+1}) > \eta F(x^i) \quad (0 < \eta < 1)$$

(5)

or

$$\frac{F(x^i) - F(x^{i+1})}{F(x^i)} > \eta$$

At the beginning of the search a relatively large improvement in $F$ is required, causing the algorithm to be selective in choosing a successful
search direction. This might be especially helpful when successful moves are used to direct future trial steps. (See Sec. 3.c. below) Later in the search, as \( F(x^i) \) approaches \( F^* \), smaller improvements are accepted. Similar success criteria could be written for more general problems.

In the same study the use of a vector-valued criterion function was introduced. Boundary conditions were to be matched for state variables representing displacement and velocity, \( x_d \) and \( x_v \), and adjoint variables, \( p \). The criterion function was defined as

\[
F = (F_d, F_v, F_p)
\]  

where each component of \( F \) is the sum of the errors in matching the boundary conditions for one class of variables. For a trial to be regarded as a success, it was required that all three components of \( F \) be reduced (the threshold strategy Eq. (5) was applied to each component). This more restrictive success criterion might be useful in avoiding a local minimum where only one or two components of \( F \) are small. Gonzalez [26] employed a vector-valued function in a Maximum-Principle optimization of the same systems solved by Maybach [25] with a scalar \( F(x) \). The number of evaluations required for convergence was reduced on the average, the most striking reductions being obtained for difficult starting points in the parameter space.

b. Control of step size

For the determination of parameter perturbations in practical optimization problems, it would seem logical to calculate the step size for each parameter \( |\Delta x_j| \) (or the variance of \( \Delta x \), for a random step-size algorithm), as a percentage of the value of \( x_j \) at the base point [22]. A constant step size can represent a very large or very small percentage change in \( x_j \) depending on the current value at the base point.

If the step size is small, a large proportion (asymptotic to 1/2) of the trial steps result in success (assuming no threshold strategy), but the average improvement in \( F \) per step is small. On the other hand a large step size results in a small ratio of successes to trial steps. On the basis of this observation several intuitive procedures for step-size adjustment have been proposed. Karnopp [31] suggests increasing \( |\Delta x| \) if an improvement occurs within two trials and decreasing \( |\Delta x| \) if none occurs within three trials. Maybach [25] reduced the step size following some number of
consecutive failures, but found that increasing the step size after consecutive successes had no significant effect on performance. Bekey et al. [29] used a constant variance of 4% of the range of each parameter. It was reported that their work and the results of a further study [32] failed to find a variance adjustment strategy yielding faster convergence than the constant variance method.

Beginning with Rastrigin's fixed step-size random search (Eq.4), Schummer and Steiglitz [18] developed an algorithm with adaptive step size. For the criterion function

\[
F(x) = \sum_{i=1}^{n} x_i^2 = \rho^2,
\]

the expected improvement per step, normalized by the present value of \( F \), was computed as a function of \( n \) and \( \eta = s/\rho \), the ratio of the step size to the distance to the optimum, i.e.,

\[
I(n, \eta) = \frac{-E(\Delta F)}{F}
\]

\( I(n, \eta) \) was maximized with respect to \( \eta \), and the optimum \( I(n) \) was evaluated for large \( n \). This led to the result that the average number of function evaluations necessary to minimize \( F \) within a fixed accuracy is asymptotically linear in \( n \). A practical algorithm, which attempts to adjust the step size to the optimum during the minimization process, was developed and compared to two deterministic algorithms, the simplicial method of Nelder and Mead [33] and a second-order Newton-Raphson method which evaluates first and second partial derivatives at each iteration. Performances were compared on the basis of the average number of function evaluations required for minimization. (First- and second- order partial derivatives were computed analytically for the Newton-Raphson algorithm, but for the comparison, calculation of these derivatives was considered equivalent to \((n+1)^2\) function evaluations.) For a quadratic function, the second-order method was superior for \( n < 78 \), but for the function

\[
F(x) = \sum_{i=1}^{n} x_i^4
\]
the adaptive random search algorithm was superior to the second order method for \( n > 2 \) and superior to the simplicial method for \( n > 10 \). The adaptive search was also tested for

\[
F = \sum_{i=1}^{n} a_i x_i^2 \quad \text{and} \quad F = \sum_{i=1}^{n} a_i x_i^4
\]

where the \( a_i \) were chosen from a probability distribution uniform on \([0,1,1]\) .

For each of these three test functions the number of function evaluations required by the adaptive random search method was proportional to \( n \).

The only other parameter optimization method for which required functions evaluations are reported to be a linear function of \( n \) is pattern search \([1,34]\).

These results indicate that creeping random search and/or pattern search might be the most efficient strategy when the number of parameters is large.

Korn and Kosako \([35]\) have successfully employed a creeping random algorithm in a 200 - parameter functional optimization problem.

c. Directional adaptation

The convergence of a creeping random search can be accelerated using information obtained from trial moves to choose the direction of future trial steps.

A simple modification for directional adaptation is absolute positive and negative biasing \([29]\) (Fig. 3). If the last step produced a success, it is used again for the next trial step, i.e. \( \Delta x^i = \Delta x^{i-1} \) (positive biasing). If the last step resulted in a failure, \( -\Delta x^{i-1} \) is used for the next trial step (negative biasing). Of course, negative biasing is not used following two successive failures, or the algorithm will loop endlessly. Also, it is wasteful to use it after the first failure following a success. Bekey et al \([29]\) reported that absolute biasing was effective in improving convergence. Stewart et al. \([28]\) used only positive biasing and found that it decreased the average number of steps required by approximately 40% compared to the search without biasing.

Directional adaptation can also be accomplished by introducing correlation between past successful steps and future random trial steps. In an algorithm employed by de Graag \([30]\) , future exploratory moves are influenced by the last successful step.

\[
\Delta x^i = a(x^i - x^k) + z^i \tag{7}
\]
where $x^k$ is the previous base point, $a > 0$, and $z^i$ is a random vector with independent, zero-mean Gaussian components (Fig. 4). Setting $a = 0.1$, as compared to $a = 0$ (no biasing), reduced by a factor of four the number of function evaluations required to solve two problems - a minimization of Rosenbrock's function $F(x_1, x_2) = 100(x_2 - x_1)^2 + (1 - x_1)^2$ from a starting point $(10, 10)$ and a four-parameter identification problem.

Matyas [36] has devised a more complex biasing scheme:

$$\Delta x^i = d^i + T^i z^i$$  \hspace{1cm} (8)

where $T^i$ is an $nxn$ matrix, the $z^i_j$ are independent and Gaussian with zero mean and unit variance, and $d^i_j$ specifies the mean of $\Delta x^i$. Adaption is accomplished by adjusting $d^i$ according to past trial steps and past successes and failures.

$$d^i = c_0 d^{i-1} + c_1 \Delta x^{i-1}$$  \hspace{1cm} (9)

where $c_0$ and $c_1$ satisfy the following conditions. If the last step $\Delta x^{i-1}$ resulted in an improvement,

$$0 < c_0 < 1, c_1 > 0, c_0 + c_1 > 1$$

otherwise

$$0 < c_0 < 1, c_1 < 0, |c_0 + c_1| < 1$$

Thus, the mean for the next trial step is weighted positively by the present mean value and weighted positively or negatively by the last trial step. The matrix $T^i$ might be used to introduce correlation between the trial step components $\Delta x^i_j$. But for a simple algorithm, $T^i$ is given by

$$T^i = b^i I$$

where $I$ is the identity matrix and $b^i$ is a scalar specifying the variance of the trial steps.
Directional adaptation has been discussed at length by Rastrigin [37], who has proposed several learning algorithms which adjust $p^i_j$, the probability of selecting a positive trial step $\Delta x^i_j$ for the $j^{th}$ parameter at the $i^{th}$ base point, as a function of past performance. Adjustment is accomplished by making $p^i_j = p^i_j(w^i_j)$, a monotonic, non-decreasing function of the memory parameter $w^i_j$. One example of Rastrigin's schemes for adjusting $w^i_j$ is the following algorithm.

$$ w^i_j = w^{i-1}_j - a \Delta x^{i-1}_j \Delta F^{i-1} \tag{10} $$

where

$$ \Delta F^{i-1} = F(x^{i-1}_j + \Delta x^{i-1}_j) - F(x^{i-1}_j) $$

and $w^i_j$ is limited by

$$ c_1 \leq w^i_j \leq c_2 $$

The adjustment of $w^i_j$ is proportional to the last change in the criterion function, the step causing this change, and a positive coefficient. For example, a positive $\Delta x^{i-1}_j$ causing an improvement ($\Delta F^{i-1} < 0$) brings about an increase in $w^i_j$ and thereby an increase in $p^i_j$, the probability of increasing $x^i_j$ at the next trial step. Rastrigin introduces other algorithms similar to Eq. (10), which allow for discarding information collected in the distant past ("forgetting") and which provide for better adaptation to the best of possible successful directions. A more complete review of this work has been written by Schumer [16].

Another technique suggested by Rastrigin is being investigated by Heydt [38]. A local search is made about an initial point $x^0$ for an improved point $x^1$. The line $x^1 - x^0$ is used to determine the axis of symmetry of an $n$-dimensional hypercone in parameter space with focus at $x^0$ (Fig. 5). The hypercone has angle $\theta$ and length $h$. $F(x)$ is measured at random points uniformly distributed inside the cone, and when an improved point $x^2$
is found, a new cone is constructed with focus at $x^2$ and an axis of symmetry defined by $x^2 - x^1$. Thus, past successes are used to determine the search direction. If an improved point is not found after some number of measurements inside a cone, $\theta$ and $h$ are increased to enlarge the search region. Such an algorithm was successful in optimizing a six-parameter satellite attitude acquisition problem, which had been solved [39] with the algorithm described by Stewart et al. [28].
4. The Global Optimum, Noisy Measurements, and Constraints

a. Locating the global optimum

In practical optimization problems it is usually important to locate the global minimum $x^*$ rather than just a local minimum. Although it is possible for a creeping random search to jump over some local minima, the strategies discussed here for accelerating the search use information about the local behaviour of the criterion function, and thus tend to descend to a local minimum. A full discussion of techniques for location the global optimum is beyond the scope of this survey. While some sophisticated techniques have been proposed [40-44], the methods are either untested or have been found to require very many functions evaluations as $n$ increases. In practice, when a local minimum $x^+$ is located, the search range may be expanded about $x^+$ in an attempt to detect a region where $F(x^+) < F(x)$ [28,29]; or local searches can be initiated from several starting points in the hope that one such search will descend to the global minimum. Information about the nature of the problem, either known a priori or made available by way of output during the optimization, might help the engineer eliminate some regions of $R$ from future consideration. Easy interaction between the operator and the system under study - by way of hybrid computation, [27,45] and/or display systems interfaced to digital systems [46] - would appear to be an aid in solving this problem.

b. Noisy measurements

Observations of the criterion function might be corrupted by noise arising from measurement techniques or from the inherent statistical nature of a problem. Noisy observations make gradient measurements difficult and can decrease the efficiency of the powerful conjugate-direction algorithms [47]. Although the design of strategies for noisy functions is a separate problem (stochastic approximation), it may be noted that random search methods - and other "direct search" methods such as pattern search or the simplicial method - are less affected by small measurements errors, because the progress of the search depends on the determination of "successes" and "failures" rather than on the accurate calculation of function differences. Also, since random search methods can have relatively little memory, a wrong move resulting from observation error affects the search for one or only a few steps. A creeping random algorithm has been used in minimizing a noisy criterion function resulting from the optimization of a system with random parameters [27].
c. Inequality Constraints

The methods reviewed here have been discussed in terms of unconstrained optimization. In many practical problems inequality constraints are present, and it is possible that the optimal point lies on or close to a constraint boundary. Techniques for using the powerful unconstrained minimization algorithms (gradient methods, conjugate-direction methods) usually involve a projection of the negative-gradient vector onto the boundaries or the construction of penalty functions inside or outside the feasible region. While these techniques have been used successfully, they increase considerably the complexity of the problem and also usually the effort required for solution. A different approach has been taken by Box [48], who began with the basic idea of the simplicial method and developed a randomized version named the "complex" algorithm. With the creeping random methods described in the previous sections, inequality constraints can be handled by restricting the trial points \( x + \Delta x \) to lie in \( R \). For small \( |\Delta x| \) the search can approach a solution \( x^* \) on a constraint boundary.
5. Discussion

This survey has attempted to bring together the results of research in the area of random methods for parameter optimization. Comparisons between the different random search algorithms— and between random and nonrandom methods—are difficult, because there is a dearth of reports describing the performance of random searches on standard test functions. It would seem desirable for future works in the area to include this type of results. For the minimization of relatively smooth unconstrained functions of several variables, the more powerful conjugate-direction algorithms are unquestionably superior. But as the number of parameters becomes large (n > 50?) random search may enjoy an advantage. Certainly the modest computational effort and storage requirements for random search become attractive as n increases and for applications where the digital computer is small or has arithmetic which is not so fast relative to the time for measurement of the criterion function (e.g., high-speed hybrid computation). The ease of handling inequality constraints with the random methods invites research into the development of creeping random algorithms for constrained problems (acceleration of the search along a constraint boundary) and comparisons with other constrained optimization techniques.
Acknowledgements
The author is grateful for the guidance of Prof. G.A. Korn of The University of Arizona, Tucson, Arizona, who directed the research project which included this study. Thanks are also extended to Prof. P. Eykhoff of Technische Hogeschool Eindhoven, Eindhoven, The Netherlands, where the author has been studying during the preparation of the paper.
References


Figure 1 An illustration of some features of a parameter optimization problem.
Figure 2 Typical progress of a creeping random search.
Figure 3 Illustration of positive and negative absolute biasing.
Figure 4 Correlation of trial steps with the last successful step.
Figure 5 Creeping random search with hyperconical search regions