A Global Search Method for Optimizing Nonlinear Systems

BRUCE E. STUCKMAN, MEMBER, IEEE

Abstract —The theory and implementation of a new global search method of optimization in n dimensions is presented, inspired by Kushner's method in one dimension. This method is meant to address optimization problems where the function has many extrema, where it may or may not be differentiable, and where it is important to reduce the number of evaluations of the function at the expense of increased computation. Comparisons are made to the performance of other global optimization techniques on a set of standard differentiable test functions. A new class of discrete-valued test functions is introduced, and the performance of the method is determined on a randomly generated set of these functions. Overall, this method contains the power of other Bayesian/sampling techniques without the need of a separate local optimization technique for improved convergence. This yields the ability for the search to operate on unknown functions which may contain one or more discrete components.

I. Introduction

The PROBLEM of finding the "best" solution to a problem is one that is of great importance to all fields of engineering and science, though the meaning of "best" is often not clear. To simplify the problem, one resorts to defining the problem in terms of a mathematical representation such that a measure of the performance is given by f, some real-valued nonlinear function of n parameters, $X=(x_1, x_2, \cdots x_n)$. The problem of finding the best then becomes a problem of maximizing or minimizing (or extremizing) the function $f(X)$. [Note: no real difference exists between minimizing and maximizing since the maximum of $f(X)$ is the minimum of $-f(X)$.] There are many ways of accomplishing the optimization analytically, yet a large class of problems exists which involves a system whose performance function is either unknown or hopelessly cumbersome. In these cases one must resort to the use of a direct search method of optimization. In general, the choice of which search method to use is common in dealing with optimization problems since so many methods exist, each with its own set of advantages and disadvantages.

When one considers the different classes of problems in which a search routine is used, probably the most difficult would be the class of problems that have many local minima. These problems require a search that is global in nature, taking into consideration the entire solution space, not simply a small part of it. This class of problems is not uncommon when dealing with actual applications since very few actual problems are without added complexity due to higher order terms and nonlinearities. Many of the problems in this class also have the additional constraint that the cost of evaluating the function is high. Thus when choosing a search routine one would be conscious of trying to minimize the number of evaluations of the function and would be willing to spend a reasonable amount of computation time in coming up with the location for the next guess. Some examples of systems that would fall in this classification are the following.

1) The Design of a Constant Voltage Reference ([1], [2]): Global search is used to minimize the deviation from the desired output voltage when subjected to the variation of certain technological parameters. The cost of a single evaluation of the function is high since a complete circuit analysis must be performed.

2) The Design of a Nonlinear Servomotor Control System [3]: Global search is used to optimize a set of meaningful parameters such as rise time, percent overshoot, and steady-state error over a set of controller coefficients. The cost of a single evaluation of the function is high since a simulation of the time response of the system is required.

For a search method to solve the problems in this class the following attributes would be useful.

a) Derivatives of the function are not required: Thus the search will operate on functions whose derivatives do not exist at particular points, functions with discrete ranges, and functions that have large flat areas ($\mathrm{grad}(f)=0$).

b) Very few free parameters, each of which can be easily chosen: It is not desirable for the search to be greatly dependent on proper choices for a variety of gains, starting points, etc. Ideally, there should be a minimum of necessary *a priori* knowledge of the function.

c) A strong database should exist that uses the past values of the function to model the function for the purposes of deciding on the "best" guess to help minimize the number of evaluations of the function.

d) The method must have the ability to look beyond the local properties of the function to establish the global maximum over a set of bounded search variables.

The set of global search techniques is small since the majority of search techniques, including the gradient-based

Manuscript received October 6, 1987; revised July 28, 1988. This paper was presented in part at the 30th Midwest Symposium on Circuits and Systems, Syracuse, NY, August 17–18, 1987.

The author is with the Department of Electrical Engineering, University of Louisville, Louisville, KY 40292.

IEEE Log Number 8823970.

techniques, are local in nature. Using the five categories presented by Dixon and Szego [4], past methods can be separated as follows:

1) nonsequential random search, this group includes Brooks [5] as well as exhaustive or grid search (an intuitive method);
2) sequential random search, contributors include Rastrigin [6], Matyas [7], Karnopp [8], Jarvis [9], Schumer and Steiglitz [10], and Baba [11];
3) clustering methods, contributors include Becker and Lago [12], Torn [13], Price [14], and Gomulka [15];
4) line/trajectory methods, contributors include Brent [16], Griewank [17], and Branin [18];
5) sampling/Bayesian methods, works include Mockus [19], Fagiuoli [20], de Baise and Frontini [21], Kushner [22], Hill and Stuckman [23], and Groch [2].

The major contribution of this work is the development of the theory and implementation of a new global search method of optimization in n dimensions, inspired by Kushner's method in one dimension. This method is meant to address optimization problems where the function has many extrema, may or may not be differentiable, and where it is important to reduce the number of evaluations of the function at the expense of increased computation.

II. The New Algorithm

In 1963, Kushner [22] proposed a search algorithm based upon the assumption that an unknown function could be modeled as a sample function of a Gaussian random process—specifically, a Brownian motion process. With this model he showed that the expected value of the unknown function $X(t)$, conditioned on all of the measurements taken, is a piecewise linear approximation of $X(t)$ itself, and that the conditional variance of the approximation is quadratic between the observation points (see Fig. 1). Utilizing these two properties, Kushner was able to calculate explicitly the location of the best point to evaluate next. The location of the next guess is obtained by finding the point with the maximum probability that $X(t)$ exceeds $^*X_n$ (the maximum $X(t)$ found after n measurements) by some positive constant E_n, or

$$\operatorname{pr}(X(t) \geqslant {}^*X_n + E_n). \tag{1}$$

Since $X(t)$ is normally distributed, this probability is given by

$$1 - \Phi\left\{({}^*X_n - E[X(t)] + E_n)/\sqrt{\operatorname{var}[X(t)]}\right\} \tag{2}$$

where Φ is the cumulative normal density function. Since Φ is a monotonically increasing function, the probability can be maximized by minimizing the simpler quantity:

$$A = ({}^*X_n - E[X(t)] + E_n)^2/\operatorname{var}[X(t)]. \tag{3}$$

From this, along with the equations for the conditional mean and variance of $X(t)$, the minimum value of A and its corresponding value of t can be found in closed form over an interval between two sample points. The values of

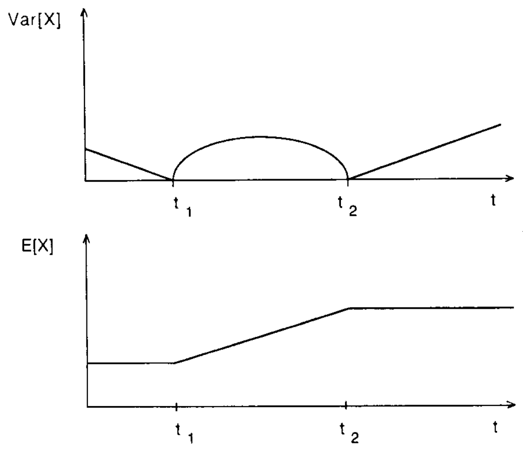


Fig. 1. Mean and variance of Brownian motion process conditioned on two observations.

$A_{\min}$ for each interval are then compared, with the minimum being the $A_{\min}$ over the entire space and with its corresponding t being the best choice for the next guess. Overall, the algorithm is as follows.

1) Evaluate the function at its endpoints.
2) Find the mean and variance for the segment between each two successive points where the function has been evaluated. (Note: initially, there is only one segment.)
3) Solve analytically for $A_{\min}$ for each segment.
4) Find the overall minimum $A_{\min}$.
5) Evaluate the function at t that corresponds to the overall minimum.
6) Form two new segments about the new measurement.
7) Go to 2).

This method performs quite well—it takes most of its guesses in areas that seem likely to contain the maximum (where the mean is high), yet the nature of the probabilities generated by the model dictate the method to venture off occasionally to guess in areas where there are few evaluations of the function (where the variance is high) in an attempt to cover the entire space. The method has attracted little attention over the last 20 years, possibly because it is strictly limited to functions of one variable. A heuristic n-dimensional model of global optimization was developed by the author [24], inspired by Kushner's method in one dimension.

To formalize the description of the problem, some problem, system, or function exists whose performance is given by $g(X)$, $X \in S$, where $g(X)$ is real valued and S is the domain to be searched such that S is a subset of E^n, the n-dimension Euclidean space, and

$$S \triangleq \text{all } X_i \text{ satisfying } b_{j1} \leqslant x_{ij} \leqslant b_{j2} \tag{4}$$

where x_{ij} is the jth component of X_i, $j = 1, 2, \cdots, n$ and where b_{j1} and b_{j2} are, respectively, the lower and upper bounds on the jth component of any X_i. Overall, we are attempting to find the global maximum $g(X^0)$ such that

$$g(X^0) \geqslant g(X), \qquad \text{for all } X \in S \tag{5}$$

An outline of the n-dimensional search strategy is as follows.

Step 1 Evaluate the unknown function at the $2n$ vertices of the n-dimensional search space.

Step 2 Construct the set of line segments that connect each set of two points where the function has been evaluated.

Step 3 Restrict the search to the line segments, and find the point for each line segment that maximizes the probability of exceeding the largest value of the function by some positive constant K_m. Find the overall maximum probability and the corresponding location.

Step 4 Evaluate the function at this point. Form new segments to this point and go to step 3).

This method assumes that the unknown function can be modeled as samples of a Brownian motion process along each possible one-dimensional (1-D) line segment passing through the search domain. Note that this model for the unknown function is not precise in a statistical sense (to do this would require a multivariate analysis of a much higher degree of complexity). Rather this model allows a convenient univariate-based search over an n-dimensional space where the 1-D trajectory to be searched is chosen from a finite set of trajectories as the one that is most likely to contain a point which will yield an improvement in the largest value of the function. This method is different from existing univariate methods since only a single point is evaluated for each chosen trajectory as opposed to an entire 1-D search, and the old trajectories as well as trajectories waiting to be chosen are all saved as possible candidates, with likelihoods being continually updated based upon what is found in other regions of the search space.

The utility of this heuristic will not be proven based upon convergence after an infinite number of evaluations of the function. Instead, its performance will be compared to other global optimization techniques on a series of standard test functions. A more in-depth description of each of the steps follows.

Step 1 Evaluate the function at the 2^n vertices of the n-dimensional space. Thus find $g(X_s)$ for all X_s such that $x_{st} = b_{t1}$ or $x_{st} = b_{t2}$ for all x_{st}, $t = 1, 2, \cdots, n$.

Step 2 Construct the set of line segments l_{st} which connect each set of two points X_s and X_t (for $s \neq t$) where the function has been evaluated.

Step 3 Restrict the search to points along the line segments by considering a given line segment l_{st}, if we consider a given Brownian motion (Wiener) process $h(\lambda)$, then

$$h(\lambda_1) - h(\lambda_2) \in N(0, c(\lambda_1 - \lambda_2)) \tag{6}$$

where c is the mean square rate of variation of the process h. Initially, we will consider $h(0) = 0$. In this case, the probability density function of $h(\lambda)$ conditioned on the known value $h(\Lambda_{st})$ is given by

$$f(h(\lambda)|h(\Lambda_{st})) = \frac{f(h(\lambda), h(\Lambda_{st}))}{f(h(\Lambda_{st}))} \tag{7}$$

since $h(\lambda)$ is a process of stationary independent increments from the properties of the Brownian motion process (see Ross [25])

$$f(h(\lambda)|h(\Lambda_{st})) = \frac{f(h(\lambda) - h(0)) f(h(\Lambda_{st}) - h(\lambda))}{f(h(\Lambda_{st}) - h(0))}. \tag{8}$$

These increments are normally distributed; therefore

$$f(h(\lambda)|h(\Lambda_{st})) = \frac{\left[\frac{1}{\sqrt{2\pi c\lambda}}\right]\exp\left[\frac{-h(\lambda)^2}{2c\lambda}\right]\left[\frac{1}{\sqrt{2\pi c(\Lambda_{st} - \lambda)}}\right]\exp\left[\frac{-(h(\Lambda_{st}) - h(\lambda))^2}{2c(\Lambda_{st} - \lambda)}\right]}{\left[\frac{1}{\sqrt{2\pi c\Lambda_{st}}}\right]\exp\left[\frac{-h(\Lambda_{st})^2}{2c\Lambda_{st}}\right]} \tag{9}$$

which yields

$$f(h(\lambda)|h(\Lambda_{st})) = \left[\frac{1}{\sqrt{2\pi c\lambda(\Lambda_{st} - \lambda)/\Lambda_{st}}}\right] \cdot \exp\left[\frac{-(h(\lambda) - \lambda h(\Lambda_{st})/\Lambda_{st})^2}{2c\lambda(\Lambda_{st} - \lambda)/\Lambda_{st}}\right]. \tag{10}$$

Therefore,

$$E(h(\lambda)|h(\Lambda_{st})) = \lambda h(\Lambda_{st})/\Lambda_{st} \tag{11}$$

and

$$\operatorname{var}(h(\lambda)|h(\Lambda_{st})) = c\lambda(\Lambda_{st} - \lambda)/\Lambda_{st}. \tag{12}$$

Now, considering the more general process $g(X) = h(\lambda) + g(X_s)$, where X is some distance λ from X_s along the segment l_{st} such that $0 \leqslant \lambda \leqslant |l_{st}|$ and $|l_{st}| = \Lambda_{st}$. The expected value of the unknown function conditioned on the known valuations of the function $g(X_s)$ and $g(X_t)$ can be described as

$$E[g(X)|g(X_s), g(X_t)] = g(X_s) + \lambda(g(X_t) - g(X_s))/\Lambda_{st}. \tag{13}$$

The conditional variance is given by

$$\operatorname{var}[g(X)|g(X_s), g(X_t)] = c\lambda(\Lambda_{st} - \lambda)/\Lambda_{st} \tag{14}$$

where c is the mean-square rate of variation of the curve along l_{st}. For each point along this line segment, we can find the probability that the unknown function will exceed $^*g_m(X)$, the largest value of the function yet found after m guesses by some positive constant K_m. This probability is given by

$$\operatorname{pr}(g(X) \geqslant {}^*g_m(X) + K_m) = 1 - \Phi\left\{({}^*g_m(X) + K_m - E[g(X)])/\sqrt{\operatorname{var}([g(X)]}\right\}. \tag{15}$$

Thus if we maximize this probability over the line segment, the location will be a possible candidate for the location of the next guess. Since Φ is a monotonically increasing function, we can maximize the probability by minimizing the simpler quantity

$$A = ({}^*g_m(X) + K_m - E[g(X)])^2/\text{var}[g(X)]. \quad (16)$$

Solving for ${}^*\lambda$, the location of $A_{\min}$ by first expanding A yields

$$A = \frac{({}^*g_m(X) + K_m - g(x_s) - \lambda(g(X_t) - g(X_s)/\Lambda_{st})}{c\lambda(\Lambda_{st} - \lambda)/\Lambda_{st}}. \quad (17)$$

Taking the derivative with respect to λ equals

$$\frac{dA}{d\lambda} = \left[2\lambda(g(X_t) - g(X_s))^2/\Lambda_{st} - (g(X_t) - g(X_s)) \cdot (g(X_s) + K_m + {}^*g_m(X))\right] c\lambda(1 - \lambda/\Lambda_{st})/\Lambda_{st}/ c^2\lambda^2(\Lambda_{st} - \lambda)/\Lambda_{st} - \left[{}^*g_m(X) + K_m - g(X_s) - \lambda(g(X_t) - g(X_s))/\Lambda_{st}\right]^2 \cdot c[1 - 2\lambda/\Lambda_{st}]/c^2\lambda^2(\Lambda_{st} - \lambda)/\Lambda_{st}; \quad (18)$$

simplifying and setting equal to zero yields

$$0 = B_2\lambda + B_1\lambda + B_0 \quad (19)$$

where

$$B_2 = c\left[(g(X_t) - g(X_s)^2 - 2(g(X_t) - g(X_s))(-g(X_s) + K_m + {}^*g_m(X)\right]/\Lambda_{st} \quad (20)$$

$$B_1 = 2c\left[{}^*g_m(X)^2 + K_m^2 + g(x_s)^2 + 2{}^*g_m(X)(K_m - g(X_s)) - 2K_m g(X_s)\right]\Lambda_{st} \quad (21)$$

$$B_0 = -c\left[{}^*g_m(X) + K_m - g(X_s)\right]^2. \quad (22)$$

The only positive solution is

$${}^*\lambda = \frac{({}^*g_m(X) + K_m - g(X_s))\Lambda_{st}}{(2{}^*g_m(X) + 2K_m - g(X_s) - g(X_t))}. \quad (23)$$

A proof that this critical point is really a minimum and not a maximum or saddle point can be derived using the second derivative test. Taking the second derivative of A with respect to ${}^*\lambda$ yields

$$\frac{dA^2}{d^2\lambda} = [2B_2\lambda + B_1]c^2\left[\lambda - \lambda^2/\Lambda_{st}\right]^2 - 2\left[\lambda - \lambda^2/\Lambda_{st}\right]\left[B_2^2\lambda^2 + B_1\lambda + B_0\right][1 - 2\lambda/\Lambda_{st}] \quad (24)$$

where B_2, B_1, and B_0 are defined by (22), (21), and (20), respectively. Evaluating at $\lambda = {}^*\lambda$, the second term goes to zero, leaving

$$\frac{dA^2}{d^2\lambda} = [2B_2{}^*\lambda + B_1]c^2\left[{}^*\lambda - {}^*\lambda^2/\Lambda_{st}\right]^2. \quad (25)$$

If we can show that this expression is positive, then (23) will yield a minimum. Note that c, ${}^*\lambda$, and Λ_{st} are all positive constants, and since ${}^*\lambda \leqslant \Lambda_{st}$, ${}^*\lambda/\Lambda_{st} \leqslant 1$. Therefore,

$$\frac{dA^2}{d^2\lambda} = [2B_2{}^*\lambda + B_1]U \quad (26)$$

where U is positive. Substituting for B_2, B_1, and ${}^*\lambda$ yields

$$\frac{dA^2}{d^2\lambda} = 2[{}^*g_m(X) + K_m - g(X_s)] \cdot [{}^*g_m(X) + K_m - g(X_t)]U. \quad (27)$$

Note that both the first and second terms are positive since ${}^*g_m(X) \geqslant g(X_s)$ and ${}^*g_m(X) \geqslant g(X_t)$. Therefore, the second derivative with respect to λ evaluated at ${}^*\lambda$ is positive, and therefore, ${}^*\lambda$ is a minimum.

Solving for the maximum value of A by substituting (23) into (17) yields

$$A_{\min} = 4(K_m + {}^*g_m(X) - g(X_s)) \cdot (K_m + {}^*g_m(X) - g(X_t))/c\Lambda_{st}. \quad (28)$$

Using these results, we can compare the values of $A_{\min}$ to find the line segment with the lowest $A_{\min}$. Equation (10) will then give us the location of the overall best candidate point over the entire set of line segments which has the highest probability of exceeding the largest value found by K_m. This point becomes the location of the next guess X_{m+1}. Notice that this can be accomplished without knowledged of c since ${}^*\lambda$ is independent of c and since c is a divisor of the $A_{\min}$ of all segments. Thus the model for the unknown function is nonparametric.

Step 4 The function is evaluated at X_{m+1}, and the process is continued iteratively as the line segment which contains the new point is subdivided into two segments, line segments are added that connect the new point to the other $m-2$ points, and the value of $A_{\min}$ is found for each of the line segments, including the newest ones.

The power of this search strategy is that the mean and the variance both contribute to the probability that the function will exceed the largest value found. Due to the linear nature of the mean, this probability will be high in regions which are close to the largest value found. However, due to the quadratic nature of the variance, areas which are relatively unexplored will have a probability which will eventually dominate. From time to time, this leads the search to venture off and evaluate the function in areas where evaluations are sparse in an attempt to find undiscovered regions of interest. This is what leads to the global properties of the method.

III. Practical Considerations of the Algorithm

In the development of an optimization technique, it is useful to have a standard by which different modifications of the technique can be compared. Therefore, most designers of such techniques have relied upon the use of a set of test functions that epitomizes the properties of an unknown function or system which the particular optimization technique was intended to address. The standard set of test functions presented by Dixon and Szego [4] serves such a purpose. However, though these functions are quite complicated, they suffer the fact that they are everywhere differentiable. This is not representative of the class of unknown functions that have one or more discrete components. This class occurs in many applications including the optimization of circuits and systems based upon discrete time simulation. To consider the more general case, it is necessary to develop a test function to supplement this set which is not everywhere differentiable. The test function designed for this use was chosen to have the following properties:

a) a set of bounded input variables as described by (4);
b) more than one local maximum to exploit the global properties of the algorithm;
c) $\mathrm{grad}(f(X)) = 0$ or does not exist for all $X \in S$ to exploit the property that derivatives of the function are not necessary; this means that the range of the function is discrete which is meant to correspond to the extreme case where the unknown function has one or more discrete components;
d) a set of such functions can be randomly generated; this will be an improvement over most test functions since more accurate results can be obtained by testing an optimization method over a large set of functions rather than a single function.

The function that was developed was the following:

$$f(X) = \begin{cases} \lfloor(\lfloor m_1\rfloor + 1/2)(\sin(a_1)/a_1)\rfloor, & 0 \leqslant x_1 \leqslant b \\ \lfloor(\lfloor m_2\rfloor + 1/2)(\sin(a_2)/a_2)\rfloor, & b < x_1 \leqslant 10 \end{cases}$$
$$\text{for } (0 \leqslant x_i \leqslant 10) \quad (29)$$

where

$$a_i = w(|x_1 - xr_{1i}| + x_2 - xr_{2i}| + \cdots + |x_n - xr_{ni}|) \quad (30)$$

for $i = 1,2$, and w is some constant, the truncation or floor function is represented by the brackets $\lfloor\ \rfloor$, and

b	uniform random variable [0,10],
m_j	uniform random variable [0,100],
xr_{11}	uniform random variable [0, b],
xr_{12}	uniform random variable [b,10],
xr_{ij}	uniform random variable [0,10],

for $(2 \leqslant i \leqslant N)$, $(j = 1,2)$.

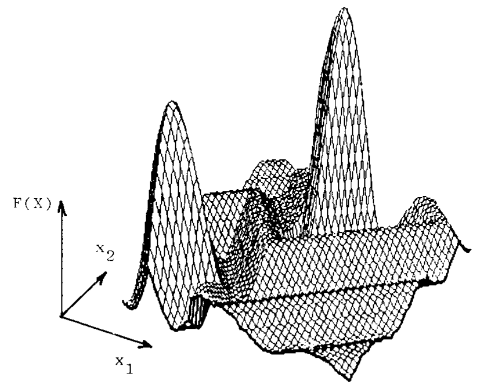
Thus the function has two major candidates for the global maximum whose positions are uniform randomly distributed in an n-dimensional space, along with many other local maxima generated by the side lobes of the sinc function. The location of the global maximum is given by

$$X^0 = \begin{cases} (xr_{11}, xr_{21} \cdots xr_{N1}), & \text{if } m_1 > m_2 \\ (xr_{12}, xr_{22} \cdots xr_{N2}), & \text{if } m_2 > m_1. \end{cases} \quad (31)$$

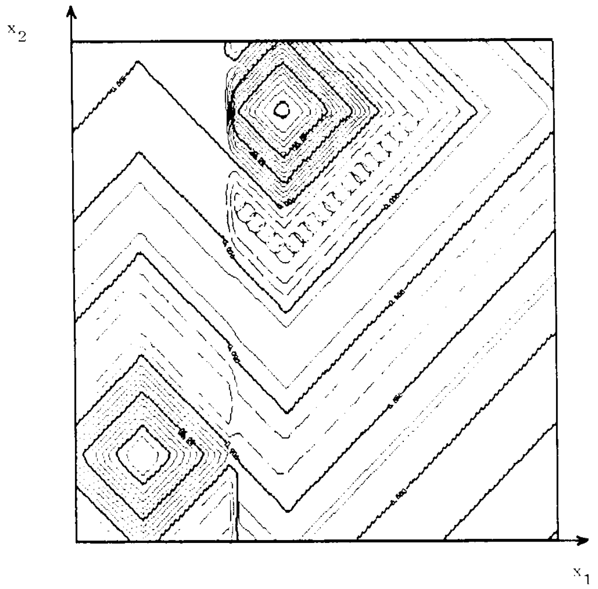
Similarly,

$$f(X^0) = \begin{cases} m_1, & \text{if } m_1 > m_2 \\ m_2, & \text{if } m_2 > m_1. \end{cases} \quad (32)$$

One such function in two dimensions is shown in Fig. 2. Note that due to the floor function used in the definition



(a)



(b)

Fig. 2. (a) Example of dual square sinc test function. (b) Contour map of example test function.

in (29) the function can only take on integer values. Therefore, the derivatives of the function are either zero or they do not exist for every point in the search space. Furthermore, there is a more pronounced discontinuity along the hypersurface $x_i = b$, the boundary between the two sinc functions.

Reducing the Computational Complexity

The motivating factor behind the development of this algorithm is the reduction in the number of evaluations of the function needed to converge upon a global extremum of an unknown function. This is to be accomplished at the expense of more computation time used to model the function based upon past observations. However, the amount of time required to perform a search cannot be completely ignored. If too much time is taken in the search, the cost of the search computation could dominate the overall cost of optimization. Therefore, the complexity of the algorithm will be considered in this regard.

Consider the computation time needed to perform a search consisting of m evaluations of the function. Due to the nature of the algorithm this time will be dominated by the speed limitations inherent in floating point arithmetic. Therefore, the computational complexity can be described in terms of the number of arithmetic operations performed in a search as a function of m, the number of points to be evaluated.

Proposition: The complexity of the algorithm given by $B(m)$ is $O(m^3)$, where m is the number of evaluations of the function.

As the algorithm is stated in Section II, after each new point is evaluated, new segments are added to each existing point in the space. Considering the worst case scenario in terms of computational complexity (when the values of the function at the new points always increase, yielding a new $*g_m(X)$ after each new point); the value of $A_{\min}$ must be recalculated for each segment after each new point. This term will dominate $B(m)$ for large m since the number of other calculations is constant for each iteration. Therefore, the number of arithmetic operations performed can be expressed as

$$B(m) \leqslant e\sum_{i=0}^{m}\binom{i}{2} = e\sum_{i=0}^{m}\frac{i^{(2)}}{2} = \frac{e(m+1)^3}{3} \tag{33}$$

where e is a suitably chosen constant. This means that the complexity of the algorithm $B(m)$ is $O(m^3)$.

This would lead to large numbers of computations for all but the smallest of problems, so the algorithm will be modified to add only a constant (Q) number of new segments beyond the subdivision of the segment containing the newest point. Since the overall mean-square variation of the unknown function is assumed to be proportional to distance along a segment, it makes sense to choose the Q segment that connect the Q points which are closest to the newest point.

Proposition: The computational complexity of this modified algorithm, given by $C(m)$, is $O(m^2)$.

The number of arithmetic operations required for the modified algorithm to perform a search which evaluates the function m times, expressed by $C(m)$, is again dominated by calculating the value of $A_{\min}$ for each of the segments. Now, however, the number of segments will grow linearly with m. Note that the number of arithmetic operations required to calculate the distance between the new point and all other points and to find the Q smallest of these distances also grows linearly with m. Therefore, $C(m)$ can be expressed as

$$C(m) \leqslant p\sum_{i=0}^{m} i = \frac{pm(m+1)}{2} \tag{34}$$

where p is a suitably chosen constant. Therefore, the complexity of the algorithm $C(m)$ is $O(m^2)$. Numerical experimentation over the set of test functions has yielded the following:

a) Q need not be large;
b) an effective way of choosing Q is given by

$$Q = \begin{cases} 10q, & \text{if the newest point yields a new maximum point} \\ q, & \text{otherwise} \end{cases} \tag{35}$$

where $q = 2n-3$, n is the dimension of the search.

Choosing the Randomness of the Search

To implement this method for optimizing some general function of n variables, one must consider how to choose the value of K_m, a free parameter of the search. If we take the limit of $*\lambda$ as K_m approaches infinity

$$\lim_{K_m\to\infty}(*\lambda) = \lim_{K_m\to\infty}\frac{(*g_m(X)+K_m-g(X_s))\Lambda_{st}}{(2*g_m(X)+2K_m-g(X_s)-g(X_t))} = \Lambda_{st}/2, \tag{36}$$

the best candidate point for a given segment approaches the point of maximum variance. This means that the search tends toward points which are farthest away from the previous guesses—in other words, unexplored areas. If we take the limit of $*\lambda$ as K_m approaches zero and if we consider the segment that contains $*g_m(X)$, if $*g_m(X) = g(X_s)$,

$$\lim_{K_m\to 0}(*\lambda) = \lim_{K_m\to 0}\frac{(*g_m(X)+K_m-g(X_s))\Lambda_{st}}{(2*g_m(X)+2K_m-g(X_s)-g(X_t))} = 0, \tag{37}$$

and if $*g_m(X) = g(X_t)$,

$$\lim_{K_m\to 0}(*\lambda) = \lim_{K_m\to 0}\frac{(*g_m(X)+K_m-g(X_s))\Lambda_{st}}{(2*g_m(X)+2K_m-g(X_s)-g(X_t))} = \Lambda_{st} \tag{38}$$

which in both cases means that the best candidate point for these segments would be the point where the maximum point in the space is found. Note that this would be the overall best candidate point for the space since the value of $A_{\min}$ would be zero. Thus a very large value of K_m will yield a more random search tending toward unexplored areas while very small values of K_m will yield a more local search in the area of the of the most promising points. In most cases, both of the extremes are to be avoided in favor of a moderate value of K_m which will yield a mixture of these two properties. Yet, the possibility exists of having K_m vary with m, the number of points evaluated. For instance, one might wish to start the search in a random mode and gradually progress to a local mode to speed up the convergence. One might also consider alternating between the predominantly random and local searches to ensure that no regions of the search space have been missed. Overall, a method that has proved to be effective in numerical experimentation with the set of test functions is as follows:

$$K_m = (f_{\max} - f_{\min})2/a + 0.0001 \tag{39}$$

where if the function is differentiable,

$$a = \begin{cases} 10, & m < 0.8n \\ 10000, & 0.8n \leqslant m \leqslant n \end{cases} \tag{40}$$

and if the function contains discrete components,

$$a = \begin{cases} 1, & m < 0.8n \\ 10000, & 0.8n \leqslant m \leqslant n. \end{cases} \tag{41}$$

Note that this implementation requires knowledge of whether the unknown function is everywhere differentiable. This is not deemed to be debilitating since the presence of discrete components in a system or function would likely be known. This has the effect of

a) compensating for the overall scaling of the function and any translations of the function values by making K_m proportional to $(f_{\max} - f_{\min})$, the largest and smallest value yet found;
b) ensuring that $K_m \neq 0$ since K_m would imply evaluating the point $^*f_m(X)$ over again (since the probability would be 1);
c) allowing the search to start in a random mode to identify the regions which look promising and switching to a more local search to converge more rapidly on a solution.

Terminating the Search

If the value of c, the mean square rate of variation of the function, was known *a priori*, a probability criterion could be used to terminate the search (i.e., stop when the probability of exceeding the maximum point found so far by K_m falls below some acceptable limit). Unfortunately, in most cases the unknown function is such that no *a priori* information of this nature is available. Thus in this implementation the value of c must be estimated based upon the evaluations of the function. The maximum likelihood estimator of the mean square rate of variation of the function is defined, after m evaluations of the function, as follows.

Consider each of the segments l_{st}. Define the differences y_i as

$$y_i = g(X_t) - g(X_s). \tag{42}$$

These differences are normally distributed:

$$y_i \in N(0, c|X_t - X_s|). \tag{43}$$

The likelihood is defined by

$$L = \sum_{i=1}^{p} f(y_i) \tag{44}$$

where p is the total number of segments after m evaluations of the function. We wish to calculate the maximum likelihood estimator for c, so we set $dL/dc = 0$. Yet this can be accomplished by simply taking the natural log. Recalling the equation for the normal distribution function

$$\frac{d\ln(L)}{dc} = \frac{1}{2cr}\sum\left[\frac{(g(X_t) - g(X_s))^2}{|X_t - X_s|}\right] - \frac{n}{2c}. \tag{45}$$

Notice that we are in no danger of $c = 0$ since this would mean that the unknown function had no variation. Solving for c yields the maximum likelihood estimator

$$\hat{c} = \frac{1}{n}\sum\left[\frac{(g(X_t) - g(X_s))^2}{|X_t - X_s|}\right]. \tag{46}$$

As a result we have the following theorem.

Theorem: The probability that the next guess will be K_m more than the largest value of the function found after m evaluations of the function can be estimated by

$$\Pr(g(X_{m+1}) \geqslant {}^*g_m(X)) \approx 1 - \Phi\left\{\sqrt{A_{\min}|_{c=\hat{c}}}\right\}. \tag{47}$$

Proof: This follows directly from (46) and (15).

This estimate can be used in terminating the search since it gives a measure of how successful it thinks the next point will be, or alternatively, how successful the search has already been. Thus terminating the search after the probability of improving the maximum on the next guess allows the user to preselect the amount of effort to be expended on determining the optimum solution to a particular problem. At termination the optimum value will be approximated by the final value of $^*f_m(X)$. As a backup to this method of termination, the maximum number of points that the user wishes to take is used to overide the probability criterion in cases where there is a need for this limitation.

IV. Implementation of the Algorithm

The method was implemented in FORTRAN on the VAX 11/780. The following are the major variables used in the program mainline:

n	number of search variables, also the dimension of the search space,
m	number of points that have already been evaluated,
$f$max	largest value of the function found after m points, or $^*g_m(X)$,
$f$min	smallest value of the function found after m points,
no	maximum number of points to be evaluated,
prob	probability that the next point to be evaluated will exceed the largest value yet found by K_m,
$t$prob	termination probability input by the user,
$f$flag	flag to indicate that a new maximum of the function has been found.

Overall, the structure of the implementation of the algorithm can be described by the pseudocode representation is as follows:

evaluate the function at the 2^n vertices of the search form segments to connect these vertices

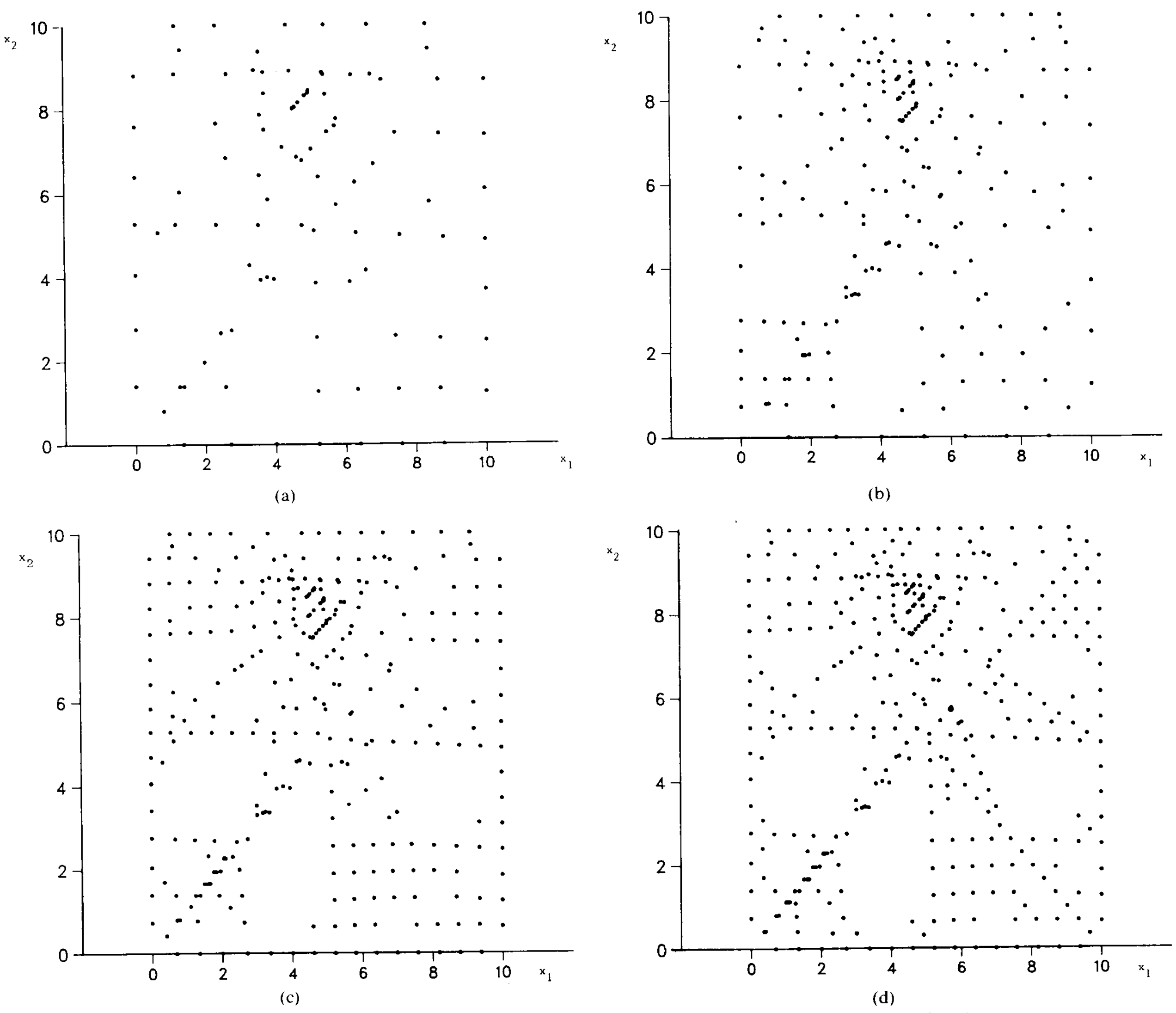


Fig. 3. (a)–(d) Scatter diagrams for first 100, 200, 300 points and entire search for example test function.

```
DO WHILE $m <$ no
    find the segment with the lowest $A_{\min}$
    determine the location of $A_{\min}$ for the best segment
    evaluate the function at this location
    IF new $f$max or new $f$min or 80 percent done
    THEN update $k$ and $A_{\min}$ for all segments
    ENDIF
    IF new $f$max
    THEN replace $Q$ with $10Q$ and set $f$flag
    ENDIF
    subdivide the best segment into two segments about the newest point
    create segments to the $Q$ closest points that are not colinear with the current segment. Find $A_{\min}$ for the new segments.
        IF $f$flag is set
        THEN replace $Q$ with $Q/10$ and reset $f$flag
        ENDIF
    IF prob $< t$prob
    THEN no $= m/0.8$ (set to 80 percent done to go into the local mode)
    ENDIF
ENDDO
global maximum $= f$max
```

Results on the Class of Nondifferentiable Test Function

This implementation presented was evaluated using the discrete dual square sinc test function presented in Section III. The search was run on one such function in two dimensions. Scatter diagrams of the progress of the search are given in Fig. 3 for the first 100, 200, 300 points, and the entire search. The global maximum of 63 was found after 368 evaluations of the function. By comparing the contour map of the function (see Fig. 2) to the scatter

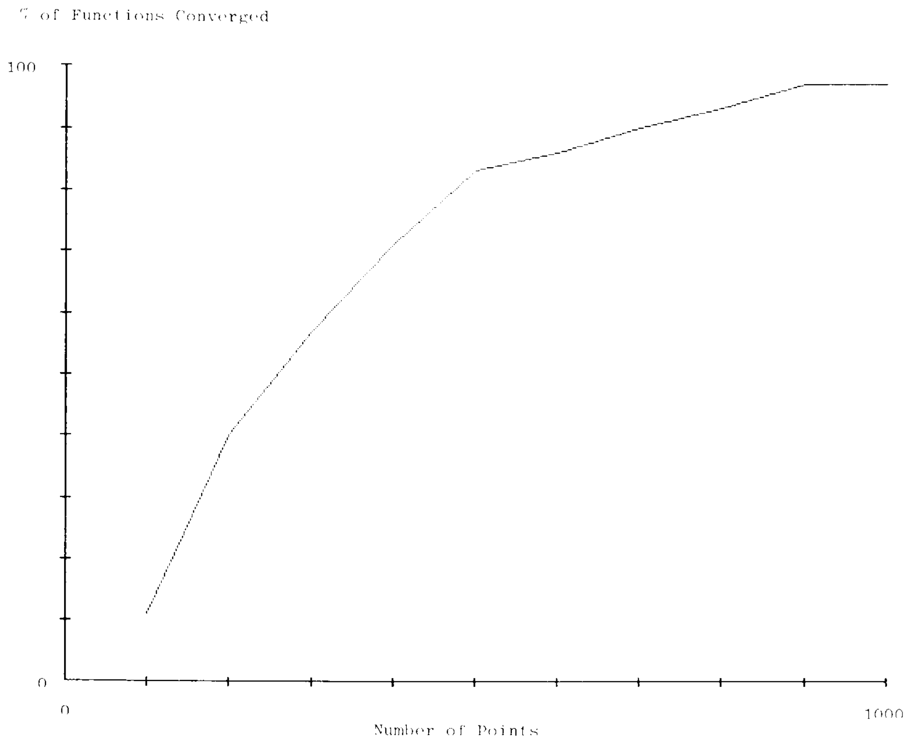


Fig. 4. Number of points needed for convergence.

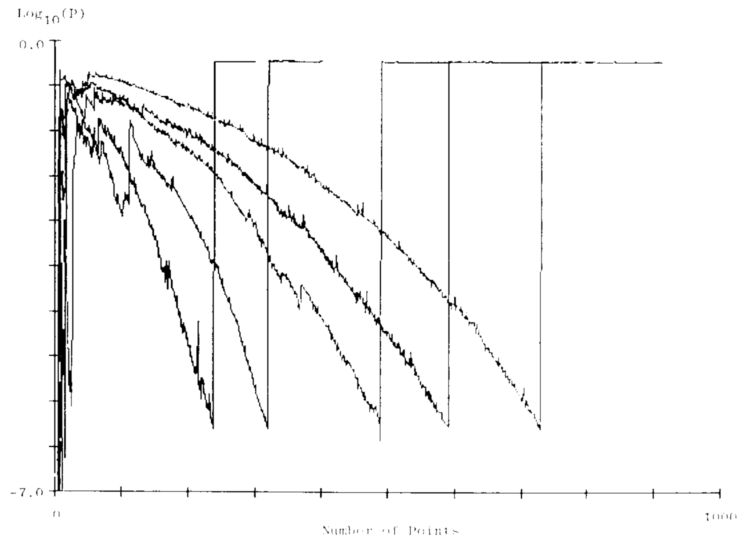


Fig. 5. Progression of probability on five dual square sinc test functions.

diagram representing the first 100 points that were taken, we can see that the search starts by showing little favoritism towards the different locations in the search space. Yet as the search progresses, in the following scatter diagrams it is evident that more points are being evaluated in locations that are "good," where the values of the function are high, as contrasted to "bad" areas, where the values of the function are low. It is also evident that these "bad" areas are not totally ignored; rather, they are only chosen less frequently (when the variance of a "bad" segment becomes relatively important enough to ensure that the maximum probability for this segment becomes larger than the maximum probability for a segment with larger mean).

The search was also tested on a set of 100 of the randomly generated two-dimensional (2-D) test functions of this form. The search found the global maximum in 97 percent of the functions after less than 1000 evaluations of the function. Note that other existing Bayesian/Sampling

TABLE I

	Functional Evaluations						
Method	G–P	RCOS	SQRN5	SQRN7	SQRN10	Hartman 3	Hartman 6
Line/Trajectory search							
Bremmermann	210*L*	250*L*	340*L*	1700*L*	2500*L*	505*L*	*L*[a]
Mod Bremmermann	300	160	375*L*	405*L*	336*L*	420	515
Zilinskas	—	5129	*L*	1212*L*	8892*L*	8641	—
Clustering							
Torn	2499	1558	3679	3606	3874	2584	3447
Gomulka/V.M.	1495	1318	6654	6084	6144	6766	11125
Price	2500	1800	3800	4900	4400	2400	7600
Sampling							
Fagiuoli	158	1600	2514	2519	2518	513	2916
De Baise-Frontini	378	599	620	788	1160	732	807
Mockus	362	189	1174	1279	1209	513	1232
Stuckman	121	494	500	500	831	93	1895

[a]An *L* indicates that the method converged on a local extrema.

TABLE II

	Computation Time (Normalized)						
Method	G–P	RCOS	SQRN5	SQRN7	SQRN10	Hartman 3	Hartman 6
Line/trajectory search							
Bremmermann	0.5	1.0	1	8	17	2	—
Mod Bremmermann	0.7	0.5	1.5	1.5	2	2	3
Zilinskas	—	80.0	—	282	214	175	—
Clustering							
Torn	4	4	10	13	15	8	16
Gomulka/V.M.	3	3	19	23	23	17	48
Price	4	4	14	20	20	8	0.5
Sampling							
Fagiuoli	0.7	5	7	9	13	5	100
De Baise Frontini	15	14	23	20	30	16	21
Mockus	—	—	—	—	—	—	—
Stuckman	0.5	5.9	17.4	17.3	34.5	0.9	136

techniques (with the exception of Hill and Stuckman [23]) have not been tested on discrete functions such as this since they incorporate local search techniques for improved convergence that require derivatives of the function. Fig. 4 presents a graph of the number of searches that have converged upon the global maximum after m points. Notice that in many cases, considerably less than 1000 points are needed. The performance of the search termination was tested using a termination probability of 10^{-6} on the first five test functions on the set. A graph of probability versus the number of points taken is given in Fig. 5. Notice that on a macroscopic scale the probability of exceeding the largest value found by K_m decreases monotonically after an initial stage of variation. After a probability of 10^{-6} is reached, the probability jumps to approximately 0.5 as the search enters the local mode to take additional points to converge upon a solution. In each case the global maximum was found with the search always terminating before 1000 points were taken. This termination probability of 10^{-6} was used on a test over the entire set of 100 test functions with the same limit of 1000 points as before. The global maximum was found for 96 percent of these functions, a one-percent reduction in performance with a significant increase in efficiency.

Results on a Set of Differentiable Test Functions

As previously mentioned, Dixon and Szego [4] have proposed that a particular set of test functions provide a basis of comparison for global optimization techniques. The criteria for how well a global optimization technique performs is based upon both how many evaluations of the function it takes to converge upon the global solution and also the amount of time it takes to reach this solution normalized to the machine being used. This normalized time to solution is defined to be the total CPU time divided by the amount of time taken to evaluate Shekel's function SQRN5 1000 times at the point (4,4,4,4) on the same machine and with the same complier and language.

The first of these test functions is the Goldstein–Price test function [26]. The function has been inverted since the search algorithm was written for maximization instead of minimization. Specifically,

$$\begin{aligned} f(X) = -&\left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2\right.\\ &\left.-14x_2+6x_1x_2+3x_2^2)\right]\\ &\cdot\left[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2\right.\\ &\left.-36x_1x_2+27x_2^2)\right] \end{aligned} \tag{48}$$

where

$$-2 \leqslant x_i \leqslant 2, \qquad i = 1, 2.$$

Properties of the Goldstein–Price test function are as follows:

- there are four local maxima;
- there is a global maxima at $(0, -1)$, with $f(X^0) = -3$;
- the function is everywhere differentiable.

The results of a search on these functions compared with the results for other practical global search algorithms (as reported by Dixon and Szego [4]) are summarized in Tables I and II.

The next function to be considered is Branin's RCOS function [18], again inverted for the purposes of maximization

$$f(x_1, x_2) = -a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1-f)\cos(x_1) + e \quad (49)$$

where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $e = 10$, $f = 1/(8\pi)$ and

$$-5 \leqslant x_1 \leqslant 10, \qquad 0 \leqslant x_2 \leqslant 15.$$

Properties of Branin's RCOS are as follows:

- it is everywhere differentiable;
- it has three maxima, all global.

The results of a search on this function compared with the results for other practical global search algorithms (as reported by Dixon and Szego [4]) are given in Tables I and II.

The third test function is actually a set of three four-dimensional test functions designated as Shekel's family of SQRN5, SQRN7, and SQRN10:

$$f(X) = \sum_{i=1}^{m} \frac{1}{(X - A_i)^T (X - A_i) + c_i} \quad (50)$$

where $X = (x_1, x_2, \cdots, x_n)$, $0 \leqslant x_i \leqslant 10$, and A_i and c_i are as given in Table III. Note that in SQRN5 $m = 5$, in SQRN7 $m = 7$, and in SQRN10 $m = 10$. These functions have the following properties:

- they are everywhere differentiable;
- they have at least m local maxima.

The results of a search on these functions compared with the results for other practical global search algorithms (as reported by Dixon and Szego [4]) are given in Tables I and II.

The last group of test functions is Hartman's family, which will be designated Hartman 3 and Hartman 6, a three-dimensional and six-dimensional test function, respectively (see Hartman [27]). When inverted for maximization, they are defined as

$$f(X) = \sum_{i=1}^{m} \left[c_i \exp - \sum_{j=1}^{n} \left[a_{ij}(x_j - p_{ij})^2 \right] \right] \quad (51)$$

and $X = (x_1, x_2, \cdots, x_n)$, $p_i = (p_{i1}, p_{i2}, \cdots, p_{in})$, $a_i = (a_{i1}, a_{i2}, \cdots)$, and where c_i is the approximate height of the ith local maximum assuming that the interactions between different local maxima are not too strong. For Hartman 3, $0 \leqslant x_i \leqslant 1$, $m = 4$, and $n = 3$, and the data are as given in Table IV.

TABLE III
DATA FOR SHEKEL'S FAMILY OF TEST FUNCTIONS

i	A_i	A_i	A_i	A_i	c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.4
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

TABLE IV
DATA FOR THE TEST FUNCTION HARTMAN 3

i		a_{ij}			c_i	p_{ij}	
1	3.0	10.0	30.0	1.0	0.3689	0.1170	0.2673
2	0.1	10.0	35.0	1.2	0.4699	0.4387	0.7470
3	3.0	10.0	30.0	3.0	0.1091	0.8732	0.5547
4	0.1	10.0	35.0	3.2	0.03815	0.5743	0.8828

For Hartman 6, $m = 4$, $n = 6$, and the data are given in Table V. The results of a search on these functions compared with the results for other practical global search algorithms (as reported by Dixon and Szego [4]) are given in Tables I and II.

From the results of the various test functions, the Bayesian/sampling methods of global optimization seem to be the most reliable in terms of minimizing the number of evaluations of the function as well as ensuring that the method converges to the global extremum of a problem. It is also evident that the method presented in this paper compares favorably with other sampling methods of global optimization even though it does not rely upon a separate local search routine for hastened convergence. This factor could be important to real-world applications where the neighborhood of the global extremum is not well behaved due to flat areas or discontinuities which would severely hamper the convergence of a local optimization technique.

V. CONCLUSION

The main goal of this method of optimization is to minimize the number of evaluations of the function needed to approximate the global extremum. Overall, the method works more efficiently than the random and nonsequential methods of optimization since it takes its guesses in areas most likely to contain the global maximum of the function, based upon a Brownian motion model for the function conditioned upon the complete set of past observations. Due to the use of all of the past evaluations of the function, this method can find an approximate solution to multipeak problems that mislead the set of local optimiza-

TABLE V
DATA FOR THE TEST FUNCTION HARTMAN 6

i	a_{ij}						c_i	p_{ij}					
1	10.0	3.0	17.0	3.5	1.7	8.0	1.0	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10.0	17.0	0.1	8.0	14.0	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3.0	3.5	1.7	10.0	17.0	8.0	3.0	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	17.0	8.0	10.0	10.0	0.1	14.0	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

tion routines since the method considers the properties of the entire search space before making its next guess.

This method of optimization falls into the category of Bayesian/sampling global optimization techniques, and its closest competitors are other methods in this category. It distinguishes itself from other such methods since, due to the formulation of the search strategy, a separate local optimization technique is not required for convergence on a solution. This has the advantage of maintaining the process of modeling the function and considering the past observations even when near the global extremum. This property can be useful if the unknown function is such that its global extremum is located in a region where the function may not be well behaved due to discontinuities, plateaus, and other deformities possibly due to the complexities of real-world problems.

Another advantage of this technique is the ability of the search to run with a minimum of user-defined parameters that plague actual implementation of other techniques. The only parameters required for the method are the maximum number of points to be taken, the dimension of the search space, knowledge of whether or not the unknown function is differentiable, and the bounds on each of the search variables.

The main limitation of this technique is that the computational complexity of the algorithm is $O(m^2)$, where m is the number of evaluations of the function. This is a necessary evil of including all past observations in the model of the function and means that large amounts of computation time may be expended on difficult problems. Thus this method is most useful when the cost of evaluating the function is high and the cost of CPU time is low. Another limitation to this technique is the large number (2^n) of evaluations of the function required to initiate the search process. For this reason it is not recommended that the search be attempted on functions of more than ten variables.

Other Applications of the Method

The applications of global optimization are many and widely varying, though one major thrust today is in the area of the design of systems. If a design process can be reduced to the selection of the values for a set of n parameters, and if there is a way of converting the performance of a trial design into a single numerical value, then optimization techniques can be used. Global optimization is needed for many problems of this type since practical system design can lead to complicated functions with many local extrema. A further contributing factor is that in many cases, the "cost" of evaluating the performance is high. This leads one to try to minimize the number of evaluations of the function which eliminates the possibility of using a trial-and-error approach with a local search routine in hopes of convergence upon a global solution.

For the global search method of optimization to work on the design of a system, the following conditions are necessary:

- the performance of the system must be a real-valued function of n variables;
- the number of search variables must be less than ten, and there must be a way of determining bounds on each variable to determine the search space.

The following is a list of conditions for which this method is the method of choice for a given optimization problem:

- the function to be optimized is multipeak;
- the amount of CPU time used is a secondary consideration when compared to minimizing the number of evaluations of the function;
- there is no guarantee that the neighborhood of the global extremum is well-behaved enough to allow a local optimization technique to converge on the actual solution once this neighborhood has been located.

Examples of optimization applied to the design of systems include pattern design for phased-array antenna systems—Stuckman and Hill [28], machine tool design—Hersom *et al.* [29], electronic filter design—Bown [30], design of a decision boundary for cluster analysis—Hill and Stuckman [23], and optical filter design—McKeown and Nag [31].

REFERENCES

[1] M. Heydemann, F. Durbin, and J. Montaron, "A tentative approach towards global optimization in the DC case," in *IEEE Proc. Int. Symp. Circuits and Systems*, Apr. 1981, pp. 847–850.

[2] A. Groch, L. M. Vidigal, and S. W. Director, "A new global optimization method for electronic circuit design," *IEEE Trans. Circuits Syst.*, vol. CAS-32, no. 2, Feb. 1985.

[3] B. E. Stuckman and N. W. Laursen, "Modern control design using global search," in *Proc. 8th Meeting Coordinating Group on Modern Control Theory*, U.S. Army, Feb. 1987.

[4] L. C. W. Dixon and G. P. Szego, "The optimisation problem: An introduction," in *Towards Global Optimisation*, L. C. W. Dixon and G. P. Szego, Eds. New York: North-Holland, 1978.

[5] S. H. Brooks, "A discussion of random methods for seeking minima," *Oper. Res.*, vol. 6, no. 2, pp. 244–251, Mar.–Apr. 1958.

[6] L. A. Rastrigin, "The convergence of the random search method in the extrema control of many-parameter system," *Automat. Remote Contr.*, vol. 24, pp. 1337–1342, 1963.

[7] J. Matyas, "Random optimization," *Automat. Remote Contr.*, vol. 26, pp. 246–253, 1965.
[8] D. C. Karnopp, "Random search techniques for optimization problems," *Automatica*, vol. 1, pp. 111–121, 1963.
[9] R. A. Jarvis, "Adaptive global search by the process of competitive evolution," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-5, no. 3, pp. 297–311, May 1975.
[10] M. A. Schumer and K. Steiglitz, "Adaptive stepsize random search," *IEEE Trans. Automat. Contr.*, vol. AC-17, pp. 1004–1007, 1972.
[11] N. Baba, "A hybrid algorithm for finding a global minimum," *Int. J. Contr.*, vol. 37, no. 5, pp. 929–942, 1983.
[12] R. W. Becker and G. V. Lago, "A global algorithm," in *Proc. 8th Allerton Conf. Circuits and System Theory*, 1970.
[13] A. A. Torn, "Cluster analysis using seed points and density-determined hyperspheres as an aid to global optimization," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-7, no. 8, pp. 610–616, Aug. 1977.
[14] W. L. Price, "Global optimisation by controlled random search," *J. Opt. Theory Appl.*, vol. 40, no. 3, pp. 333–348, July 1983.
[15] J. Gomulka, "A user's experience with Torns clustering algorithm," in *Towards Global Optimisation 2*, L. C. W. Dixon and G. P. Szego, Eds. New York: North-Holland, 1978.
[16] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.
[17] A. D. Griewank, "Generalized descent for global optimization," *J. Opt. Theory Appl.*, vol. 34, no. 1, pp. 11–38, May 1981.
[18] F. K. Branin, "A widely convergent method for finding multiple solutions of simultaneous nonlinear equations," *IBM J. Res. Develop.*, pp. 504–522, Sept. 1972.
[19] J. B. Mockus, "On Bayesian methods of optimization," in *Towards Global Optimisation*, L. C. W. Dixon and G. P. Szego, Eds. New York: North-Holland, 1975.
[20] E. Fagiuoli, P. Pianca, and M. Zecchin, "A mixed stochastic-deterministic technique for global optimisation," in *Towards Global Optimisation*, L. C. W. Dixon and G. P. Szego, Eds. New York: North-Holland, 1978.
[21] L. de Baise and F. Frontini, "A stochastic method for global optimisation: Its structure and numerical performance," in *Towards Global Optimisation 2*, L. C. W. Dixon and G. P. Szego, Eds. New York: North-Holland, 1978.
[22] H. J. Kushner, "A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise," in *Proc. Joint Automatic Control Conf.*, 1963.
[23] J. C. Hill and B. E. Stuckman, "Global suboptimization for functions of two variables," in *Proc. Conf. Artificial Intelligence*, Rochester, MI, 1983.
[24] B. E. Stuckman, "A new global search algorithm for functions of N bounded variables," Center of Robotics and Advanced Automation, Oakland University, Rochester, MI, TR-86-8-ESE-01, 1986.
[25] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1983.
[26] A. A. Goldstein and I. F. Price, "On descent from local minima," *Math. Comput.*, vol. 25, no. 115, July 1971.
[27] J. K. Hartman, Naval Postgraduate School, Monterey, CA, Rep. NP55HH72051A, 1972.
[28] B. E. Stuckman and J. C. Hill, "Pattern steering in HF phased array antennas," in *Proc. 29th Midwest Symp. Circuits and Systems*, Aug. 1986.
[29] S. E. Hersom, L. C. W. Dixon, and M. C. Biggs, "The optimisation of machine tool cost subject to constraints," Numerical Optimisation Centre, Hartfield, England, Rep. 5, 1969.
[30] G. C. Brown, "Optimisation methods in circuit design and modelling—comparison and critique," in *Optimisation Techniques in Circuit and Control Applications*, IEEE Conf. Pub. 66, 1970.
[31] J. J. McKeown and A. Nag, "An application of optimisation techniques to the design of an optical filter," presented at the IMA Conf., Bristol, England, 1974.

Bruce E. Stuckman (S'80–M'81–S'82–S'84–M'87) received the B.S. degree in engineering, the M.S. degree in electrical and computer engineering, and the Ph.D. degree in systems engineering from Oakland University, Rochester, MI, in 1981, 1983, and 1987, respectively. He was an Oakland University Teaching Fellow from 1983 to 1987.

Currently, he is a member of the faculty of the University of Louisville, where he is an Assistant Professor of Electrical Engineering. His research interest include design optimization, control theory, and data communications.

Dr. Stuckman is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.