ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220535936

A Direct Search Algorithm for Global Optimisation of Multivariate Functions.

Article *in* Australian Computer Journal · January 1991

Source: DBLP

CITATIONS
14

READS
234

2 authors, including:

Kurt Benke
State Government Victoria
111 PUBLICATIONS **1,851** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Global species richness View project

All content following this page was uploaded by Kurt Benke on 08 April 2014.

The user has requested enhancement of the downloaded file.

A direct search algorithm for global optimisation of multivariate functions

K.K. Benke and D.R. Skinner
Materials Research Laboratory (MRL)-DSTO,
PO Box 50, Melbourne, Victoria, 3032

We describe a direct search method for locating the global optimum of a multimodal function. This is an adaptive probabilistic algorithm suitable for any function of many variables subject to arbitrary constraints. The algorithm is based on a simple model for noise reduction and uses an iterative method for averaging random perturbations in the parameter estimates. No prior assumptions are required about the continuity of the search domain, or about the continuity, differentiability and modality of the function. The algorithm is very simple, requiring little preparation and is suitable for application to functions that are non-linear, noisy and of high dimensionality. The performance of the algorithm is demonstrated by a number of numerical examples, including a highly dimensioned problem in pattern recognition.

Keywords and Phrases: global optimisation, direct search procedures, adaptive random search, numerical methods, learning algorithms, pattern recognition, artificial intelligence, neural networks.

CR Categories: F.2.1, G.1.6, I.2.8, I.5.

Copyright © 1991, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the ACJ's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received December 1986, and in final form September 1989.

1. INTRODUCTION

An important goal in the design of adaptive systems, for use in pattern recognition and cybernetics, is the global search for a set of optimum operating conditions. The estimation of the optimal parameters of a mathematical model can be accomplished using special sequential methods, known as search procedures. In the general case, these search procedures require the existence of an explicit functional relationship predicting a response for each combination of control variables (Emshoff and Sisson, 1970).

The problem of finding an optimum on a multi-dimensional response surface has received considerable attention and various methods, both deterministic and probabilistic, are described in the literature (see, for example, Emshoff and Sisson, 1970; Gottfried and Weisman, 1973; Hillier and Lieberman, 1974; Taha, 1976; Glorioso and Osorio, 1980). Traditionally, numerical search procedures have often been based on gradient approaches, whose origins can be found in the theory and practice of differential calculus. At each step, the gradient is evaluated and a new search direction is chosen. Each new value differs from the previous value, depending on the specified step size, which determines search position and speed of convergence. Computation time increases rapidly with higher dimensionality, and there is a risk of entrapment at a local extremum since the final value achieved is effectively determined by the starting point. Stochastic methods, where the points at which a function evaluation occurs are chosen in a random manner, have been developed in an effort to improve the probability of locating the global extremum.

Many functions met in practice are non-differentiable and noisy, and may defy attempts to achieve an analytic solution (moreover, a noisy response surface may also introduce uncertainty in the calculation of the gradient). For these analytically intractable problems, statistical approaches are often the only recourse available. Swann (1974), and Jarvis (1975), in reviewing advantages associated with random search techniques, note that they are untroubled by plateaus, holes or discontinuities —which can cause serious problems with deterministic techniques.

Probabilistic search methods are particularly suitable for finding the global optimum on a multimodal surface, and are characterised by their extreme simplicity, especially when applied to non-linear functions of high dimensionality (Swann, 1974). Constrained simple random sampling can suffer, however, from the problem of slow convergence, requiring many trials for statistical accuracy. As shown by Gottfried and Weisman (1973), a deterministic gradient-based method is generally more efficient *if* the system is well behaved and can be characterised by a simple function that is smooth and continuous, unimodal and of low dimensionality.

A *purely* random search strategy is unmodified by any accumulated data, requiring little input and making very little use of any new information. The proximity of the final solution to the optimum is dependent on the number of trials, but independent of surface topography. For example, the probability of locating the maximum of a one-dimensional function to within a fraction β of the permissible range of the independent variable, after k trials, is given by $P(\beta) = 1 - (1 - \beta)^k$.

A random process selects the control parameters in a non-sequential, or parallel, manner and if the statistical properties of the search procedure itself can be controlled, then greatly improved convergence is possible. An *adaptive* probabilistic search is an attempt to provide faster convergence and greater statistical accuracy, whilst retaining the fundamental advantages of the simple random search.

In this context, a learning or adaptive system may be defined (Narendra and Thathachar, 1974) as "one characterised by its ability to improve its behaviour with time, in some sense tending towards an ultimate goal". In this paper, such a process is taken to include one for which:

(i) an operation is applied iteratively to the data,
(ii) the operation is a function of a number of parameters,
(iii) at each iteration, there is an opportunity to adjust some or all of these parameters, and
(iv) parameter adjustment is directed towards improving a performance index.

This description of learning is consistent with the usage of Narendra and Thathachar (1974), Shimura (1978) and Kohonen (1984), as applied to stochastic automata and artificial neural networks.

In the past, probabilistic learning algorithms, known as stochastic automata, have been applied to optimisation problems (McMurtry and Fu, 1966; Shapiro and Narendra, 1969; Jarvis, 1969; Wiswanathan and Narendra, 1973; Devroye, 1976). The stochastic automaton uses a reinforcement scheme designed to modify the search pattern. This involves a penalty or reward, based on the relative success of previous trials. Successes lead to transitions to higher states analogous to those occurring in a discrete Markov process.

Surveys of these learning automata reveal that they have potential in many areas of adaptive control, pattern recognition and systems analysis (Narendra and Thathachar, 1974; Narendra and Lakshmivarahan, 1977; Tsypkin and Poznyak, 1977; Oomen and Hansen, 1984). However, they can be mathematically complex in terms of their structure and computational implementation. In addition, the application of learning automata to parameter optimisation is still in a developmental stage. For systems with a large number of dimensions, convergence can be very slow. This method of optimisation cannot be applied directly if the parameter space is defined for a continuous range of values.

Another recent approach to multivariate optimisation, referred to as simulated annealing, uses time-dependent probabilistic rules as an aid to avoiding entrapment at a local optimum (Kirkpatrick, Gelatt and Vecchi, 1983; Hinton and Sejnowski, 1983). These rules are formally equivalent to the conditions for thermally induced escape from a potential well, and, as the optimisation proceeds, are gradually made more stringent in a manner analogous to the annealing of solids. This approach has been developed into a learning algorithm for an artificial neural network known as the Boltzmann machine (Ackley, Hinton and Sejnowski, 1985). Unfortunately, convergence is inherently slow, and attempts to speed up the process by a too-rapid tightening of escape criteria can lead to local entrapment.

In the next section, we describe an optimisation algorithm that is simple, requires little preparation and is applicable to a wide range of problems. It has the fundamental advantages of the probabilistic search process, viz. improved performance in non-linear or multimodal environments of high dimensionality, and alleviates its major weakness, i.e. slow convergence and poor statistical accuracy. We show that, as a proposed global algorithm, it compares very favourably with a number of common global algorithms used in the past as benchmarks. The numerical examples are chosen mainly for ease of comparison between global algorithms, and in some cases could have been solved by other methods.

2. THE OPTIMISATION ALGORITHM

We consider specifically the maximisation of a function of many variables that is everywhere positive and finite, but application to most practical forms of optimisation represents a trivial extension. The function to be maximised is of arbitrary complexity and is treated as a non-linear mathematical programming problem with prescribed constraints. The approach used for optimisation has been applied to a number of diverse problems and consistently finds the global optimum, while converging much faster and more accurately than many competing global search algorithms.

Suppose we have a positive figure of merit, f, which is a function of a large number of variables expressed as a vector $\mathbf{X} = (x_1, x_2, ..., x_d)$ defined in a d-dimensional Euclidean space. The problem is to find the set of components x_j which will maximise f subject to a given set of constraints. The function may be non-differentiable and non-linear, so that the problem is not amenable to established techniques based on differential calculus or linear programming. The components may appear in the constraints as non-linear combinations, and the parameters may or may not be restricted to integral or positive values.

The problem may be expressed more formally as follows. Let f be an objective function defining a hypersurface in a d-dimensional real Euclidean space, R^d. Then the

constrained optimisation problem is:

Maximise $f = f(\mathbf{X})$; $f > 0$
subject to $\mathbf{X} \in F^d$; $F^d \subseteq R^d$

where F^d is the subspace defining the feasible region. We employ a probabilistic search for f_{max}, that is, we attempt to generate a sequence of vectors $\{\mathbf{X}_k\}$, sampled from within F^d, which converge to the optimal vector $\mathbf{X}_{opt}$ by a stochastic process, such that

$$\lim_{k\to\infty} \mathbf{X}_k \rightarrow \mathbf{X}_{opt}$$

or, more specifically,

$$\text{Prob}\{||\mathbf{X}_k - \mathbf{X}_{opt}|| > \delta\} \rightarrow 0 \text{ as } k \rightarrow \infty \quad ; \delta > 0$$

so that $\mathbf{X}_k$ tends to $\mathbf{X}_{opt}$ *in probability.*

Suppose now we have two independent estimates for $\mathbf{X}_{opt}$, denoted by $\mathbf{X}_1$ and $\mathbf{X}_2$. These may be expressed as

$$\mathbf{X}_1 = \mathbf{X}_{opt} + \Delta_1 \tag{1}$$

$$\mathbf{X}_2 = \mathbf{X}_{opt} + \Delta_2 \tag{2}$$

where Δ_1 and Δ_2 represent error vectors. By analogy with electronic noise reduction through signal averaging, we may attempt to reduce the error vector by taking a weighted-mean vector of the form

$$\begin{aligned}\mathbf{X}_3 &= \alpha\mathbf{X}_1 + (1 - \alpha)\mathbf{X}_2 \\ &= \mathbf{X}_{opt} + \alpha\Delta_1 + (1 - \alpha)\Delta_2\end{aligned} \tag{3}$$

where $0 \leq \alpha \leq 1$. If we define

$$\Delta_3 = \alpha\Delta_1 + (1 - \alpha)\Delta_2 \tag{4}$$

then

$$\mathbf{X}_3 = \mathbf{X}_{opt} + \Delta_3 \tag{5}$$

(If the feasible region is disconnected or not convex, the value of $\mathbf{X}_3$ obtained by weighted averaging may not obey the problem constraints. We have not specifically considered such problems, but for connected regions we suggest dividing in the ratio $\alpha : (1 - \alpha)$ the shortest line that connects $\mathbf{X}_1$ and $\mathbf{X}_2$ and that lies wholly within the feasible region, provided such a line can be found in practice.)

We wish to know the value of α that would be expected to minimise the magnitude of the vector Δ_3, which is given by

$$\begin{aligned}||\Delta_3|| &= \{(\alpha\Delta_1 + (1 - \alpha)\Delta_2) \cdot (\alpha\Delta_1 + (1 - \alpha)\Delta_2)\}^{1/2} \\ &= \{\alpha^2||\Delta_1||^2 + (1 - \alpha)^2 ||\Delta_2||^2 + \\ &\quad 2\alpha(1 - \alpha)\Delta_1 \cdot \Delta_2)\}^{1/2}\end{aligned} \tag{6}$$

In practice, we will attempt to minimise the square of the magnitude, and for this purpose we will define

$$Z \equiv E[||\Delta_3||^2 | \Delta_1, \Delta_2]$$

that is, Z is the conditional expectation value for $||\Delta_3||^2$ given Δ_1 and Δ_2.

In the absence of any information about the components $\delta_{1,j}$ and $\delta_{2,j}$ of the error vectors of Δ_1 and Δ_2, we assume that each component comes from a finite set of identical uncorrelated ensembles of random values with zero mean. This implies that

$$E[\sum_{j=1}^{d} \delta_{1,j}.\delta_{2,j}] = 0 \tag{7}$$

Hence

$$E[\Delta_1 \cdot \Delta_2] = 0 \tag{8}$$

and therefore

$$Z = \alpha^2||\Delta_1||^2 + (1 - \alpha)^2 ||\Delta_2||^2 \tag{9}$$

By differentiating this last equation with respect to α, it is easily shown that Z will be minimum for the condition

$$\alpha = ||\Delta_2||^2 / (||\Delta_1||^2 + ||\Delta_2||^2) \tag{10}$$

that is, the vectors should be weighted inversely as the square of their error vectors. The expectation value for the square of the magnitude of the weighted-mean error vector will then be

$$Z = ||\Delta_1||^2||\Delta_2||^2 / (||\Delta_1||^2 + ||\Delta_2||^2) \tag{11}$$

If the distributions of Δ_1 and Δ_2 are concentrated near their mean values, we can replace $||\Delta_1||^2$ and $||\Delta_2||^2$ by their expectation values. It follows that Z must be less than both $E[||\Delta_1||^2]$ and $E[||\Delta_2||^2]$. Therefore, the expected error in the weighted mean vector $\mathbf{X}_3$ should be smaller than that in *either* $\mathbf{X}_1$ or $\mathbf{X}_2$.

Consider now the case where the two vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ are independent random selections of vectors within a search domain, and the expectation value for the square of the magnitude of error vector for such a random selection is $E[||\Delta||^2] \equiv Z_1$. If we could combine $\mathbf{X}_1$ and $\mathbf{X}_2$ inversely as the squares of their error vectors, we would expect the magnitude of the new error vector to be about $Z_2 = Z_1 / 2$. In general, if Z_k is the expectation for $||\Delta||^2$ after the optimal combination of k vectors, optimal combination of the parameter vector $\mathbf{X}$ with a new random selection will give

$$Z_{k+1} = Z_k Z_1 / (Z_k + Z_1) \quad ; k > 1 \tag{12}$$

and hence

$$Z_k = Z_1 / k \tag{13}$$

The important point to note about this variance reduction is that the ratio of Z_k *to* Z_1, is independent of the number of dimensions, d, in which the vectors are defined. (However, we might surmise that Z_1 would be roughly proportional to d for similar problems and that the requirements for the precision of $\mathbf{X}_{opt}$ would become more stringent as d increased.) We would therefore expect (and indeed demonstrate later) that, in the case of multi-dimensional search, the estimate for $\mathbf{X}_{opt}$ would converge to the true value much faster than would the estimate from constrained random search for which it may be shown that

$$Z_k \cong Z_1 / k^{2/d} \tag{14}$$

where k here refers to the number of function evaluations.

(In practice, using the strategy leading to Equation (13), we require two function evaluations per iteration, one for the new random parameter set and one for the weighted mean. Under some conditions, the algorithm will converge faster if more than two function evaluations are used.)

Unfortunately, it is not practicable to weight vectors exactly as described above, since we would need to know the value of $\mathbf{X}_{opt}$ in order to compute α. However, this is not an insuperable difficulty. We would expect that the error vector magnitude $||\Delta||$ would have a negative correlation with the figure of merit f. If, therefore, we weight the vectors $\mathbf{X}_j$ directly as f, we will have at least an approximation to the optimal weighting.

It is informative to see how this deviation from the ideal ratio, $\alpha : (1 - \alpha)$, would affect our estimates for Z_k. If we consider a simple unimodal function, it is clear that the practical weighting factor $f(\mathbf{X}_1) / (f(\mathbf{X}_1) + f(\mathbf{X}_2))$ can be a reasonable approximation to the ideal $||\Delta_2||^2 / (||\Delta_1||^2 + ||\Delta_2||^2)$ when the vectors span a range from the neighbourhood of the optimum to a region where f is much smaller than its maximum. In contrast, if both vectors are close to the optimum, the values of f will be similar, so that the weights assigned to the vectors will all be about the same. This means that, if the region of the domain occupied by the global peak is reduced, the probability of finding a vector close to the optimum is also reduced, but once such a vector is found the approximation to ideal weighting is improved.

We demonstrate this relationship for a simple function of many variables, i.e.

$$f(\mathbf{X}) = \exp(-b||\mathbf{X}||^2 / d) \qquad (15)$$

where

$$\forall j, \; -1 \leq x_j \leq 1; \; 1 \leq j \leq d$$

and d again refers to the number of dimensions in $\mathbf{X}$. For this function, it is trivial to show that Z_1 is equal to $d/3$, independent of b, so the value of Z_k as defined above would also be independent of b. Figure 1 shows the expectation Z_k as a function for k for the case of $d = 1000$ (a very highly dimensional function), together with a typical actual performance of the algorithm for the cases $b = 2$, 10 and 25. For $b = 2$, the central peak (to the point of inflexion) occupies most of the domain, whilst for $b = 10$ and 25 it occupies progressively less. We would therefore expect that the estimate for $||\Delta||^2$ after a large number of iterations would be poor for the first case, and much better for the others. This is borne out by the graph, which shows the actual $||\Delta||^2$ diverging from Z_k after about five iterations for $b = 2$, but after about 100 iterations for $b = 10$. For $b = 25$, the value of $||\Delta||^2$ diverges early from Z_k due to the lowered probability of finding a vector near the optimum, but returns to near the expectation value when the weighted averaging becomes effective.

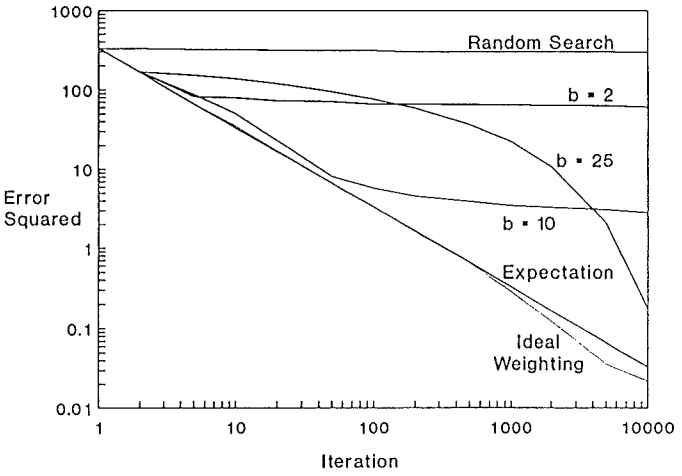

Figure 1. Values achieved for the square of the error-vector magnitude with random search and with practical weighted averaging for the function $f(\mathbf{X}) = \exp(-b||\mathbf{X}||^2/d$, $d = 1000$, $b = 2$, 10 and 25. Expectation and actual values in the case of ideal weighted averaging are also shown.

Figure 1 also shows $||\Delta||^2$ for the impractical case where α is actually computed from the error vector, thus using knowledge of the sought-for answer. Now, Z_k represents a good approximation to $||\Delta||^2$, which dips somewhat below expectation for large k, probably because weighted means are rejected when they represent no improvement. Finally, the figure shows the results of a simple constrained random search, in which, as would be expected for a problem of such high dimensionality, there is no significant reduction in $||\Delta||^2$ even after 10 000 iterations. These results confirm that the estimate of Z_k/Z_1, which is independent of the number of dimensions d, can be appropriate over a significant number of iterations, depending on the function f. The results also confirm the counter-intuitive notion that the search can converge more quickly to a narrow peak than to a broad one. What they do not show is how well the search converges to an optimum on a multimodal surface, and, because this will be very dependent on the nature of the function, it appears that this can be established only by experiment.

It may be that there are certain symmetry operators $S(\mathbf{X})$ that can be applied to the vector without altering the figure of merit, i.e.

$$f(S(\mathbf{X})) \equiv f(\mathbf{X})$$

A typical case would be sign inversion, where $S(\mathbf{X}) = -\mathbf{X}$. Where n_s such operators exist, two estimates for the optimum vector may, in fact, approximate to different but equivalent vectors $\mathbf{X}_{opt}$. In this situation, the feasible region F^d actually comprises $n_s + 1$ equivalent subspaces. If we form the weighted mean not only with each new random vector, but also with the new vector after application of each symmetry operator, we effectively need to investigate only one of the subspaces. The volume of the feasible region has thus been reduced by a factor of around $n_s + 1$, and we would therefore expect to reduce the number of

iterations to reach a given precision by a similar factor. However, the number of function evaluations for each iteration of the algorithm will now be $n_s + 2$ instead of 2, so the required number of function evaluations will be multiplied by a factor of about $(n_s + 2) / (2n_s + 2)$, e.g. 3/4 for $n_s = 1$.

In order to make a systematic and global search for an optimal solution, whilst taking maximum advantage of the variance-reduction strategy described above, we propose the following algorithm for iterative optimisation, for the case where there is a single symmetry operator.

Given that there exists an operator $S(\mathbf{X})$ such that

$$f(S(\mathbf{X})) \equiv f(\mathbf{X})$$

and defining

$$\mathbf{X}_b(k) \equiv \text{best estimate for } \mathbf{X}_{opt} \text{ after } k \text{ iterations}$$

$$\mathbf{X}_r(k) \equiv \text{random vector, } \mathbf{X}_r(k) \in F^d$$

$$\mathbf{X}_{w1}(k) \equiv \frac{f(\mathbf{X}_b(k))\mathbf{X}_b(k) + f(\mathbf{X}_r(k))\mathbf{X}_r(k)}{f(\mathbf{X}_b(k)) + f(\mathbf{X}_r(k))}$$

$$\mathbf{X}_{w2}(k) \equiv \frac{f(\mathbf{X}_b(k))\mathbf{X}_b(k) + f(\mathbf{X}_r(k))S(\mathbf{X}_r(k))}{f(\mathbf{X}_b(k)) + f(\mathbf{X}_r(k))}$$

and

$$T(k) \equiv \{ \mathbf{X}_b(k), \mathbf{X}_r(k), \mathbf{X}_{w1}(k), \mathbf{X}_{w2}(k) \}$$

then $\forall \mathbf{X}_b(k), \mathbf{X}_r(k), \mathbf{X}_{w1}(k), \mathbf{X}_{w2}(k) \in F^d$

$\mathbf{X}_b(1) = \mathbf{X}_r(1)$ and for $k \geq 1$

$$\mathbf{X}_b(k+1) = \begin{cases} \mathbf{X}_b(k) \text{ if } f(\mathbf{X}_b(k)) = \max_{\mathbf{X} \in T(k)} f(\mathbf{X}) \text{ else} \\ \mathbf{X}_r(k) \text{ if } f(\mathbf{X}_r(k)) = \max_{\mathbf{X} \in T(k)} f(\mathbf{X}) \text{ else} \\ \mathbf{X}_{w1}(k) \text{ if } f(\mathbf{X}_{w1}(k)) = \max_{\mathbf{X} \in T(k)} f(\mathbf{X}) \text{ else} \\ \mathbf{X}_{w2}(k) \end{cases}$$

Where there is no symmetry operator $S(\mathbf{X})$, the deletions relevant to $\mathbf{X}_{w2}(k)$ are obvious, as are the extensions for more than one $S(\mathbf{X})$.

The third possible choice for $\mathbf{X}_b(k+1)$ is, of course, a weighted sum, or centroid, between the current best estimate and a noise injection, whilst, where $S(\mathbf{X}) = -\mathbf{X}$, the fourth possibility represents a weighted difference. It has been found, in practice, that allowance for a weighted difference can speed convergence even when $f(-\mathbf{X}) \neq f(\mathbf{X})$. The conditions under which this represents a computational saving are currently under investigation.

Pseudo-code for this algorithm, which we call the "centroid algorithm", is given in Table 1. After several iterations have been completed, $\mathbf{X}_1$ represents the evolving trend, $\mathbf{X}_2$ a random input and $\mathbf{X}_3$ and $\mathbf{X}_4$ perturbations to the trend. In each iteration, step 3.7 makes a decision on substitution for $\mathbf{X}_1$ after comparing several inputs. Usually, there are only a few early substitutions due to random inputs. After a limited number of iterations, usually about two or three, the substitutions are predominantly learned rather than random, that is, are from $\mathbf{X}_3$ or $\mathbf{X}_4$.

Table 1. Psuedo-Code for Centroid Algorithm.

```
1. Generate random parameter vector X1 within specified constraints.
2. f1 = Function(X1)
3. FOR required number of trials DO 3.1 to 3.7
   BEGIN find improved parameter vector
      3.1 Generate random parameter vector X2 within constraints
      3.2 f2 = Function(X2)
      3.3 Compute the weighted mean X3 of X1 and X2 according to
            X3 = (f1X1 + f2X2) / (f1 + f2)
      3.4 Compute a further weighted mean X4 according to
            X4 = (f1X1 + f2S(X2)) / (f1 + f2)
      3.5 f3 = Function(X3)
      3.6 f4 = Function(X4)
      3.7 IF f1 = Max(f1, f2, f3, f4) THEN
            no action
          ELSE IF f2 = Max(f1, f2, f3, f4) THEN
          BEGIN replacement due to random vector
            X1 = X2
            f1 = f2
          END
          ELSE IF f3 = Max(f1, f2, f3, f4) THEN
          BEGIN replacement due to first weighted mean
            X1 = X3
            f1 = f3
          END
          ELSE f4 must be the maximum, and
          BEGIN replacement due to second weighted mean
            X1 = X4
            f1 = f4
          END
   END
4. Optimum parameter vector is current value of X1
   Optimum function value is current value of f1
5. END of algorithm
```

3. NUMERICAL EXAMPLES

In this section, we apply the algorithm to several problems with known solutions. In the first case, a multimodal problem is solved using the centroid algorithm and also a purely random search for comparison. In the second and third problems, the procedure is applied to multimodal functions and compared with several algorithms used for global optimisation. The final problem considered is relevant to pattern recognition and digital spatial filtering, and illustrates the performance under conditions of high dimensionality, non-linearity and multiple constraints.

Problem 1. A Multimodal Function with Closely Competing Optima

For the first problem, we devised a function specifically to test the global performance of the centroid algorithm and

compare it with that of a simple random search (sometimes referred to as a Monte Carlo optimisation). This function is the sum of five two-dimensional Gaussian distributions and represents a multimodal surface with a number of peaks closely contesting for optimum status. The task is to find the global maximum of the following function, independent of starting point, using an iterative algorithm:

$$f(x, y) = \sum_{i=1}^{5} a_i \exp(-((x - x_i)^2 + (y - y_i)^2) / s_i^2) \quad (16)$$
$$-2 \leq x \leq 2, -2 \leq y \leq 2$$

The values for a_i, x_i, y_i and s_i are provided in Table 2. A contour map of the function is plotted in Figure 2 (a) and an isometric projection depicted in Figure 2 (b). The global maximum of the function is:

$$f(-0.01356, -0.01356) = 1.29695$$

The global maximum is located on a peak which rises above all local maxima over an area of about 0.05% of the search domain. This peak appears in Figure 2 (b) as a small spur on the side of a ridge, which itself consists of a barely perceptible saddle point between two local maxima,

$$f(-0.289, -0.206) = 1.217$$
$$f(-0.206, -0.289) = 1.217$$

There are also two more distant well-resolved maxima,

$$f(-0.003, 0.994) = 1.207$$
$$f(0.994, -0.003) = 1.207$$

The difference in height (above the base plane) between the global maximum and *any* of the four local maxima is no more than 7%. The five peaks are therefore competing very closely for optimum status. The function also contains one minimum and several saddle points. The combined effect of all these stationary points is to surround the global peak and thus thwart any gradient-based deterministic method. On the outer perimeter of the active area of the function, far from the stationary points, there are flat regions providing very little information for gradient-based methods (see Figure 2 (b)).

We will compare the performance of the centroid algorithm with that of a pure random search, ignoring the symmetry operator $S((x,y)) = (y,x)$. For assessment purposes, we will use two indices of performance. The first index is a measure of global performance and is defined as the number of function evaluations required to find a position on the global peak, with a function value higher than all local maxima (similarly, in the case of minimisa-

Table 2. Parameters for Multi-Gaussian Test Function.

i	a_i	x_i	y_i	s_i
1	0.5	0.0	0.0	0.1
2	1.2	1.0	0.0	0.5
3	1.0	0.0	—0.5	0.5
4	1.0	—0.5	0.0	0.5
5	1.2	0.0	1.0	0.5

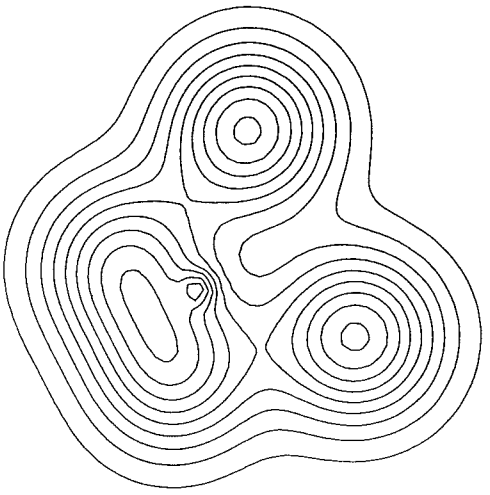

Figure 2 (a). Contour plot of the multi-Gaussian function specified by Table 2 and used in the first numerical example.

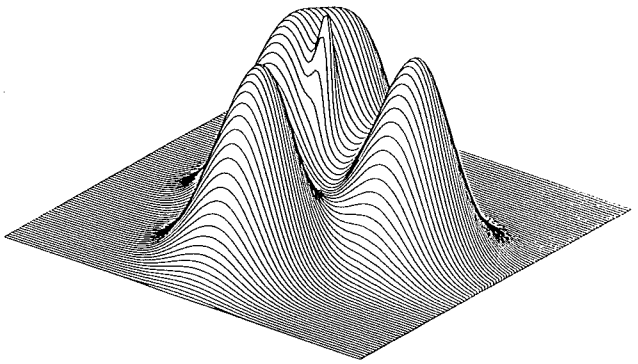

Figure 2 (b). Isometric plot of the function of Figure 2 (a), as seen looking towards the origin from the (+x, +y) quadrant.

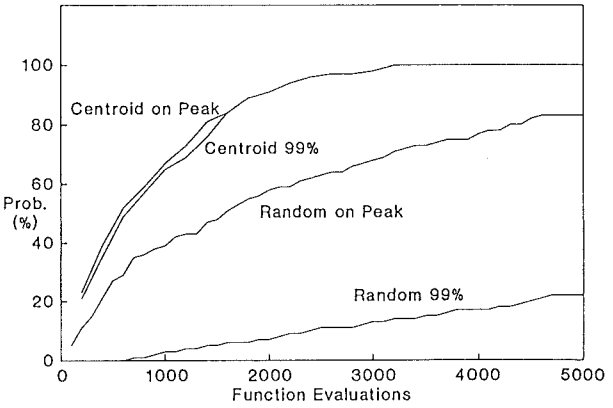

Figure 2 (c). Probabilities of finding the global peak, and achieving a function value of 99% of the maximum, for both the centroid algorithm and constrained random search when maximising the function of Figures 2 (a) and 2 (b). Results, in terms of function evaluations, are averaged over 100 experiments.

tion we require a position within the global valley having a value lower than all local minima). The second index is a measure of speed of convergence and is defined as the number of function evaluations required to solve the problem to a specified precision (for example, 99% of the maximum function value). The latter index is commonly used in unimodal search (see Box, 1965).

To achieve statistical confidence, we averaged the results of 100 independent experiments for each algorithm. Each experiment was started at a random point within the feasible region and halted after 5000 function evaluations. The results are plotted in Figure 2 (c). After 5000 function evaluations, the random search found a position on the global peak in 85% of experiments. The centroid algorithm found the global peak with a probability of 85% after only 1200 function evaluations, representing an improvement of 4:1 in global performance. This is an intriguing result, since a purely random search is generally regarded as the best method for finding the global peak (Gottfried and Weisman, 1973; Bekey and Ung, 1974).

As expected, the random search method was unable to match the centroid algorithm with respect to speed of convergence to the maximum function value. The random search converged to a function value within 99% of the global maximum in 20% of experiments after 5000 function evaluations. The centroid algorithm achieved the same result in fewer than 200 function evaluations, representing an improvement of 25:1 in speed of convergence. It is interesting to note that the plotted results for the global performance and the speed of convergence of the centroid algorithm are almost identical. In other words, the algorithm performs simultaneously as a global search technique and as a hill-climbing technique.

The advantage of the centroid algorithm over constrained random search is not as clear-cut when the global maximum is totally isolated and far from the domain of other significant function values. If a sixth peak is added to the function of Figure 2, using the specifications

$$a_6 = 1.35, x_6 = -1.5, y_6 = -1.5, s_6 = 0.1,$$

the new global maximum becomes a thin, isolated peak rising from the floor to a value only 4% higher than the previous maximum, as shown in Figure 3 (a). The results of an investigation into convergence in this case, again ignoring the symmetry operator, are shown in Figure 3 (b). In terms of function evaluations, even for this difficult case, the centroid algorithm still converged to a function value of 99% of the maximum much faster, by a factor of around 3, than did random search. However, it was now not much better than half as fast as the constrained random search in finding a position on the global peak itself, since the adaptive component of the algorithm did not contribute significantly to the early stages of the search (the authors are indebted to one of the reviewers, who suggested the possibility of this behaviour).

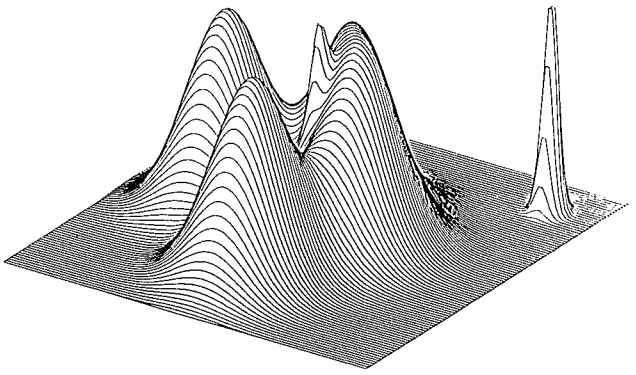

Figure 3 (a). Isometric plot of the six-peak Gaussian function, as seen looking from the (—x, +y) quadrant.

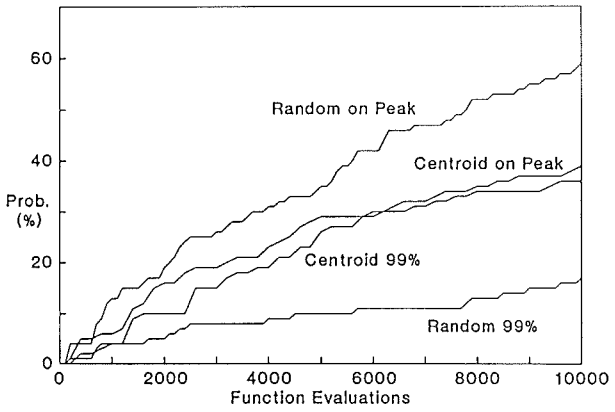

Figure 3 (b). Probabilities of finding the global peak, and achieving a function value of 99% of the maximum, for both the centroid algorithm and constrained random search when maximising the function of Figure 3 (a). Results, in terms of function evaluations, are averaged over 100 experiments.

Problem 2. Comparison with Other Adaptive Random Search Techniques

The second problem studied is to find the location of the global minimum of the following non-linear function, which was studied originally by Uosaki, Imamura, Tasaka and Sugiyama (1970), and later in greater detail by Bekey and Ung (1974):

$$f(x, y) = (1 - 8x + 7x^2 - \tfrac{7}{3}x^3 + \tfrac{1}{4}x^4)\, y^2 \exp(-y) \quad (17)$$
$$0 \leq x \leq 5, 0 \leq y \leq 6$$

This surface has two minima at

$f(1,2) = -1.12779$ (local minimum)
$f(4,2) = -2.34581$ (global minimum)

Uosaki *et al* investigated the properties of an algorithm based on a modified Kiefer-Wolfowitz procedure (itself an extension of the stochastic approximation procedure of Robbins and Munro (Shimura, 1978)). Their algorithm attempts to find the global optimum by the simple expedient of adding to each observation noise with zero mean

and non-zero variance which decreases with each observation. Their experiments showed that from a starting point of (1.0, 4.5), which is near the local minimum, the global minimum could be located. Bekey and Ung found, however, that this could not be achieved reliably and that the global performance depends on the random number sequence generated during the course of the optimisation.

Nevertheless, the algorithm of Uosaki *et al* produced superior global performance relative to the "complex" method of Box (1965), which is a probabilistic extension of the simplex methods described by Spendley, Hext and Himsworth (1962), and Nelder and Mead (1965). The test results of Uosaki *et al* indicate that the method of Box can find the global extremum in only about 50% of experiments. Recent research by Rangaiah and Krishnaswamy (1987) shows also that the method of Box is lacking somewhat in global peformance; in their tests it converged to the global optimum in no more than 50% of experiments.

Bekey and Ung solved the above problem much more efficiently using an algorithm based on a modified random-creep procedure. This algorithm first locates a local minimum and then searches the parameter space with vector steps whose mean length gradually increases. Their results are reproduced in Table 3. Starting at the point (1.0, 4.5), the algorithm required 349 function evaluations to reach the global valley and a minimum function value was obtained after 451 function evaluations. The algorithm produced greatly improved global performance relative to the previous methods, finding the global minimum within 500 function evaluations in 26 out of 27 experiments.

When the centroid algorithm was applied to this problem (noting that minimising $f(x, y)$ is equivalent to maximising $-f(x, y)$, and ignoring symmetry conditions for simplicity), it required only 20 function evaluations to reach the global valley and the location of the minimum function value was obtained with greater precision after only 106 function valuations (see Table 3). At this point, the error from the true minimum was an order of magnitude lower than that of the Bekey-Ung algorithm. The global valley was found within 20 function evaluations in 27 out of 27 experiments. The median number of function evaluations required to find the global valley was only 6 (the variation being due to the particular sequence of random numbers generated after starting from the initial point (1.0, 4.5)). The centroid algorithm is therefore faster than the Bekey-Ung algorithm by a factor of at least 16:1 in the number of function evaluations required to reach the global valley, and by a factor of 4:1 in the number of function evaluations required to converge to the minimum function value.

It is interesting to note that, since the starting point was near the local minimum, both algorithms converged to this stationary point first before proceeding to the global minimum. The centroid algorithm converged almost to the local minimum value within four function evaluations before switching to the global minimum, whereas the algorithm of Bekey and Ung required 85 function evaluations to match this precision and was still lingering on the local minimum after 108 function evaluations.

Many algorithms claimed to have a global capability are in fact simply probabilistic versions of existing deter-

Table 3. Comparative Performance of Direct Search Algorithms.

(a) Bekey-Ung Algorithm.

Function Evaluations	x	y	f
0	1.00000	4.50000	—0.46866
3	1.07864	4.14765	—0.56397
5	0.88992	3.85862	—0.64812
6	0.73725	3.63153	—0.68240
7	0.78464	3.58483	—0.71298
17	0.79484	3.27404	—0.81487
19	0.89123	3.22341	—0.85388
23	1.12829	3.05294	—0.90724
25	0.93272	2.91856	—0.95516
27	0.83677	2.75890	—0.98259
28	0.76319	2.58384	—0.99818
30	0.70626	2.42434	—0.99820
35	0.68263	2.06214	—1.02059
42	0.75268	2.10778	—1.06371
44	0.80029	2.03451	—1.08912
46	0.83687	1.84182	—1.09569
47	1.10311	1.84612	—1.11297
56	0.97345	1.83422	—1.11904
71	0.90508	1.96726	—1.11954
85	0.92422	1.90217	—1.12004
87	0.97299	2.04229	—1.12669
103	1.02506	1.98295	—1.12721
108	1.00528	1.98600	—1.12771
349	4.08324	3.53003	—1.57444
350	3.99468	3.28736	—1.74914
352	3.99070	3.07293	—1.89385
354	3.93506	2.99573	—1.93899
355	4.16349	2.93650	—1.94227
358	4.04506	2.84550	—2.03576
361	4.06679	2.54815	—2.19399
364	4.13033	2.15666	—2.30275
367	3.88618	2.03345	—2.32544
370	3.90530	2.06060	—2.32991
373	4.05143	1.93819	—2.33911
385	4.00596	1.95753	—2.34470
396	4.00477	1.97394	—2.34540
451	3.99093	2.01405	—2.34555

(b) Centroid Algorithm

0	1.00000	4.50000	—0.46866
2	1.16650	2.37807	—1.07383
4	0.95133	2.01478	—1.12573
20	3.30404	2.38261	—1.77300
24	3.86722	2.03921	—2.31837
32	3.89228	2.12935	—2.31876
38	3.94973	2.04625	—2.34059
68	3.95468	2.02009	—2.34233
86	3.95867	1.98391	—2.34295
106	4.00289	2.00373	—2.34579

ministic algorithms. For example, Beltrami and Indusi (1972) developed an adaptive random search algorithm by refining the algorithm of Lawrence and Steiglitz (1972), who in turn had randomised the pattern search method of Hooke and Jeeves (1959). The algorithm has a limited global search capability, for example, detecting the global minimum of the following function on 50% of occasions:

$$f(x, y) = 1.41x^4 - 12.76x^3 + 39.91x^2 - 51.93x + 24.37 + (y - 3.9)^2 \quad (18)$$

The global minimum occurs at $f = -3.98$ at (3.48, 3.9) and the centroid algorithm will find this consistently in under 100 function evaluations, despite starting at random points located within the active area of the local minimum itself at (1.4, 3.9).

The Rastrigin function (see Polovinkin, 1981) is a more complicated example of a multimodal function. We have applied the centroid algorithm to the location of the minimum of this function, which is given by

$$f(x, y) = x^2 + y^2 - \cos(18x) - \cos(18y) \quad -1 \leq x \leq 1, -1 \leq y \leq 1 \quad (19)$$

where the accuracy required for locating the global minimum point, $f(0,0) = -2$, is $x^2 + y^2 \leq 4/(5917^\pi)$. This function (used as a test in the past by Soviet mathematicians) is smooth and continuous, with 50 local minima in a lattice arrangement, and is quite amenable to a random method incorporating some form of gradient-descent procedure for local search. Even with this type of function, the centroid algorithm with no prior knowledge or arbitrary parameters, and using one symmetry condition, i.e. $\mathbf{X}_{w1}$ and $\mathbf{X}_{w2}$ representing weighted mean and difference of $\mathbf{X}_b$ and $\mathbf{X}_r$, typically required only about 380 function evaluations.

In contrast, Polovinkin reported that Monte Carlo search required 5917 evaluations, and multi-start random and multi-start gradient methods 1176 and 556, respectively. He did find, however, that an involved algorithm of "competing points" could accomplish this in as few as 111 function evaluations, but only on the basis of considerable *a priori* information (e.g. volume of global minimum is larger than that of all local minima), assumptions that were clearly invalid for general application. Experimental details were lacking, but preparation and running times appear to be significant. In the general literature, the conditions of experimental comparisons are not always completely described, making evaluation of relative efficiencies of algorithms (which may vary from problem to problem) a somewhat hazardous occupation.

Problem 3. Comparison with Stochastic Automata

The third demonstration of this algorithm is an application to a very noisy bimodal test function (Figure 4 (a)). The function $E(x)$, defined only for integral abscissae, was used by Shapiro and Narendra (1969), and others (Wiswanathan and Narendra, 1973; Devroye, 1976) to test the performance of various stochastic automata for the purpose of parameter optimisation. The noisy measurements, $f(x, r)$, are related to the expected values, $E(x)$, by the following function:

$$f(x, r) = E(x) + r \quad (20)$$

The random noise variable, r, is uniformly distributed over the range $(-2.0 \leq r \leq 2.0)$. A particular value of the measurement $f(x, r)$, at a given iteration during optimisation, has a uniform probability of occurring within a band about the function $E(x)$ to be maximised. The function was designed to preclude the use of many conventional optimi-

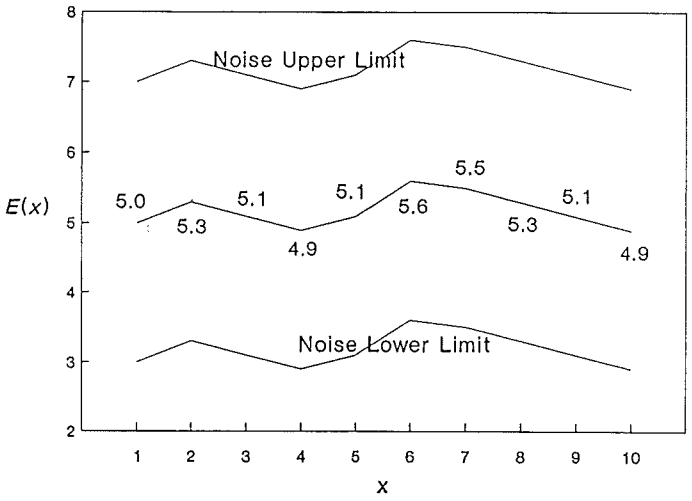

Figure 4 (a). Benchmark function designed by Shapiro and Narendra (1969) for testing stochastic automata. The function is defined only for integral abscissae, and is subject to random variation over the range defined by the outer lines.

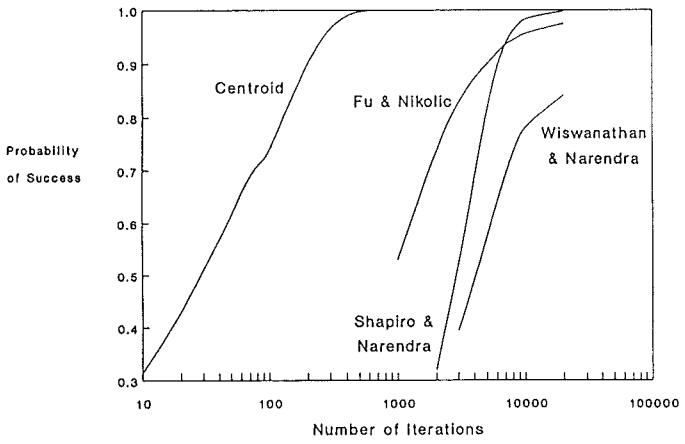

Figure 4 (b). Comparison of the performance of the centroid algorithm with typical results of the stochastic automata of Shapiro and Narendra (1969), Wiswanathan and Narendra (1973) and Fu and Nikolic (1966) as collated by Devroye (1976). The probability of finding the maximum of the function of Figure 4 (a) is plotted against the number of algorithm iterations.

sation methods, such as gradient-based approaches. The function is multimodal and the variations in $E(x)$ are much smaller than the variance of the added noise (that is, the signal is effectively submerged in the noise).

We carried out 1000 experiments, using different sets of random numbers, and at each iteration in each experiment recorded the value of x which, with the added random noise, gave the highest function value. This information was used to compute the probability of obtaining the optimum value of $x(6)$ as a function of the number of iterations (see Figure 4 (b)). This probability is 0.95 after 260 trials, calculated using the centroid algorithm with no symmetry operator. For comparison, we also plotted the performance of several stochastic automata. The best is the algorithm of Shapiro and Narendra (1969), which reaches 0.95 in 7400 trials. If the computing loads for iterations of the two algorithms are comparable, then the centroid algorithm is an order of magnitude faster in convergence.

Problem 4. A 13-Dimensional Problem Relevant to Computer Vision

The final problem is an application in the field of machine vision, and requires the generation of a discrete convolution mask, or spatial filter (Benke and Skinner, 1987). The mask is a square array of integer weights $w(m, n)$, and operates on a digitized image which takes the form of a rectangular array of pixels with grey levels expressed as integers $g(i, j)$. The discrete convolution operation (Pratt, 1978) generates a new image $h(i, j)$, the filtered image, according to the equation:

$$h(i,j) = \sum_{m=-N}^{N} \sum_{n=-N}^{N} w(m,n)\, g(i-m, j-n) \qquad (21)$$

where $1 \le i \le I,\ 1 \le j \le J; \quad \forall i, j,\ 0 \le g(i,j) \le G$

and G, I, J and N are parameters specific to each problem.

The filtered image is, of course, defined only over an array of size $(I - 2N) \times (J - 2N)$. The problem is to find the mask parameters $w(m, n)$ that maximise the normalised variance, or energy, $v(w)$ of the filtered image $h(i, j)$ when convolved with a given image $g(i, j)$ according to Equation (20), where

$$v(w) = \frac{\sum_{i=N+1}^{I-N} \sum_{j=N+1}^{J-N} h^2(i,j)}{(I-2N)(J-2N)\, G^2 \sum_{m=-N}^{N} \sum_{n=-N}^{N} w^2(m,n)} \qquad (22)$$

This energy is independent of multiplicative scaling changes in g and w, and for a uniformly patterned image is largely independent of the size of the image sample $(I \times J)$. The optimisation is the machine-vision analogue of finding a feature detector in the human visual system that is optimally matched to an observed pattern, and represents a *learning algorithm* for visual pattern recognition.

Table 4. Generation of Convolution Mask using Centroid Algorithm.

35 -12 5 -12 35 -66 -93 -36 -93 -66 38 91 43 91 38 51 -2 -100 -2 51 -22 33 -18 33 -22	52 4 -42 4 52 -1 21 -73 21 -1 13 -76 9 -76 13 42 43 -66 43 42 44 -6 -100 -6 44	74 -27 -96 -27 74 79 -22 -88 -22 79 72 -30 -85 -30 72 71 -33 -87 -33 71 65 -21 -100 -21 65
Iteration 1 Substitutions 1 Energy 3.3%	Iteration 10 Substitutions 10 Energy 44.5%	Iteration 100 Substitutions 55 Energy 99.4%

82 -32 -101 -32 82 82 -33 -100 -33 82 82 -32 -100 -32 82 83 -31 -100 -31 83 82 -32 -101 -32 82	81 -31 -100 -31 81 81 -31 -100 -31 81 81 -31 -100 -31 81 81 -31 -100 -31 81 81 -31 -100 -31 81
Iteration 1000 Substitutions 88 Energy 99.99%	Iteration 4492 Substitutions 94 Energy 100%

In the particular problem under consideration, the mask is a 5 x 5 matrix, that is $N = 2$, subject to the constraints

$$\sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) = 0$$

$$w(m,n) = w(-m,n) \quad ;-2 \le m \le 2, -2 \le n \le 2$$

$$\max(|w(m,n)|) \cong 100 \quad ;-2 \le m \le 2, -2 \le n \le 2$$

(These constraints are for illustration only — we have also optimised 21 x 21 masks with real coefficients and without any symmetry constraints, i.e. investigated a function in 439 dimensions.) The problem is therefore a search for a maximum of a function defined at approximately 10^{30} discrete points in a 13-dimensional space. An exhaustive search of all possible solutions is clearly impracticable and, for most images, it is not feasible to find the optimum mask by theoretical analysis. However, it is simple to prove that, when all the edges contained in the binary image are vertical, the optimum mask should have the element values shown in the last array in Table 4. There is one symmetry operator to take into account here, since the figure of merit, the filtered image energy, is invariant to sign inversion of the mask.

In order to generate random convolution masks efficiently within the problem constraints, sets of thirteen uniformly distributed random numbers were multiplied by a 13 x 25 constraint matrix to give the values for the mask elements (this technique is applicable in all cases where the constraints are linear). The results of a search for the optimum mask for a vertical pattern are shown in Table 4. The computing load can be assessed from the fact that the number of *function evaluations* is 3 x (*Iteration number*) —2, since n_s is equal to unity and the initial random mask generation is considered to be iteration number 1. This table also shows, as a percentage of its maximum possible value, the energy obtained using each mask.

It can be seen that the first estimate appears truly random within the constraints. After ten iterations, each of which caused a new matrix to be substituted for the current

Table 5. Generation of Convolution Mask using Constrained Random Search.

35	-12	5	-12	35
-66	-93	-36	-93	-66
38	91	43	91	38
51	-2	-100	-2	51
-22	33	-18	33	-22

Iteration 1
Substitutions 1
Energy 3.3%

-34	-30	8	-30	-34
6	71	100	71	6
-33	-3	-16	-3	-33
-54	39	100	39	-54
-24	-55	42	-55	-24

Iteration 10
Substitutions 4
Energy 33.4%

74	-100	6	-100	74
28	-100	-82	-100	28
76	11	-3	11	76
2	-68	20	-68	2
58	44	9	44	58

Iteration 100
Substitutions 5
Energy 34.7%

90	1	-100	1	90
75	-62	-100	-62	75
31	37	-95	37	31
62	30	-84	30	62
38	-82	-61	-82	38

Iteration 1000
Substitutions 9
Energy 71.3%

-66	54	29	54	-66
-75	-14	100	-14	-75
-57	-31	100	-31	-57
-37	63	72	63	-37
-58	23	95	23	-58

Iteration 10000
Substitutions 12
Energy 78.5%

best estimate, there is a significant grouping of negative values towards the centre. After 100 iterations, with 55 substitutions, the matrix is recognisably close to the theoretical form, with a deficit of less than 1% in the figure of merit. The experiment was continued to 1000 and 10 000 iterations, with the known ideal matrix finally appearing at number 4492. It was found that different starting points for the pseudo-random number generator gave the illustrated matrix or the sign-inverted version with about equal probability, and that around 200 iterations generally produced a figure of merit within 0.5% of the optimum value.

In order to demonstrate that the adaptive mechanism constitutes a significant part of the mask-generation process, the above trial was repeated with the mask averaging removed, i.e. with a purely random search for matrix elements within the constraints. The results of this exercise are shown in Table 5, where it can be seen that, after 10 000 iterations, the deficiency in the figure of merit is greater than 20% and individual parameters are subject to significant error. Comparison with Table 4 amply demonstrates the efficacy of the learning process, even taking into account that, in this case, the computing load per iteration is about three times greater for the centroid algorithm than for constrained random search. (For example, the performance of the centroid algorithm at 100 iterations, or 300 function evaluations, is far better than that of constrained random search after 10 000 function evaluations.)

4. DISCUSSION

The algorithm is most useful for problems which may be theoretically intractable, and where data may be discontinuous, partial or approximate, and perhaps very noisy. The probabilistic direct search approach appears to be particularly effective for functions which are non-linear, highly dimensional and which may contain plateaus or discontinuities. Swann (1974) notes that a direct search algorithm may often prove to be more reliable and stable than a gradient-based method, as no assumptions are made about the function, and much less preparation is required.

Unlike many traditional techniques, such as methods using Lagrange multipliers, the algorithm obviates the need for solving large sets of simultaneous equations. Only $(n_s + 3)(n_p + 1)$ feasible values are stored, where n_s is the number of symmetry operators and n_p is the number of parameters. In fact, the software for the optimisation of masks of size up to 21 x 21 is currently in use for vision research on a desk-top computer and incorporates an efficient new algorithm for fast iterative convolution.

For each iteration, the parameter set is selected from within the feasible region using a random process (this can be achieved very efficiently when the constraints are linear). Each parameter in the set is selected independently for each iteration, which is generally not true for gradient methods. This suggests that for problems of extremely high dimensionality, i.e. with thousands of variables, the algorithm can be implemented even more rapidly using parallel processing techniques.

It is clear, particularly from Table 4, that, as with the constrained random search strategy, the optimum can be found within an arbitrarily small error margin which is dependent on the number of iterations in the simulation, so that if the optimum itself is not found by either a random or learned input during one of the iterations, then the algorithm converges to it in a probabilistic sense.

Although the algorithm is suitable for general application, it may not converge as rapidly as an efficient gradient method in the case of, say, a unimodal function which is simple and smooth, continuous and differentiable, and of low dimensionality. Even for this limited class of problem, there are other considerations influencing overall efficiency. Himmelblau (1972) notes that the total cost associated with the solution of an optimisation problem must combine both the preparation time and the computer time used. He concludes that a solution obtained using a less efficient algorithm with easily prepared code may in fact cost much less than using a highly efficient code that requires many hours of preparation. He stresses that finding analytic derivatives is a major source of human error apart from being very time-consuming. It is clear that the preparation time required for the centroid algorithm is very small, being only slightly more than for a pure random search.

Swann (1974) observes that, in order to improve preparation time for a gradient-based method, a numerical approximation to differentiation can lead to truncation or cancellation errors which may nullify the underlying theory of the algorithm, resulting in slow or non-existent convergence. In many problems, a measure of efficiency based on the number of function evaluations may unfairly favour a gradient-based algorithm. For example, in the case of complex functions of high dimensionality, the time required to determine the point for the next function eva-

luation can often be many times greater than that required for the evaluation of the function itself (see Himmelblau, 1972). One reason for this is the requirement for the evaluation of the first and, if necessary, the second derivatives. As an example of preparation time for the centroid algorithm, using a shell program written in Pascal for an IBM AT compatible desk-top computer, only 15 minutes additional programming time was required to solve the Rastrigin problem (see Equation 19).

Most gradient methods are supported by optimality proofs, whereas most direct search methods are not, and it is generally not possible to derive convergence criteria for direct search (see Swann, 1974, and the review by Shimura, 1978, who cites a number of heuristic algorithms published without convergence proofs). Whilst we have not provided a rigorous analysis of the convergence behaviour of the centroid algorithm, considerations based on expectations for noise reduction suggest that the size of the error vector, $\Delta_k = \mathbf{X}_k - \mathbf{X}_{opt}$, between the unknown optimal solution and the best estimate after k iterations of the algorithm, is largely independent of the number of dimensions in the domain of $\mathbf{X}$. This is supported by an investigation of the variation of Δ during optimisation of a simple unimodal function (Equation 15).

5. SUMMARY

We describe a global direct search algorithm suitable for the optimisation of an arbitrary multivariate function. The centroid algorithm is noteworthy for its extreme simplicity, requiring very little coding and preparation time, and is applicable to functions which may be non-linear, noisy or highly dimensional, and may have discrete or continuous domains. A fundamental feature of the algorithm is its performance on multimodal functions, especially the difficult cases where approximate solutions are normally obtained by recourse to random search techniques.

The performance of the algorithm is compared with well known global algorithms including random search, various adaptive and guided random search methods, and stochastic automata. Faster and more accurate convergence to a global solution was demonstrated on benchmark problems (selected for the convenient comparison of global algorithms rather than for their complexity or the absence of other methods of solution), and also in some practical applications. The numerical examples include a number of non-linear multimodal functions, illustrating performance under conditions of noise, non-differentiability and high dimensionality, and with discrete search domains.

In addition to producing significantly improved global performance, the centroid algorithm has no arbitrarily selected parameter settings, such as step size or search direction, and appears to be particularly effective in applications where very little information is available on the function topography. If required, the convergence can be accelerated by systematically reducing the domain during the course of the optimisation, but at the expense of a diminishing global capability. Likewise, the precision can be improved by performing a second optimisation over a reduced neighbourhood. The procedure has potential applications in pattern recognition, adaptive filtering, computer vision and in the training of artificial neural networks.

6. ACKNOWLEDGEMENTS

We would like to express our appreciation to Dr Peter Beckwith and Dr John Ternan for many technical discussions concerning the algorithm and this paper. We would also like to thank Mr David Walker for discussions and for suggesting an efficient software implementation for the convolution experiment. Finally, we are grateful for the constructive criticism which was offered by the anonymous reviewers of this paper and which resulted in improved presentation and significant clarification.

7. REFERENCES

ACKLEY, D.H., HINTON, G.E. and SEJNOWSKI, T.J. (1985): A Learning Algorithm for Boltzmann Machines, *Cognitive Science,* Vol. 9, pp. 147-169.

BEKEY, G.A. and UNG, M.T. (1974): A Comparative Evaluation of Two Global Search Algorithms, *IEEE Trans. Syst. Man, Cybern.,* Vol. SMC-4, No. 1, pp. 112-116.

BELTRAMI, E.J. and INDUSI, J.P. (1972): An Adaptive Random Search Algorithm for Constrained Minimisation, *IEEE Trans. Computers,* Vol. C-21, No. 9, pp. 1004-1008.

BENKE, K.K. and SKINNER, D.R. (1987): Segmentation of Visually Similar Textures by Convolution Filtering, *Australian Computer J.,* Vol. 19, No. 3, pp. 134-139.

BOX, M.J. (1965): A New Method of Constrained Optimisation and a Comparison with Other Methods, *Computer J.,* Vol. 8, pp. 42-52.

DEVROYE, L.C. (1976): A Class of Optimal Performance Directed Probabilistic Automata, *IEEE Trans. Syst. Man, Cybern.,* Vol. SMC-6, No. 11, pp. 777-783.

EMSHOFF, J.R. and SISSON, R.L. (1970): *Design and Use of Computer Simulation Models,* New York: Macmillan Publishing Co.

FU, K.S. and NIKOLIC, Z.J. (1966): On Some Reinforcement Techniques and their Relation to the Stochastic Approximation, *IEEE Trans. Automat. Control,* Vol. AC-11, No. 4, pp. 756-758.

GLORIOSO, R.M. and COLON OSORIO, F.C. (1980): *Engineering Intelligent Systems,* Massachusetts: Digital Press.

GOTTFRIED, B.S. and WEISMAN, J. (1973): *Introduction to Optimisation Theory,* Englewood Cliffs, New Jersey: Prentice-Hall.

HILLIER, F.S. and LIEBERMAN, G.J. (1974): *Operations Research,* San Francisco: Holden-Day Inc.

HIMMELBLAU, D.M. (1972): *Applied Nonlinear Programming,* New York: McGraw-Hill Book Co.

HINTON, G.E. and SEJNOWSKI, T.J. (1983): Optimal Perceptual Inference, *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition,* pp. 448-453, Silver Spring, Maryland.

HOOKE, R. and JEEVES, F.A. (1959): Direct Search Solution of Numerical and Statistical Problems, *Journal of the ACM,* Vol. 8, p. 212.

JARVIS, R.A. (1969): Adaptive Global Search in a Time-Invariant Environment using a Probabilistic Automaton, *Proc. IREE, Australia,* pp. 210-226.

JARVIS, R.A. (1975): Optimisation Strategies in Adaptive Control: A Selective Survey, *IEEE Trans. Syst. Man, Cybern.,* Vol. SMC-5, pp. 83-94.

KIRKPATRICK, S., GELATT, C.D. Jr. and VECCHI, M.P. (1983): Optimisation by Simulated Annealing, *Science,* Vol. 220, pp. 671-680.

KOHONEN, T. (1984): *Self-Organisation and Associative Memory,* Berlin: Springer-Verlag.

LAWRENCE, J.P. and STEIGLITZ, K. (1972): Randomised Pattern Search, *IEEE Trans. Computers,* Vol. C-21, No. 4, pp. 382-385.
McMURTRY, G.J. and FU, K.S. (1966): A Variable Structure Automaton used as a Multimodal Searching Technique, *IEEE Trans. Autom. Control,* Vol. AC-11, No. 3, pp. 379-387.
NARENDRA, K.S. and LAKSHMIVARAHAN, S. (1977): Learning Automata — A Critique, *J. Cybern. Inform. Sci.* (USA), pp. 53-65.
NARENDRA, K.S. and THATHACHAR, M.A.L. (1974): Learning Automata — A Survey, *IEEE Trans. Syst. Man, Cybern.,* Vol. SMC-4, No. 4, pp. 323-334.
NELDER, J.A. and MEAD, R. (1965): A Simplex Method for Function Minimisation, *Computer J.* Vol. 17, p. 308.
OOMEN, B.J. and HANSEN, E. (1984): The Asymptotic Optimality of Discretised Linear Reward-Inaction Learning Automata, *IEEE Trans. Syst. Man, Cybern.,* Vol. SMC-14, No. 3, pp. 542-545.
POLOVINKIN, A.I. (ed) (1981): Automation of Searching Design, *Radio i Sviaz,* p. 344, Moscow.
PRATT, W.K. (1978): *Digital Image Processing,* New York: John Wiley and Sons.
RANGAIAH, G.P. and KRISHNASWAMY, P.R. (1987): Multiple Minima in Chemical Engineering Applications of Optimisation, *Internat. Conf. on Optimiz. Techniques and Applic.* pp. 436-442, Singapore.
SHAPIRO, I.J. and NARENDRA, K.S. (1969): Use of Stochastic Automata for Parameter Self-Optimisation with Multimodal Performance Criteria, *IEEE Trans. Syst. Sci., Cybern.,* Vol. SSC-5, pp. 352-360.
SHIMURA, M. (1978): Learning Procedures in Pattern Classifiers — Introduction and Survey, *Proc. 4th Intern. Joint Conf. on Pattern Recog.,* pp. 125-138, Kyoto, Japan.
SPENDLEY, W., HEXT, G.R. and HIMSWORTH, F.R. (1962): Sequential Applications of Simplex Designs in Optimisation and Evolutionary Operation, *Technometrics,* Vol. 4, p. 441.
SWANN, W.H. (1974): Constrained Optimisation by Direct Search, appears as Ch. 7 in GILL, P.E. and MURRAY, W., Eds.: *Numerical Methods for Constrained Optimisation,* London: Academic Press.
TAHA, H.A. (1976): *Operations Research,* New York: Macmillan Publishing Co.
TSYPKIN, Ya. Z. and POZNYAK, A.S. (1977): Learning Automata, *J. Cybern. Inform. Sci. (USA),* pp. 128-161.
UOSAKI, K., IMAMURA, H., TASAKA, M. and SUGIYAMA, H. (1970): A Heuristic Method for Maxima Searching in the case of Multimodal Surfaces, *Technol. Rept. of Osaka Univ. Japan,* Vol. 20, pp. 337-344.
WISWANATHAN, R. and NARENDRA, K.S. (1973): Stochastic Automata Models with Applications to Learning Systems, *IEEE Trans. Syst. Man, Cybern.,* Vol. SMC-3, pp. 107-111.

BIOGRAPHICAL NOTES

Kurt Benke received BSc and MSc degrees in Physics from the University of Melbourne, and a PhD in Mathematics and Computing (for studies in computer vision) from Deakin University. He also holds a postgraduate diploma in Applied Statistics from the Royal Melbourne Institute of Technology. He was originally employed with the Kodak Research Laboratory, where he gained research experience in theoretical and experimental optical physics, X-ray physics, and electromagnetic scattering. He is currently a Senior Research Scientist with the DSTO Materials Research Laboratory, and has been actively involved in research in human and machine vision, robotic inspection, pattern recognition and sonar surveillance.

David Skinner received a BSc in Physics from the Queen's University of Belfast, Northern Ireland, in 1957, and until 1960 worked with Short Brothers and Harland on the design of aircraft transducers. Since then he has been with the DSTO Materials Research Laboratory, Melbourne. He is a Principal Research Scientist and Task Manager for research in sidescan sonar image analysis. His research experience includes work in theoretical and experimental xerography, laser physics and its applications, remote colorimetry, and in visual camouflage and other countersurveillance. Research interests include digital image processing, visual perception, optical physics and adaptive systems.

CALL FOR PAPERS

SPECIAL ISSUE ON ALGORITHMS

The May 1992 issue of the Journal will focus on the theme of Algorithms. Papers are sought on the broad spectrum of topics in Algorithms, including but not limited to:

* complexity of algorithms and data structures
* parallel algorithms
* computational geometry
* VLSI algorithms
* randomised algorithms

Papers should be addressed to the broad Australian Computer Journal audience, and should not be too narrow, except when giving examples to illustrate broad concepts.

Papers must be presented in conformance with this Journal's style, including Harvard-style citations and reference list, and excluding footnotes and endnotes.

Potential authors are requested to register their interest with either of the Guest Editors by 12 August 1991 indicating a provisional title and a (non-binding) 50-200 word outline of the intended contribution. A package of Instructions for Authors will be provided by return mail.

Dr Peter Eades
Department of Computer Science
The University of Queensland
St Lucia Queensland 4072 Australia
Telephone +61 07 365 2904
Facsimile +61 7 365 1999

Dr Alistair Moffatt
Department of Computer Science
University of Melbourne
Parkville Victoria 3052 Australia
Telephone +61 03 344 5229
Facsimile +61 3 348 1184

View publication stats