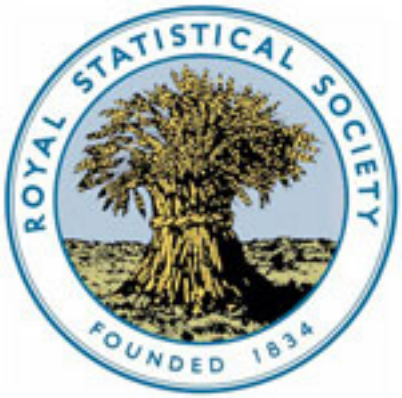


The Statistician (1995)
44, *No.* 2, *pp.* 241–257

Optimization using simulated annealing

By S. P. BROOKS and B. J. T. MORGAN†

University of Kent, Canterbury, UK

[Received July 1993. Revised November 1994]

SUMMARY

Much work has been published on the theoretical aspects of simulated annealing. This paper provides a brief overview of this theory and provides an introduction to the practical aspects of function optimization using this approach. Different implementations of the general simulated annealing algorithm are discussed, and two examples are used to illustrate the behaviour of the algorithm in low dimensions. A third example illustrates a hybrid approach, combining simulated annealing with traditional techniques.

Keywords: Boltzmann distribution; Cauchy likelihood; Hybrid algorithm; Markov chains; Maximum likelihood; Normal mixture models; Optimization; Simulated annealing

1. Introduction

At the heart of simulated annealing is an analogy with thermodynamics, specifically with the way that metals, and some liquids, cool and crystallize. If the temperature is lowered slowly then this cooling process is called annealing, and a characteristic property of annealing is lowering the temperature gradually, *in stages*, allowing thermal equilibrium to be attained at each stage. At high temperatures molecules are free to move, but as the temperature decreases thermal mobility is lost, and the molecules tend to line themselves up in a rigid structure. This rigid state is, in fact, a state of minimum energy, and, as long as the temperature is decreased slowly, nature is almost certain to find it. If the temperature is decreased rather more rapidly, this state may not be achieved, and a polycrystalline or amorphous state with higher energy may be found instead.

Most function minimization routines would be analogous to this rapid cooling method. In the case where multiple optima are present it is easy for optimization algorithms to find a local optimum, corresponding to the amorphous state in nature. Simulated annealing, however, is analogous to decreasing the temperature slowly, allowing ample time for the redistribution of the molecules. Hence, we regard simulated annealing as a minimization algorithm based on nature. For function optimization, we would equate function values with energy levels and parameter settings with the positions of molecules. The role of the temperature in optimization will be clear from Section 2.

We begin by discussing the development of the simulated annealing algorithm and show how the algorithm can be formulated as a Markov chain. In Section 3 we discuss different implementations of the algorithm, before studying its behaviour in low dimensions in Section 4. Finally, in Section 5 we consider a hybrid approach which combines an annealing algorithm with a more traditional optimization algorithm, and we show how the hybrid can outperform either of these algorithms on their own.

†*Address for correspondence*: Institute of Mathematics and Statistics, Cornwallis Building, University of Kent, Canterbury, Kent, CT2 7NF, UK.
E-mail: B.J.T.Morgan@ukc.ac.uk

 0039–0526/95/44241

2. The annealing algorithm

2.1. *Theory of simulated annealing*

The annealing process can be described as follows. We consider a succession of decreasing temperatures, starting from some maximum T_0. At each temperature T, the system is allowed to reach thermal equilibrium in which the probability that the system is in some state with energy E is given by the Boltzmann distribution

$$P(E = k) = \frac{1}{Z(T)} \exp\left(-\frac{k}{k_b T}\right) \tag{1}$$

where $Z(T)$ is a normalization function and k_b is known as the Boltzmann constant. This is the prescription of the Boltzmann theorem of statistical physics (originally Boltzmann (1877), but to be found in any standard statistical physics text). As T decreases, the range of the Boltzmann distribution concentrates on states with the lowest energy, as high energy states become increasingly unlikely. Eventually, T becomes so low that the system 'freezes', and if the temperature is lowered sufficiently slowly this frozen state will have minimum energy.

An algorithm by Metropolis *et al.* (1953) for the efficient simulation of atoms in equilibrium at a given, fixed, temperature was based on this process: given a current state of the system with energy E_0, we choose a new state by displacing a randomly chosen particle. If this new state has energy E, and $E < E_0$, then the system remains in this new state, and then another new state is selected as before. However, if $E \geqslant E_0$ then the probability of remaining in this new state is given by

$$\exp\left\{-\frac{(E - E_0)}{k_b T}\right\}.$$

This acceptance rule is known as the *Metropolis criterion.* When this process is repeated, the system eventually reaches a state of equilibrium for that temperature, in which the probability distribution of the accepted points is given by the Boltzmann distribution of equation (1)—see Metropolis *et al.* (1953). This process is known as the *Metropolis algorithm.*

The Metropolis algorithm is concerned with only a single fixed temperature. It was generalized by Kirkpatrick *et al.* (1983), who introduced an *annealing schedule* which defines how the temperature is reduced. Beginning with a high initial temperature T_0, the Metropolis algorithm is followed until equilibrium is achieved. The temperature is then decreased, according to the annealing schedule, and the Metropolis algorithm is then followed at this new temperature until a new equilibrium is achieved and the temperature is decreased once more. This process is repeated until the system freezes. If the annealing schedule is sufficiently slow, then the system will freeze in a state of minimum energy, corresponding to the global minimum of our objective function. Convergence of this algorithm has been proved by reformulating the algorithm as a Markov chain.

2.2. *Simulated annealing as a Markov chain*

Much is now known about the convergence and long-term behaviour of annealing algorithms using Markov chain theory—see, for example, Mitra *et al.* (1986), Lundy and Mees (1986) and Ingber (1994). We can reformalize the annealing algorithm as a Markov chain by defining a set of conditional probabilities $p_{ij}(T)$, for all i, j, where $p_{ij}(T)$ is the probability that the chain (or system) moves from energy state i to state j once the chain has reached equilibrium at temperature T. If T were to be kept constant, then the Markov chain is homogeneous, and its *transition matrix* can be written as

$$p_{ij}(T) = u_{ij}(T)\, m_{ij}(T) \qquad j \neq i$$

and

$$p_{ii}(T) = 1 - \sum_{j \neq i} u_{ij}(T)\, m_{ij}(T)$$

where $u_{ij}(T)$ is the *generation probability*, i.e. the probability of generating state j as the next state after state i at temperature T, and $m_{ij}(T)$ is the *acceptance probability*, based on the Metropolis criterion, assigning a probability to the event that state j is accepted as the next state after state i, at temperature T. The corresponding matrices $U(T)$ and $M(T)$ are called the generation and acceptance matrices respectively. It is this reformalization of simulated annealing as a Markov chain that enables us to analyse its behaviour, particularly in the limiting case.

If $\Psi(k)$ is the state of the chain after k transitions, and Θ is the set of all points at which the global minimum is achieved, then we are interested in finding the smallest value of k such that

$$P\{\Psi(k) \in \Theta\} = 1. \tag{2}$$

It can be shown that, under certain conditions on the matrices $U(T_i)$ and $M(T_i)$ and also on the convergence of the sequence of temperatures $\{T_i\}$, equation (2) holds asymptotically, i.e.

$$\lim_{k \to \infty} [P\{\Psi(k) \in \Theta\}] = 1;$$

see Geman and Geman (1984), Aarts and Korst (1988) and Bertsimas and Tsitsiklis (1993). Therefore, under these conditions, the algorithm is guaranteed to converge, but not necessarily in finite time.

However, it might be of more interest to find the smallest value of k such that

$$P\{\Psi(k) \in \Theta_\varepsilon\} = 1 \tag{3}$$

where Θ_ε is a set of solutions 'close' to the global optimum in some sense. This idea has particular application to the ideas discussed in the final section of this paper, but no work has yet been published in this area.

3. Implementations of the algorithm

3.1. *General simulated annealing algorithm*

We may implement a general simulated annealing algorithm by the following steps.

Step 1: beginning at an initial temperature T_0, we pick an initial set of parameter values with function value E.

Step 2: randomly select another point in the parameter space, within a neighbourhood of the original, and calculate the corresponding function value.

Step 3: compare the two points in terms of their function value, using the Metropolis criterion as follows. Let $\Delta = E_{\text{new}} - E_{\text{old}}$, and move the system to the new point if and only if a random variable U, distributed uniformly over (0, 1), satisfies

$$U \leqslant \exp(-\Delta/T)$$

when T is the current temperature, or equivalently

$$E_{\text{new}} \leqslant E_{\text{old}} - T \log U.$$

Thus $E_{\text{new}} - E_{\text{old}}$ is compared with an exponential random variable with mean T. Note that we always move to the new point if its corresponding function value is lower than that of the old point, and that at any temperature there is a chance for the system to move 'upwards'. Accepting a point, we equate with *success*.

Step 4: whether the system has moved or not, repeat steps 2–3. At each stage compare the function value of new points with the function value of the present point until the sequence of accepted points is judged, by some criterion, to have reached a state of equilibrium.

Step 5: once an equilibrium state has been achieved for a given temperature, the temperature is lowered to a new temperature as defined by the annealing schedule. The process then begins again from step 2, taking as initial state the point following the last iteration of the algorithm, until some stopping criterion is met, and the system is considered to have *frozen.*

Since we continue steps 2–3 until a state of equilibrium is attained, the starting values in step 1 are arbitrary, and have no effect on the solution.

Given this algorithm there are numerous ways in which it may be implemented. In the following subsections, we shall discuss two methods of implementation, and refer to several other approaches based on this general annealing algorithm.

3.2. *Simple implementation*

We devised an algorithm based on that given by Press *et al.* (1989), p. 326, for the solution of the travelling salesman problem. Although this algorithm was intended only for discrete problems, its use for continuous variables can be justified by considering each variable up to only a small number of significant figures. It was found experimentally, however, that it was not necessary to discretize the variables in this way, and that the algorithm was considerably faster and more reliable without this constraint. Of course, this still leaves the variables considered only to machine accuracy and thus, essentially, discrete.

If we wish to minimize a function $f(x_1, x_2, \ldots, x_p)$, then this particular implementation of the algorithm works as follows: at step 2 of the general annealing algorithm, the new point is chosen by first selecting one of the p variables at random, and then randomly selecting a new value for that variable within the bounds set for it by the problem at hand. Thus the new point takes the same variable values as the old point except for one. This is one way to choose new points so that they are in a neighbourhood of the old point; other methods are discussed later. At step 4, the repetition of steps 2–3 occurs exactly N times, after, say, s successful moves from one point to another. N should, if possible, be selected so that it could be reasonably assumed that the system is in equilibrium by that time. At step 5, if $s > 0$ then we decrease the temperature by letting T become ρT, where $0 \leqslant \rho \leqslant 1$ defines the annealing schedule, and then begin again. If, however, $s = 0$ then we can consider the system to have frozen, as no successful moves were made, and the algorithm stops.

This implementation was found to be both fast and reliable for most of the functions that we considered. The convergence time was found to be highly problem dependent, as the complexity of the problem directly affected the number of temperature reductions necessary for convergence. This number was commonly found to be several hundred. It was also highly dependent on the values of the parameters ρ, N and T_0. A large N gives an accurate solution, but at the expense of convergence time. Doubling the value of N more than doubled the execution time. Increasing ρ increases the reliability of the algorithm in reaching the global optimum, and corresponds to a slower cooling of the system. The value of T_0 is also important. T_0 must be sufficiently large for any point within the parameter space to have a reasonable chance of being visited, but if it is too large then too much time is spent in a 'molten' state. In practice it may be necessary to try the algorithm for several values of T_0 before deciding on a suitable value. This is the algorithm that we shall be referring to for the remaining sections of this paper, but first we shall introduce a popular alternative.

3.3. *Alternative, more complex, implementation*

An alternative implementation was suggested by Corana *et al.* (1987). In this version of the algorithm, new points are chosen to be within a neighbourhood of the old points by introducing a step vector, which associates each variable x_i with a maximum step value v_i. This determines how far the new point can be from the old in the direction of co-ordinate i.

Then a new point is generated by altering the first variable and this is either accepted or rejected. Then the second variable is altered and is accepted or rejected. This continues until all p variables have been altered and thus p new points have been successively accepted or rejected according to the Metropolis criterion. This process is repeated N_s times, after which the step vector v is adjusted so that approximately half of all of the points selected are accepted. This whole cycle is then repeated N_r times, after which the temperature is decreased from T to ρT. Termination of the algorithm occurs when the sequence of points reached after each $N_s N_r$ step cycle is such that their average function value reaches a stable state.

This algorithm is considerably more complex than the first. Introducing a step vector as above can allow the algorithm to shape the space within which it may move, to that within which it ought to be, as defined by the objective function. Thus the algorithm may create valleys down which to travel towards the optimum. However, this has the drawback that, when these valleys are not in the direction of a co-ordinate axis, convergence can be extremely slow. In practice we found that this algorithm reliably converged to the global minimum for simpler examples such as those given in Section 3, but that more complex problems led to unreliable results. It is for this reason that we shall refer only to the implementation in Section 3.2 for the remainder of this paper.

There are numerous other approaches to the implementation of simulated annealing algorithms. Among the more interesting and recent of these are fast simulated annealing, where distributions other than the Boltzmann distribution are used (see Szu and Hartley (1987)), simulated reannealing, where the ranges of the parameters are adaptively altered as the algorithm progresses (see Ingber (1992)) and a segmented algorithm suggested by Atkinson (1992). Ingber (1994) provides a brief review of the most recent work on many algorithms based around that of simulated annealing, together with areas of their application. Osman and Christofides (1994) described a non-monotonic temperature change scheme, which allows periodic *raising* of temperature in addition to the standing cooling. For a broader review, see Reeves (1993), who covers topics such as Tabu search and genetic algorithms. Tabu search methods can be quite complex, involving a memory structure which is regularly updated and prevents certain directions from being explored.

4. Behaviour of annealing algorithm in low dimensions

In this section we shall consider two simple examples of functions with multiple optima and show how traditional algorithms can have considerable difficulty in finding the global minimum. We shall then show how an annealing algorithm overcomes such problems by examining its behaviour as the algorithm converges to the minimum.

4.1. *One-dimensional example*

We shall take as our first example the Cauchy density function

$$f(x) = \frac{\beta}{\pi\{\beta^2 + (x - \alpha)^2\}}$$

for $-\infty \leqslant x \leqslant \infty$, $\beta \geqslant 0$, $-\infty \leqslant \alpha \leqslant \infty$.

The log-likelihood surface for this density has been shown to give various iterative methods of maximum likelihood estimation considerable difficulty when estimating α for some fixed values of β (Barnett, 1966). If we take a random sample $x_1, x_2, \ldots, x_n$, then the log-likelihood can be written as

$$l = n \log \beta - \sum_{i=1}^{n} \log\{\beta^2 + (x_i - \alpha)^2\} - n \log \pi.$$

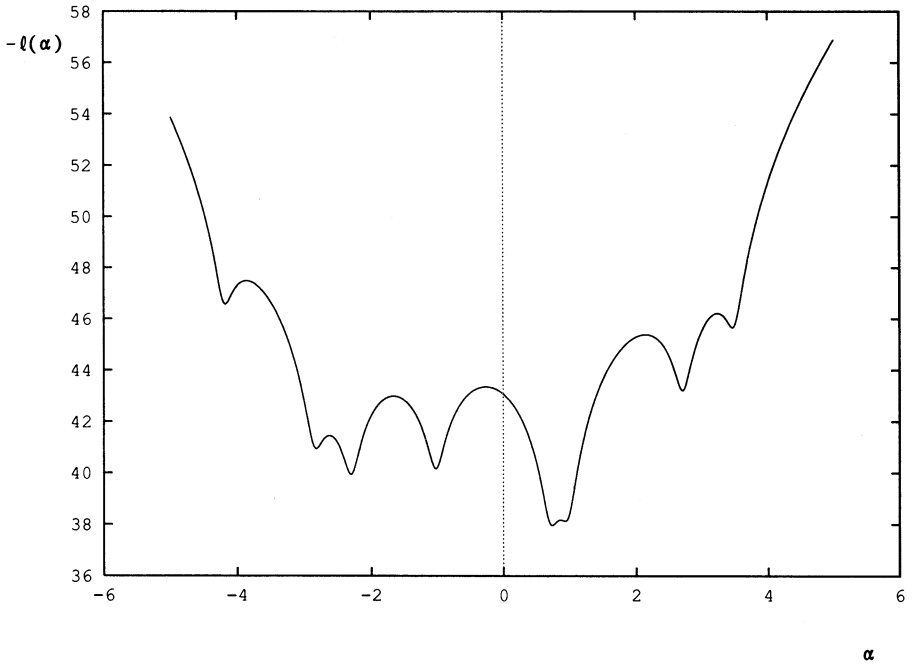


Fig. 1. Cauchy log-likelihood with eight data points (the minimum is at $\alpha = 0.7327$)

With β fixed, the maximum likelihood estimate for α is that which minimizes

$$l(\alpha) = \sum_{i=1}^{n} \log\{\beta^2 + (x_i - \alpha)^2\},$$

and if we fix $\beta = 0.1$ then for the random sample

$$(-4.20,\ -2.85,\ -2.30,\ -1.02,\ 0.70,\ 0.98,\ 2.72,\ 3.50)$$

we obtain the log-likelihood given in Fig. 1.

Depending on the level of competence of the programmer, the Numerical Algorithms Group (NAG) manual recommends two algorithms for the minimization of general multi-parameter functions (Numerical Algorithms Group, 1991). These were used to minimize the function of Fig. 1. The first is a quasi-Newtonian (QN) algorithm implemented in Fortran by the NAG library routine E04JAF. The second algorithm is a sequential quadratic programming (SQP) algorithm (see Gill *et al.* (1981), p. 237) implemented by the NAG routine E04UCF.

To test and compare these algorithms, we randomly generated 1000 starting points within the range $[-6, 6]$ and recorded the points at which the algorithms converged for each of those starting points. The QN algorithm correctly identified the global minimum for only 263 of the starting points, and each of the local minima was given as the global solution for at least 40 of the starting points. Similar results were found with the SQP algorithm, which correctly identified the global minimum for only 341 starting points.

The NAG manual recommends that, for a one-dimensional function, routine E04ABF should be used. This routine is based on Powell's quadratic interpolation algorithm (see Bunday (1984), p. 20). Although this algorithm correctly identified the global minimum for the Cauchy likelihood for this particular set of data, it was also found to be similarly unreliable for other Cauchy data sets. Thus the standard algorithms for function minimization appear

to have considerable difficulty in achieving the global minimum owing to the high number of local minima within the parameter range.

The annealing algorithm was then used in a similar manner to produce the minimum value, beginning from 1000 randomly chosen starting points. To make a fair comparison, the annealing parameters T_0, N and ρ were set to values giving an expected convergence time that was similar to those of the other routines. For this example, experience suggests suitable values to be $T_0 = 10.0$, $N = 300$ and $\rho = 0.95$. This produces a fairly slow cooling rate and with this value of N, which experience suggests is quite low, we have a high chance of stopping the algorithm before the system has frozen. Thus we would not expect the solution to be accurate to many decimal places. We found that 100% of the solutions lay in the range [0.70, 0.94] and that only 1% were outside the interval [0.70, 0.80] corresponding to the tiny well containing the true minimum. Thus our annealing algorithm is considerably more reliable than the two general QN and SQP algorithms, but the accuracy of the solution is correct to only the first decimal place.

Fig. 2 shows all the points which were both selected and accepted by the Metropolis criterion, during a typical execution of the annealing algorithm. Note the high concentration of accepted points around the bottom of the large well containing both the global minimum and the local minimum closest to it. Note also that each of the local minima has been visited by the algorithm and thus each has been considered and subsequently rejected in favour of the global minimum. The outline of the function profile is quite distinct but, of course, the exploration of the parameter space was by no means exhaustive.

Fig. 3 represents the successful jumps in another way. Fig. 3(a) shows

$$\frac{l(\alpha_{\text{opt}}) - l(\alpha_i)}{l(\alpha_{\text{opt}})}$$

for the ith new accepted point α_i, plotted against the number r_i of randomly selected points already considered, whether accepted or not. Here α_{opt} denotes the value of α corresponding

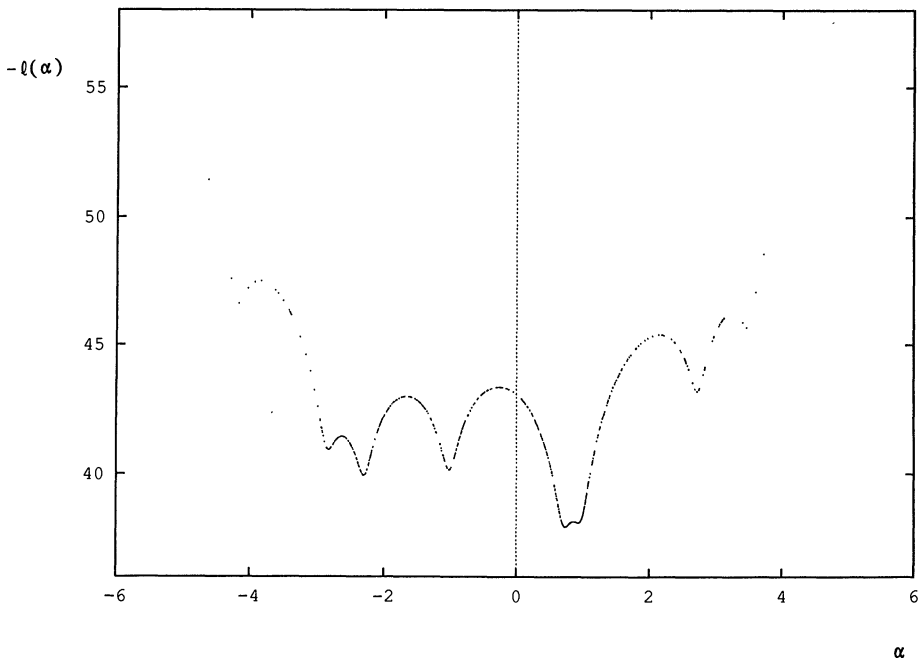
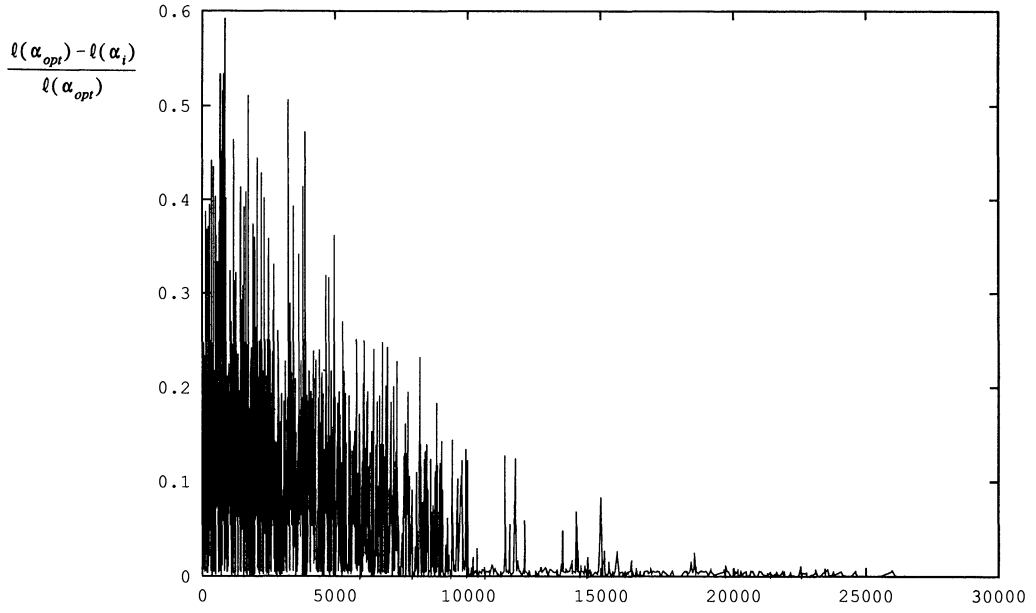
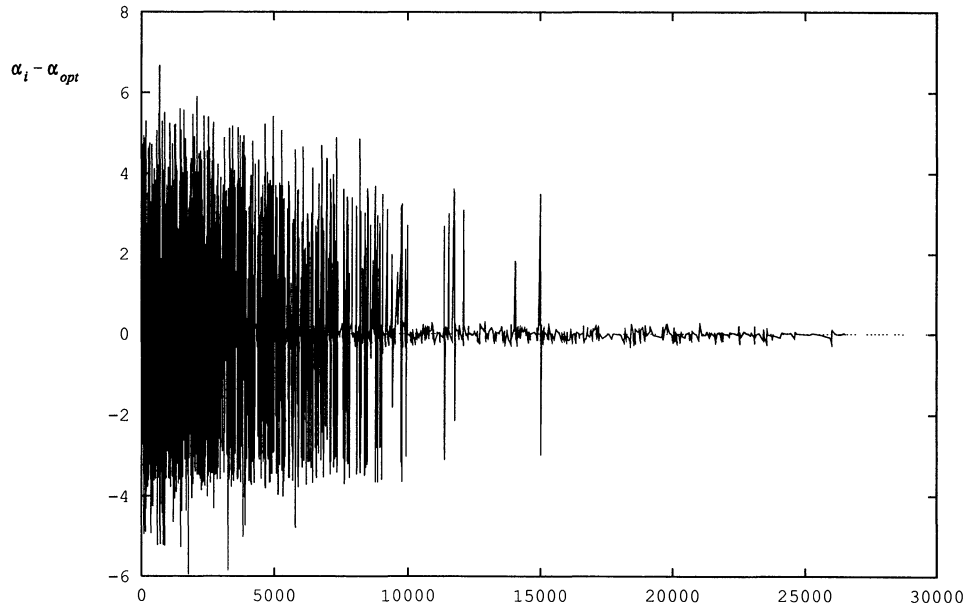


Fig. 2. Values of $-l(\alpha)$ for points accepted by the annealing algorithm ($T_0 = 6$; $N = 250$; $\rho = 0.92$)



(a)



(b)

Fig. 3. (a) Convergence of the function value to the minimum; (b) convergence of α to the minimum

to the global optimum. We can see that, initially, high function values are obtained with some regularity, but as the temperature is decreased the function value of accepted points decreases, as does the likelihood of an uphill move. We can see that beyond $r_i = 16\,000$ the function values are generally within 1% of the global minimum.

Fig. 3(b) shows $\alpha_i - \alpha_{opt}$ plotted against r_i, and we can see how the algorithm has jumped from one point to the next in a spatial sense. As in Fig. 3(a), we can see that the algorithm explores points a considerable distance from the minimum for low r_i, but that beyond $r_i = 15\,000$ the algorithm has identified an area containing the minimum and restricts its search to points within an ever decreasing neighbourhood of the true minimum.

These figures reveal the efficiency of the algorithm for the current values of the annealing parameters. The erratic behaviour evident in both graphs for low r_i corresponds to the system initially being in a molten state. If the system is left in this state too long, then the initial temperature T_0 may have been too high, and time may have been wasted unnecessarily. Similarly if, initially, the algorithm does not explore the parameter space fully at first, then the value of T_0 may have been too low. The length of the tail, evident in both graphs, tells us how efficient the value of N was in determining the frozen state of the system. If the tail is too long then the algorithm could have been stopped earlier by decreasing N. If the tail is too short then there may not be enough evidence that the system has frozen, and our results may become unreliable. Thus these graphs are useful diagnostic tools. We have found it simplest not to alter N during any single optimization but an adaptive approach, based on the diagnostics above, is an obvious alternative.

4.2. *Two-dimensional example*

This example is taken from Bohachevsky *et al.* (1986). The function is given by

$$f(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7 \qquad \text{for } -1 \leqslant x, y \leqslant 1.$$

This function has its minimum at (0, 0). It is symmetric and has a high number of local minima and saddlepoints, illustrated in Fig. 4.

As before, we test SQP and QN algorithms on this surface. To see how the algorithms performed with this function, we randomly selected 100 000 starting points, uniformly over the square of Fig. 4 and recorded both those that led to the correct solution and those that did not. Fig. 5 indicates those starting points which led to the global minimum.

Similar plots of points for which the algorithms failed to converge confirm that the entire parameter space was uniformly covered, and that any features within these graphs are features of the problem at hand, and not of the random number generator used to generate the starting points. We can see that the graphs are essentially symmetrical about the axes, as we would expect. From Fig. 5(a) we can see that fewer than 20% of the starting points lead to convergence of the QN algorithm to the global minimum, and that these are mainly within a central rectangle defined by the four peaks surrounding the global minimum. Beyond this rectangle, the algorithm searches for other valleys, though in some particularly steep places local valleys can be overshot. Fig. 5(b) shows similar results for the SQP algorithm. Of the two, the SQP algorithm performs better. Thus traditional methods of minimizing this function have difficulty in converging to the true minimum.

The annealing algorithm performed considerably better. Fig. 6 shows the points to which our annealing algorithm converged, given 1000 random starting points within the parameter space and for two different annealing parameter settings. We can see how increasing N, T_0 and ρ increases the accuracy of the final solution.

The annealing algorithm is very reliable in that all the solutions are within the central well containing the global minimum. We can also see that the final solution is accurate only to the first decimal place. Accuracy can be improved by altering the parameters of the algorithm, but at the expense of the time taken for convergence.

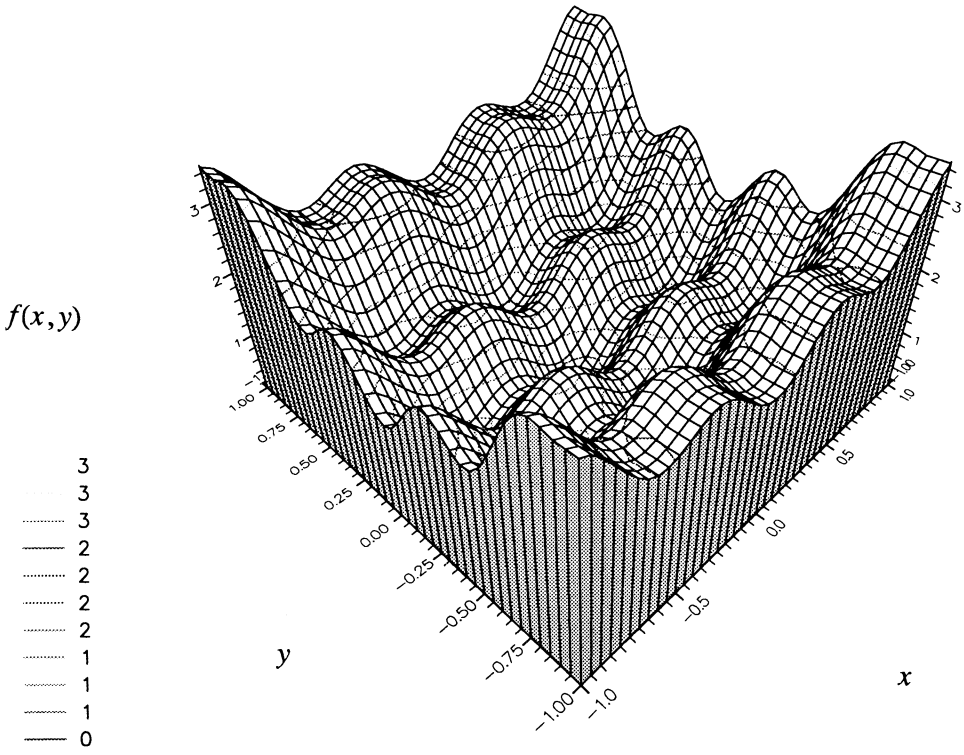


Fig. 4. Isometric projection of $f(x, y)$

4.3. *Accuracy*

From the two examples discussed so far we can draw the following conclusions. The annealing algorithm appears to be very reliable, in that it always converged to within a neighbourhood of the global minimum. The size of this neighbourhood can be reduced by altering the parameters of the algorithm, but this can be expensive in terms of time.

In most cases we would like to find the minimizing solution to several decimal places. It is clear that the annealing algorithm could produce results with such accuracy, but that the execution time would be prohibitive. A more traditional algorithm can produce solutions to machine accuracy but can have considerable difficulty in finding the correct solution. The next section discusses methods for combining an annealing algorithm with a more traditional algorithm to form a hybrid algorithm which is both reliable and accurate.

5. Hybrid algorithm

5.1. *Alternative procedures*

In this section we shall discuss a hybrid algorithm consisting of two distinct components. The first component is an annealing algorithm which is used to produce starting points for the second component, a more traditional algorithm. Simkin and Trowbridge (1992) and Drago *et al.* (1992) discussed such an approach in which the annealing component is stopped prematurely, after say N_t temperature reductions, producing a single starting point for the second component, by using the point at which the annealing algorithm was stopped. The difficulty with this approach is knowing when to stop the annealing algorithm so that the final point is within the well containing only the global minimum. This returns us to the

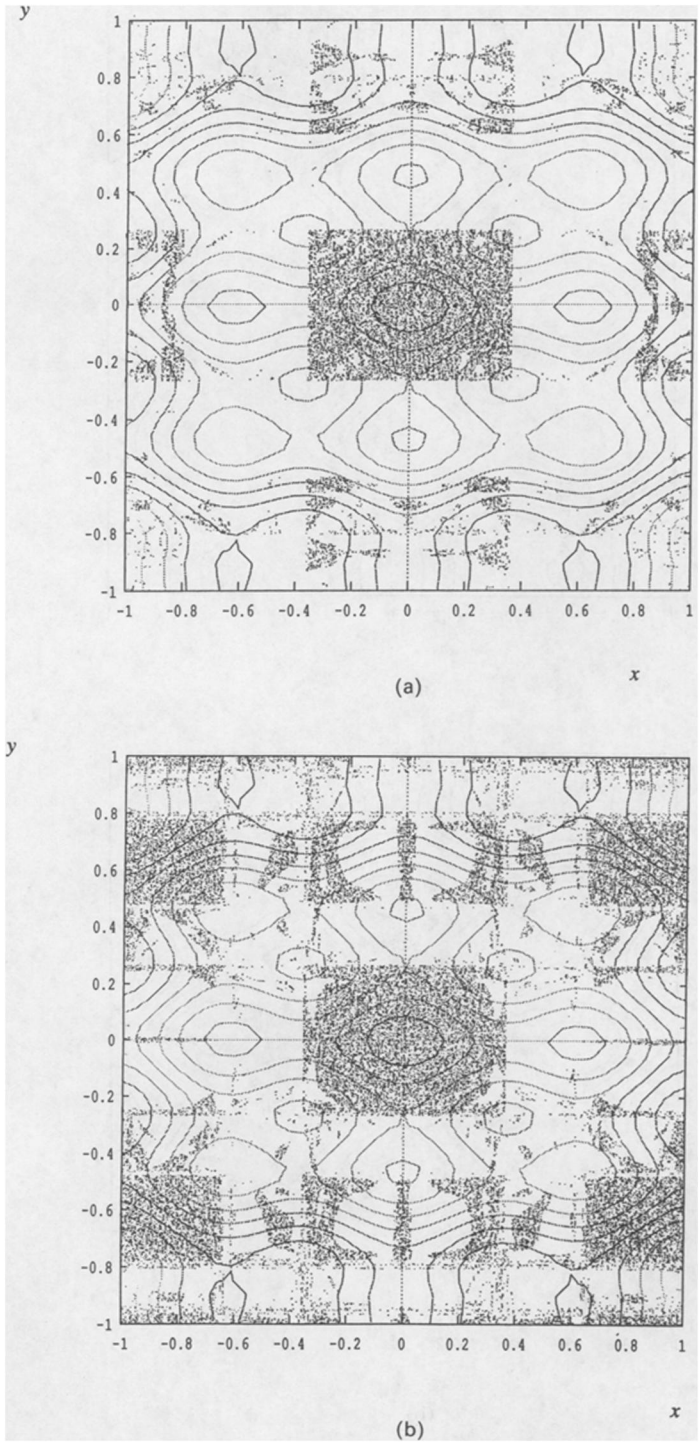
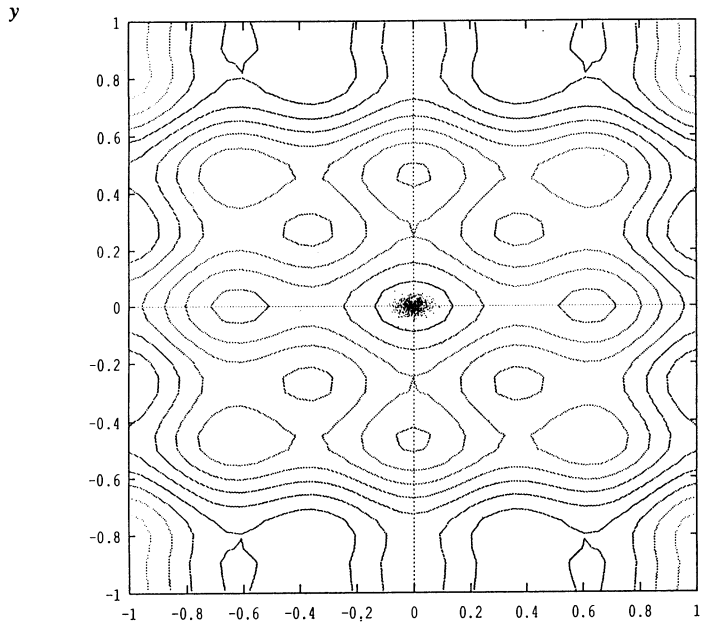
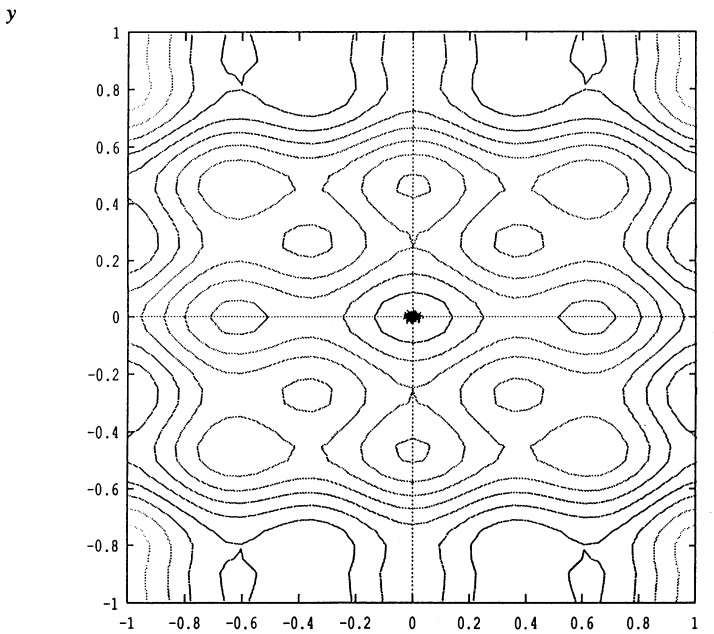


Fig. 5. (a) Points for which the QN algorithm converged to the true minimum, with contours superimposed; (b) points for which the SQP algorithm converged to the true minimum



(a)



(b)

Fig. 6. Final solutions given by the annealing algorithm for 1000 random starts: (a) $T_0 = 1$; $N = 500$; $\rho = 0.9$ (all solutions are within the central contour); (b) $T_0 = 10$; $N = 1000$; $\rho = 0.95$ (all solutions are within the central contour)

problem discussed in Section 2.2 and to equation (3). A second problem with this approach is that, if there are many local minima with function values very similar to the global minimum, then convergence may be very slow.

Brooks and Morgan (1994) suggested an alternative method, in which the annealing component is again stopped prematurely, but instead of taking a single end point to start the second component we take each of the points accepted at the final temperature, together with the best point overall, as starting points. The second component is then run once from each starting point and the best solution generated from these points is given as the hybrid solution. Here the choice of the value of N_t is not so important. As long as N is sufficiently large for the algorithm to have settled down to the equilibrium distribution then, as we have seen in Section 4, we can reasonably expect at least one of these points to be within the required well and thus lead to the second component finding the global minimum. Brooks (1993) discusses this hybrid algorithm in greater detail as well as providing the Fortran code.

5.2. *Univariate normal mixtures*

We shall now provide an example to demonstrate the behaviour of this hybrid algorithm and to compare it with the two traditional algorithms recommended by the NAG and with annealing on its own. The problem is to estimate the parameters $\mu_1, \sigma_1, \mu_2, \sigma_2$ and γ from the density of a mixture of two univariate normals, given by

$$f(x) = \gamma f_1(x) + (1 - \gamma) f_2(x) \tag{4}$$

where $f_i(x)$ is the normal density with mean μ_i and standard deviation σ_i, $i = 1, 2$.

Given a random sample $x_1, x_2, \ldots, x_n$ we can estimate these parameters by minimizing minus the log-likelihood:

$$l(\mu_1, \sigma_1, \mu_2, \sigma_2, \gamma) = \sum_{i=1}^{n} \log f(x_i).$$

The problem of estimating the parameters in a mixture of two normal densities by maximum likelihood is well known to be particulary difficult. See Everitt and Hand (1981), p. 36. These difficulties arise for two reasons. The first is that

$$l(x_i, 0, \mu_2, \sigma_2, \gamma) = l(\mu_1, \sigma_1, x_i, 0, \gamma) = \infty, \qquad i = 1, \ldots, n,$$

so for each sample point there are two singularities to the likelihood surface. The second problem is that if two or more sample values are close together then this causes a local maximum in the likelihood surface. Thus it is commonly found that the likelihood is quite 'bumpy'—see Day (1969). One way to overcome this problem is to assume that $\sigma_1^2 = \sigma_2^2$, but of course this is quite restrictive. Duda and Hart (1973) showed that this restriction can be lifted if we restrict our attention to the largest of the finite maxima of the likelihood surface. Also, simulation studies by Hosmer (1973) showed that, with sensible starting points, iterative maximum likelihood estimators will not generally converge to points associated with singularities. Selecting good starting points is not easy but, as shown below, the annealing component of the hybrid appears to do this very well.

To highlight the problems encountered by traditional algorithms we randomly selected 1000 starting points for both the QN and the SQP algorithms, and recorded the resulting solutions for a selection of data sets. These data sets were generated from the density given in equation (4) as in Everitt (1984). 10 samples, each of 50 observations, were generated from models with three different sets of parameter values given below:

(a) samples 1–10, $\mu_1 = 0.0$, $\sigma_1^2 = 0.5$, $\mu_2 = 3.0$, $\sigma_2^2 = 1.0$, $\gamma = 0.4$;
(b) samples 11–20, $\mu_1 = 0.0$, $\sigma_1^2 = 1.0$, $\mu_2 = 3.0$, $\sigma_2^2 = 2.0$, $\gamma = 0.4$;
(c) samples 21–30, $\mu_1 = 0.0$, $\sigma_1^2 = 1.0$, $\mu_2 = 3.0$, $\sigma_2^2 = 2.0$, $\gamma = 0.2$.

The first two samples have the same means and mixture parameters, but different variances, so that there is greater overlap in the second. The final sample takes the same means and variances as the second sample but decreases the mixing parameter so that the second component becomes more dominant.

The hybrid algorithm was also used to produce a solution for each of the 30 data sets, with the second component being the QN algorithm implemented by the NAG routine E04JAF. For comparison, we also used the annealing algorithm on its own to produce a solution for each data set, so that four methods were used in all.

5.3. *Results*

As we would have expected, the two traditional algorithms repeatedly stick at points associated with singularities. Disregarding those solutions and looking only at solutions corresponding to finite minima of the log-likelihood, we find that these algorithms generally converged to the global minimum for approximately 250 of the 1000 starting points. Between two and 10 of the 100 singularities were given as solutions for some starting points, though rarely did more than two or three starting points converge to the same singularity.

The hybrid algorithm performed extremely well. The annealing parameters were set to the values $N = 2000$, $\rho = 0.90$ and $T_0 = 20$ with the stopping parameter set to $N_t = 15$. With this large value of N, and low value of N_t, about 10 starting points were generated by the first component for each data set, but with as many as 25 for some data sets.

The final solutions were extremely accurate. One very obvious difference between the solutions given by the hybrid algorithm and the traditional methods is that the hybrid found no solutions at points associated with singularities. This is a predictable property of the annealing algorithm, though the probability of finding one of these points increases with the number of new points considered. The traditional algorithms systematically search the parameter space looking for negative gradients, to which they are attracted. Thus if the path of points generated by the traditional iterative methods passes close to a singularity then it will be attracted by the small but infinitely deep well surrounding it and will thus converge on that point. The annealing algorithm is attracted to a singularity only if a randomly selected point falls deep within such a well, where the algorithm cannot reasonably be expected to jump back out again. These wells are very steep but have only a small radius and thus they occupy only a tiny fraction of the entire parameter space. Hence the likelihood that the annealing algorithm becomes stuck in such a well is very low, and the behaviour of the annealing algorithm is not adversely affected by the singularities. Thus the first component can produce sensible starting points, some of which are sufficiently close to the global minimum to allow the second component to converge on the global minimum.

The annealing algorithm on its own, with parameters T_0, ρ and N set to values 10, 0.90 and 1000 respectively, also fared well but it was very slow to converge. The annealing algorithm, when allowed to run 10 times as long as the hybrid, still could not provide acceptably accurate solutions.

Thus the hybrid algorithm appears to work very well. Traditional routines on their own need good starting points which the annealing component of the hybrid can provide. So, whereas both the annealing and the traditional algorithms encounter considerable difficulties when used alone, these may be overcome by a combination of the two.

5.4. *Assessing number of components in a mixture*

McLachlan (1987) discussed a simulation experiment for assessing the null distribution of $-2 \log \lambda$ for the test $H_0: \gamma = 0$ *versus* $H_1: \gamma \neq 0$ with univariate normal densities, where λ is the *likelihood ratio test statistic.* 500 samples of size 100 were generated under H_0 and the values of $-2 \log \lambda$ were recorded for each sample. These values were then used to estimate the null distribution.

Given the occurrence of local maxima in the likelihood for mixture models, many optimization methods offer no guarantee that the largest of the local maxima will be located for any observed sample. Hence the values of $-2 \log \lambda$, produced in such a simulation, may be biased downwards if the global maximum cannot be reliably found. McLachlan (1987) suggested that this bias be limited by undertaking a systematic search for all local maxima and produced a simulated null distribution between those of χ^2_4 and χ^2_6. The simulated $-2 \log \lambda$ values that McLachlan produced had mean 5.96 and variance 13.86 suggesting that, though the variance was a little large, the simulated null distribution was approximately χ^2_6.

To investigate the extent of the downward bias, we used the hybrid algorithm to calculate $-2 \log \lambda$ for each of 250 simulated samples of size 100. To ensure that accurate and reliable results were obtained, we took the very high value of the annealing parameter $N = 4000$, so that a large number of starting points would be generated for each optimization. Fig. 7 shows a histogram indicating the numbers of starting points generated for the five-parameter optimization, i.e. under H_1. The other annealing parameters were set to the values $T_0 = 10$, $\rho = 0.90$ and $N_t = 10$.

The $-2 \log \lambda$ values produced in this way had mean value 8.50 and variance 39.49. This differs considerably from the results obtained by McLachlan (1987) and contradicts the suggestion that $-2 \log \lambda$ has a χ^2-distribution under H_0. Fig. 8 plots the simulated null distribution together with the χ^2-distribution with degrees of freedom 6, 8 and 10. This plot confirms the fact that $-2 \log \lambda$ does not have a χ^2-distribution under H_0, as suggested by McLachlan.

The very nature of this problem means that any incorrect solutions will underestimate the true value of $-2 \log \lambda$. The fact that the hybrid algorithm obtains larger values than those produced by McLachlan (1987) provides further evidence that the hybrid algorithm outperforms traditional techniques. Further discussion of this mixture problem is provided by Feng and McCulloch (1995).

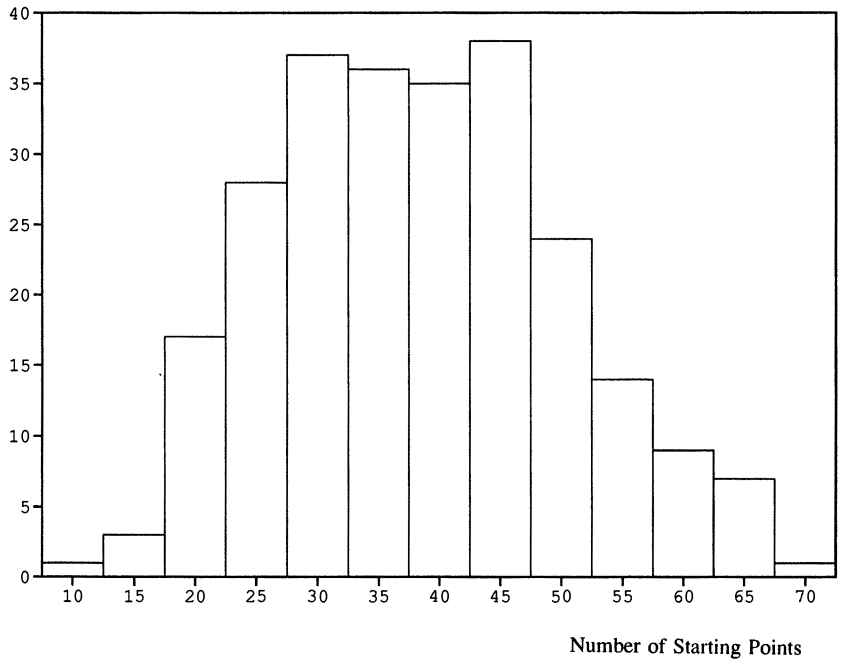


Fig. 7. Histogram showing the number of starting points generated by the annealing component in the hybrid algorithm with the parameter values $N = 4000$, $T_0 = 10$, $\rho = 0.90$ and $N_t = 10$

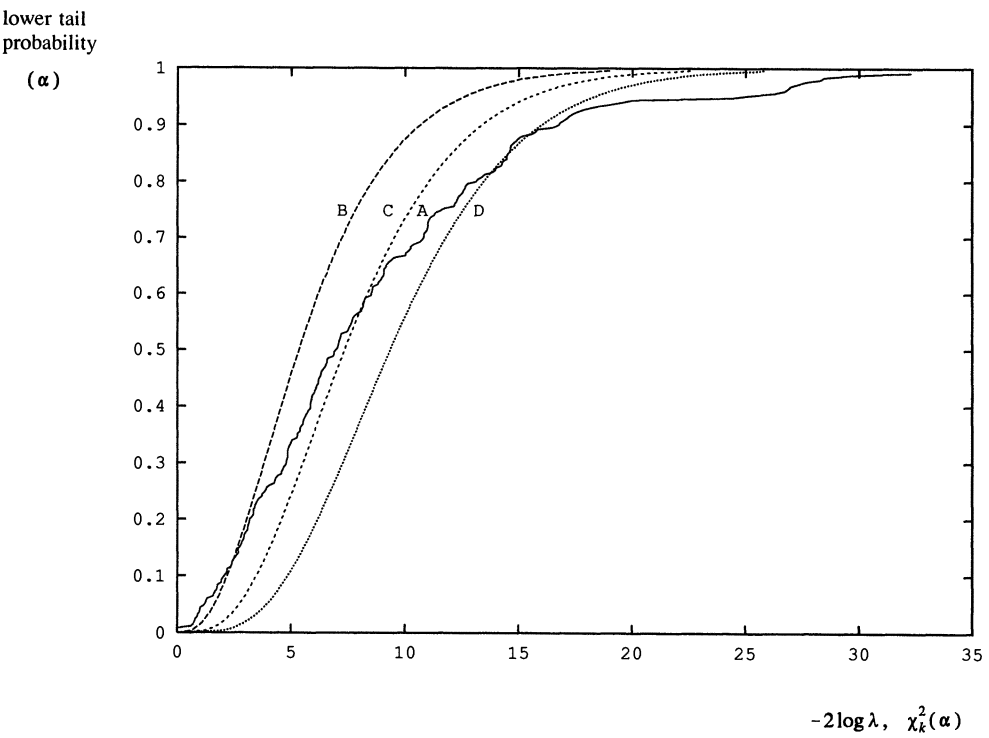


Fig. 8. Plots of distribution functions: A, simulated null distribution of $-2\log\lambda$; B, χ^2_6; C, χ^2_8; D, χ^2_{10}

Acknowledgements

We gratefully acknowledge the helpful comments on an earlier draft of this paper of a referee, A. C. Atkinson, S. T. Buckland, Z. D. Feng, W. J. Krzanowski, G. J. McLachlan and I. H. Osman.

References

Aarts, E. H. L. and Korst, J. H. M. (1988) *Simulated Annealing and Boltzmann Machines.* Chichester: Wiley.

Atkinson, A. C. (1992) A segmented algorithm for simulated annealing. *Statist. Comput.*, **2**, 203–212.

Barnett, V. D. (1966) Evaluation of the maximum-likelihood estimator where the likelihood has multiple roots. *Biometrika*, **53**, 151–165.

Bertsimas, D. and Tsitsiklis, J. N. (1993) Simulated annealing. *Statist. Sci.*, **8**, 10–15.

Bohachevsky, I., Johnson, M. E. and Stein, M. L. (1986) Generalized simulated annealing for function optimization. *Technometrics*, **28**, 209–217.

Boltzmann, L. (1877) *Sitzber. Akad. Wiss, Wien*, **76**, 373.

Brooks, S. P. (1993) A hybrid optimization algorithm. Submitted to *Appl. Statist.*

Brooks, S. P. and Morgan, B. J. T. (1994) Automatic starting point selection for function optimisation. *Statist. Comput.*, **4**, 173–177.

Bunday, B. D. (1984) *Basic Optimisation Methods.* London: Arnold.

Corana, A., Marchesi, C. and Ridella, S. (1987) Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Trans. Math. Softwr.*, **13**, 262–280.

Day, N. E. (1969) Estimating the components of a mixture of normal distributions. *Biometrika*, **56**, 463–474.

Drago, G., Manella, A., Nervi, M., Repetto, M. and Secondo, G. (1992) A combined strategy for optimization in non linear magnetic problems using simulated annealing and search techniques. *IEEE Trans. Magn.*, **28**, 1541–1544.

Duda, R. and Hart, P. (1973) *Pattern Classification and Scene Analysis.* New York: Wiley.

Everitt, B. S. (1984) Maximum likelihood estimation of the parameters in a mixture of two univariate normal distributions; a comparison of different algorithms. *Statistician*, **33**, 205–215.

Everitt, B. S. and Hand, D. J. (1981) *Finite Mixture Distributions.* London: Chapman and Hall.
Feng, Z. D. and McCulloch, C. E. (1995) On the likelihood ratio test statistic for the number of components in a normal mixture with unequal variance. *Biometrics*, to be published.
Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattn Anal. Mach. Intell.*, **6**, 721–741.
Gill, P. E., Murray, W. and Wright, M. H. (1981) *Practical Optimization.* London: Academic Press.
Hosmer, D. W. (1973) A comparison of iterative maximum likelihood estimates of the parameters of a mixture of two normal distributions under three different types of sample. *Biometrics*, **29**, 761–770.
Ingber, L. (1992) Genetic algorithms and very fast simulated reannealing: a comparison. *Math. Comput. Modlling*, **16**, 87–100.
———(1994) Simulated annealing: practice *versus* theory. *Math. Comput. Modlling*, **18**, 29–57.
Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.
Lundy, M. and Mees, A. (1986) Convergence of an annealing algorithm. *Math. Progrmmng*, **34**, 111–124.
McLachlan, G. J. (1987) On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Appl. Statist.*, **36**, 318–324.
Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, A. (1953) Equation of state calculations by fast computing machines. *J. Chem. Phys.*, **21**, 1087–1092.
Mitra, D., Romeo, F. and Sangiovanni-Vincentelli, A. L. (1986) Convergence and finite-time behaviour of simulated annealing. *Adv. Appl. Probab.*, **18**, 747–771.
Numerical Algorithms Group (1991) *Numerical Algorithms Group Library Manual.* Oxford: Numerical Algorithms Group.
Osman, I. H. and Christofides, N. (1994) Capacitated clustering problems by hybrid simulated annealing and tabu search. *Int. Trans. Oper. Res.*, **1**, 317–336.
Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. (1989) *Numerical Recipes.* Cambridge: Cambridge University Press.
Reeves, C. R. (1993) *Modern Heuristic Techniques.* Oxford: Blackwell Scientific.
Simkin, J. and Trowbridge, C. W. (1992) Optimizing electromagnetic devices combining direct search methods with simulated annealing. *IEEE Trans. Magn.*, **28**, 1545–1548.
Szu, H. and Hartley, R. (1987) Fast simulated annealing. *Phys. Lett.* A, **122**, 157–162.