Pergamon

Computers Math. Applic. Vol. 33, No. 1/2, pp. 81–94, 1997

0898-1221/97 $17.00 + 0.00

PII: S0898-1221(96)00221-0

Simulated Annealing Simulated

V. FABIAN
Department of Statistics and Probability
Michigan State University
Wells Hall, East Lansing, MI 48824, U.S.A.

Abstract—The performance of simulated annealing methods for finding a global minimum point of a function is studied.

Keywords—Simulated annealing, Simulation, Stochastic approximation, Random search, Deterministic search, Gradient method, Slow approach to the limiting distribution, Doubtful performance of simulating annealing.

1. INTRODUCTION

Simulated annealing methods are methods proposed for the problem of finding, numerically, a point of the global minimum of a function defined on a subset of a k-dimensional Euclidean space. The motivation of the methods lies in the physical process of annealing, in which a solid is heated to a liquid state and, when cooled sufficiently slowly, takes up the configuration with minimal inner energy. Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller [1] described this process mathematically. Simulating annealing uses this mathematical description for the minimization of other functions than the energy. The first results have been published by Černý [2,3], Kirpatrick, Gelatt Jr., and Vecchi [4,5], and Geman and Geman [6]. For a related earlier result, see Hasminskij [7]. Most of the early considerations concern minimization of functions defined on a finite set. Kushner [8] and Gelfand and Mitter [9] obtained results for functions with infinite domains. Laarhoven and Aarts [10], Laarhoven [11] and Pflug [12] are monographs on simulated annealing. Steel [13], in a review of [11], calls simulated annealing *the most exciting algorithmic development of the decade.*

Some of the literature, e.g., [11], reports surprisingly good results when applying simulated annealing to difficult problems. The description of the results is often short on detail; e.g., in [11], not the number of steps in the method, but computer time is reported and the behavior of the function minimized, the position of local minima, etc., is difficult to ascertain.

Analogies of simulated annealing with stochastic approximation made us suspicious of the performance of the former. This has been the motivation of the present study in which simulated annealing methods are tested on very simple functions.

Our results confirm the suspicion and contrast frequent reports in literature of surprisingly good performance. On the other hand, our results do no contradict the theoretical results, which are mostly asymptotic, and more modest than informal claims. Pflug [14], in a talk, argued that simulated annealing cannot work satisfactorily in the case of a very simple function (similar to that which we consider in Section 2), but we were unable to understand the proof, unable to get a written proof, and it seems the result was not published. Laarhoven [11, p. 33], recommends a cooling schedule (a rule for letting a parameter of the method slowly approach 0), but warns that the schedule *precludes any guarantee for the proximity of the final configuration to a globally*

optimal one. Another theoretical result that constitutes a warning about the performance of simulated annealing methods appears in the last quoted monograph and is discussed in Section 2.

Section 2 is concerned with the simulated annealing method, applied as originally proposed, to a function on a finite domain. The method is described by a Markov chain $x = \langle x_n \rangle$, where x_n is the estimate of the point of global minimum after n steps (i.e., after n function evaluations). We consider a class of very simple functions with the domain a set of size $2k + 1$ and with a local minimum 0 at $-k$ and a global minimum -1 at $k + 1$ (cf. Figure 1 below). We consider $k = 4, 6, 8, 10$ and find the convergence of the distribution of x_n to the limiting distribution very slow. We have chosen a parameter in the simulated annealing method such that the limiting distribution gives probability 0.8 to the point of global minimum and considered the initial distribution concentrated at the point $-k$. (Choosing a higher probability than 0.8 would make the convergence slower still.) For $k = 10$ and $n = 10^8$, the probability of $\{x_n = k + 1\}$ is 0.00502 (cf. Table 1). *For the same k, just 22 function evaluations suffice to determine the point $k + 1$ of global minimum exactly.*

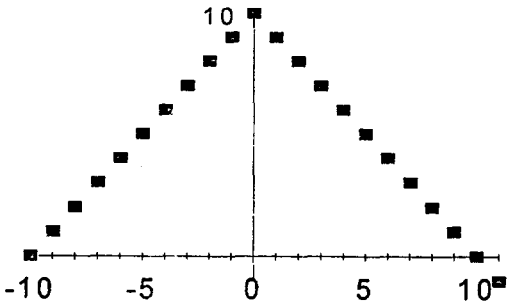

Figure 1.

Table 1. The speed with which $p_{k+1}^{(n)}$ approaches the limiting value 0.8.

k	4	6	8	10	10
initial density	$p_{-k}^{(0)} = 1$	$p_{-k}^{(0)} = 1$	$p_{-k}^{(0)} = 1$	$p_{-k}^{(0)} = 1$	uniform
$p_{k+1}^{(n)}$	0.7	0.7	0.31954	0.00502	0.466751
n	62,134	5,035,077	10^8	10^8	10^8

Starting with Section 3, minimization of functions defined on an infinite set is considered, again for rather very simple functions (see Figure 2 in Section 3.1). Each of the two particular methods considered depends on two parameters a and σ and even with "best" (i.e., best we found) values for these parameters, the performance is rather bad. In the case described in Table 4, Section 3.2, the expected squared deviation from the point of global minimum is approximately 2.24 after $n = 10,000$ steps. With this many function evaluations, a primitive deterministic search would estimate the point of global minimum with a squared deviation 10^{-8}. Other cases are similar and worse. In addition, the performance of the methods depends strongly on the choice of the two parameters a and σ and the optimal values for these depend, in turn, on the function to be minimized.

In Section 3.5, we consider a simple improvement $z = \langle z_n \rangle$ over the simulating annealing method x. The improvement is considerable with properly chosen a and σ. However, it was not clear whether the improvement persists in the multidimensional case. In Section 4, we study the behavior for a very simple function F defined on $[-3, 3]^k$ and, indeed, the advantage of z over x diminishes with increasing dimension. We also compare here the behavior of x and z with random search, discrete search, and discrete search followed up by a gradient method. The discrete search followed up by the gradient method performs much better than any of the other

methods for the specific function considered. The last property may change if other functions are considered, because then the deterministic search may fail to get close to the point of global minimum. However, easy description of the performance of the deterministic search for functions in the class of all Lipschitz (C) continuous functions is available; no such general description of the behavior of the simulated annealing estimator x_n for a given n is available.

We think that our examples make the usefulness of simulated annealing doubtful.

2. FINITE DOMAIN

Consider a positive integer k and the function f defined on the set $D = \{-k, -k+1, \ldots, k+1\}$ by the relation $f(x) = k - |x|$ (see Figure 1 for the $k = 10$ case). The function has a local minimum at $-k$, with the function value 0, and a global minimum at $k+1$, with the value -1. It takes $2(k+1)$ function evaluations to find, with certainty, the point $k+1$ of global minimum.

Consider the homogeneous simulated annealing method. This method needs a definition of neighborhoods (not topological neighborhoods) for points in the function domain D. Define these neighborhoods by

$$N(-k) = \{-k, -k+1\}, \qquad N(k+1) = \{k, k+1\}, \quad \text{and}$$
$$N(x) = \{x-1, x+1\}, \qquad \text{for } -k < x \leq k.$$

In addition, the method depends on a positive constant c. The method changes x_{n-1} to a point x_n, the approximation at time n, as follows. First, a point y is chosen at random in $N(x_{n-1})$. If $\Delta = f(y) - f(x) \leq 0$, then x_n is set equal to y; if $\Delta > 0$, then, conditionally on the past, x_n is set equal to y with probability $\alpha = e^{-\Delta/c}$, and to x_{n-1} with probability $1 - \alpha$ (see [11, Section 2.2]). This simplifies in our case, because Δ equals 1 if it is positive. The limiting distribution of x is known and described in the literature under much more general assumptions than here (e.g., [11, p. 20, (2.43)]), but it may be easier to derive it in our special case anew. The transition probabilities for the homogeneous Markov chain x are then as follows, with $\beta = 0.5 - \alpha/2$:

$$p_{-k,-k} = 1 - \frac{\alpha}{2}, \quad p_{-k,-k+1} = \frac{\alpha}{2}, \tag{2.1}$$

$$p_{i,i-1} = 0.5, \quad p_{i,i} = \beta, p_{i,i+1} = \frac{\alpha}{2}, \qquad \text{for } -k < i < 0, \tag{2.2i}$$

$$p_{0,-1} = \frac{1}{2}, \quad p_{0,1} = \frac{1}{2}, \tag{2.3}$$

$$p_{i,i-1} = \frac{\alpha}{2}, \quad p_{i,i} = \beta, \quad p_{i,i+1} = \frac{1}{2}, \qquad \text{for } 0 < i \leq k, \tag{2.4i}$$

$$p_{k+1,k} = \frac{\alpha}{2}, \quad p_{k+1,k+1} = 1 - \frac{\alpha}{2}. \tag{2.5}$$

Arranging these into a matrix and reading the columns easily gives relations between the discrete density (abbreviated henceforth to *density*) q at time n and $\tilde{q}$ at time $n+1$, if, on the left-hand side, q is replaced $\tilde{q}$. As stated, the equations determine the stationary density q, the limit of the discrete densities of x_n for any initial density.

$$q_{-k} = \left(1 - \frac{\alpha}{2}\right) q_{-k} + 0.5 q_{-k+1}, \tag{2.6}$$

$$q_i = \frac{\alpha}{2} q_{i-1} + \beta q_i + 0.5 q_{i+1}, \qquad \text{for } -k < i < 0, \tag{2.7i}$$

$$q_0 = \frac{\alpha}{2}(q_{-1} + q_1), \tag{2.8}$$

$$q_i = 0.5 q_{i-1} + \beta q_i + \frac{\alpha}{2} q_{i+1}, \qquad \text{for } 0 < i \leq k, \tag{2.9i}$$

$$q_{k+1} = 0.5 q_k + \left(1 - \frac{\alpha}{2}\right) q_{k+1}. \tag{2.10}$$

Equation (2.6) is equivalent to $q_{-k+1} = \alpha q_{-k}$ and assuming $\alpha q_{i-1} = q_i$, we obtain, from (2.7i), that also $\alpha q_i = q_{i+1}$. Thus, by induction, the last relation holds for all $i = -k, \dots, -1$. Consequently, $q_i = \alpha^{k+i} q_{-k}$ for $i = -k, \dots, 0$. By symmetry, $q_i = \alpha^{k+1-i} q_{k+1}$ for $i = 0, \dots, k+1$. For $i = 0$, the two relations imply that $q_{-k} = \alpha q_{k+1}$. These relations give

$$q_i = \alpha^{k+1-|i|} q_{k+1}, \qquad \text{for } i = -k, \dots, k+1, \tag{2.11}$$

and it is easy to verify that (2.8) also holds. The sum of all q_i is $((1+\alpha)(1-\alpha^{k+1}))/(1-\alpha)$ q_{k+1} and

$$q_{k+1} = \frac{1-\alpha}{(1+\alpha)(1-\alpha^{k+1})}. \tag{2.12}$$

For small α and k not very small, this is close to $(1-\alpha)/(1+\alpha)$ and close to β if $\alpha = (1-\beta)/(1+\beta)$.

Consider the choice $\beta = 0.8$ and the corresponding $\alpha = 0.2/1.8$ (for $k = 4, 6, 8, 10$, this results in a β that differs from 0.8 by, respectively, 1.4×10^{-5}, 1.7×10^{-7}, 2.1×10^{-9}, 2.5×10^{-11}). Thus, the stationary and limiting density p has $p_{k+1} = 0.8$ (with the small error indicated above, inconsequential in the considerations below). How fast does the density approach its limit? For several k, we consider the value of $p_{k+1}^{(n)} = P\{x_n = k+1\}$ and find n for which $p_{k+1}^{(n)} \geq 0.7$, if such an n is at most 10^8; if such an n is larger than 10^8, we determine $p_{k+1}^{(n)}$ for $n = 10^8$. We have chosen the initial density concentrated on $-k$, except in one case we have chosen the initial density to be uniform over $\{-k, \dots, k+1\}$. The results are in Table 1. Table 2 gives the density for case $k = 10$ and $n = 10^8$ for the two initial discrete densities.

Table 2. Values of $p_i^{(n)}$ for $n = 10^8$, $k = 10$, and two initial densities.

Initial density: $p_{-k}^{(0)} = 1$				Initial density: uniform			
i	$p_i^{(n)}$	i	$p_i^{(n)}$	i	$p_i^{(n)}$	i	$p_i^{(n)}$
-10	0.88387	1	1.28×10^{-10}	-10	0.42214	1	1.87×10^{-10}
-9	0.09821	2	1.40×10^{-10}	-9	0.04690	2	1.26×10^{-9}
-8	0.01091	3	2.43×10^{-10}	-8	0.00521	3	1.09×10^{-8}
-7	0.00121	4	1.18×10^{-9}	-7	0.00058	4	9.76×10^{-8}
-6	0.00013	5	9.57×10^{-9}	-6	0.00006	5	8.78×10^{-7}
-5	1.50×10^{-5}	6	8.51×10^{-8}	-5	7.15×10^{-6}	6	7.90×10^{-6}
-4	1.66×10^{-6}	7	7.65×10^{-7}	-4	7.94×10^{-7}	7	0.000071
-3	1.85×10^{-7}	8	6.89×10^{-6}	-3	8.82×10^{-8}	8	0.000640
-2	2.04×10^{-8}	9	0.00006	-2	9.75×10^{-9}	9	0.005762
-1	2.15×10^{-9}	10	0.00056	-1	1.04×10^{-9}	10	0.051861
0	1.27×10^{-10}	11	0.00502	0	6.80×10^{-11}	11	0.466751

We see that, for the function considered and the initial density such that $p_{-k}^{(0)} = 1$, the approach of p_n to the limiting density is very slow. That is, for our choice of α that leads to the limiting density with the very modest property, that it gives probability 0.8 (not, e.g., 0.99) to the point $k+1$ of the global minimum. The results are consistent with a heuristic argument that for α such that the chain has a moderately large probability to stay at 11, it has a small probability to cross from -10 to 0. The results also complete some theoretical bounds on the speed of such convergence. The theoretical results (see [11, Section 2.4, relation (2.64)] in particular) are that, for the density p_n to be within a positive ε from the limiting density, it is enough that n be at most a constant times N^2 where N is the size of the function domain.

Our results show the following.

(i) An extremely slow approach of the density p_n to its limit.

(ii) An extremely poor performance of the simulated annealing method when compared to a deterministic search that requires $2k + 1$ function evaluations and locates the point of global minimum exactly.

For the homogeneous case, in general, we have another property.

(iii) If, for some n, $x_n = u$ is already close to a point of global minimum, the process $x_{n+1}, x_{n+2}, \ldots$, with the initial density giving probability 1 to u, will still wander through all the points of the domain, including all the points of local minima and local maxima. This is because it has the same limiting density as the original process.

These results show difficulties for the nonhomogeneous process, in which α is slowly decreasing with n. In the consideration of this nonhomogeneous process (see [11, Section 3.2]), the attempt is made to decrease α when the density of x_n is close to the limiting density; it is not known when this is the case, and we have shown it may take an extremely large number of steps.

The advice given on when to stop the algorithm is often (e.g., [11, Section 3.2, p. 31]) to stop when the function values do not change much anymore. However, that may well happen at a point of local minimum, and, from (iii), if the function values do not change too much for a while, they will start changing eventually. A heuristic argument why (iii) persists in the nonhomogeneous case is that the process started from x_n at time n still will *find* a global minimum, and not only for the particular function considered, and so it must go through the whole domain again.

There are reports on applications of simulated annealing methods where these methods quickly found a point with the function value nearly minimal (e.g., [11, Chapter 4]). This may be possible only if the methods do not have to go through several steps uphill (i.e., with increasing function values), which was the case for our f here. Thus, successful applications may be perhaps obtained when the function values at points of local minima are close to the minimal function value, or when it takes only a few steps uphill to escape from a local minimum.

It should be also noted that the neighborhoods are to be chosen and are not an inherent part of function itself. The behavior of the simulated annealing methods depends strongly, however, on the choice of the neighborhoods. It is quite obvious that our results would be different, if the neighborhood of $-k$ was chosen to include the point $k + 1$.

Lundy and Meese [15] construct functions f_N on size $(2N/\delta + 1)^2$ domain, with δ a small positive number, and claim that the time T_a to reach the point 0 of the minimum has expectation $ET_a = O(N)$ for the simulated annealing method, while for a competitor (gradient method with the initial point randomized), the analogous time T_c satisfies $ET_c = 4N^2/\delta$. The function has many local minima, but from each, one step slightly uphill gets to a long sequence of downhill steps. The competitor is fine tuned to its disadvantage such that it can reach 0 only if the initial point falls close to 0, in a set of size $(2/\delta + 1)^2$. However, even with such choices the claim is questionable, because of an unjustified step in the proof. Functions f_N are chosen in such a way that the probability the annealing method makes an uphill step from a point of local minimum towards 0 is considerably larger than that (say p) of a step in a direction away from 0. The authors declare the latter negligible and give the proof for the case that p is indeed 0. But this way they no longer study a simulated annealing method, but a different method with a different limiting density. The main difficulty with the omission of the small probability is that as N increases to infinity, there is a very large number of events, all with probability p, the occurrence of any of which the proof neglects to consider.

3. INFINITE DOMAIN

3.1. Two Methods

With this section, we begin considerations of two simulated annealing methods for functions defined on an infinite domain, a subset of the k-dimensional space R^k. The methods do not require a choice of neighborhoods, but the theoretical results do not describe the behavior in

such detail as for the homogeneous method in the finite domain case. The methods need a choice of two parameters a and σ.

Considered here are two methods K and GM, proposed and studied in [8] and [9]. We study the behavior of the two methods when these are applied to about the simplest two functions, the function in Figure 2, restricted to the domain $[-3, 3]$, and the function not restricted. It might be easier for the reader if we refer to these two functions by the domain rather then by two names. The function f is defined in Table 3 and its graph is shown in Figure 2.

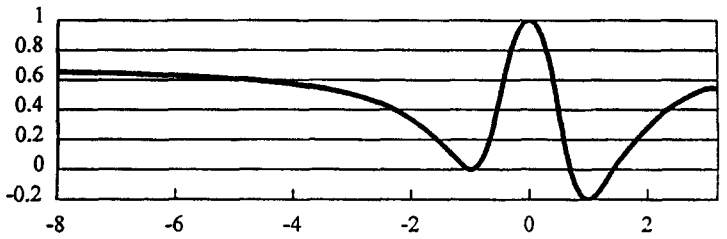

Figure 2. Function f.

Table 3. Function f.

On	$f(x)$
$[-8, -1.3)$	$0.099 - 0.39 \arctan(1.3 + x)$
$[-1.3, -1)$	$(1 + x)^2(5 + 3x)$
$[-1, -0.5)$	$2(1 + x)^2$
$[-0.5, 0)$	$1 - 2x^2$
$[0, 0.5)$	$1 - x^2(2 + 0.8x)$
$[0.5, 1)$	$-0.2 + 2(x - 1)^2(1.4 - 0.4x)$
$[1, 1.5)$	$-0.2 + 2(x - 1)^2(2 - x)$
$[1.5, 3]$	$0.05 + 0.5 \arctan(x - 1.5)$

The two methods require the existence of the second derivative of the function. The function f has a second derivative, bounded by 6.4, everywhere except at points $-1.3, -0.5, 0.5, 1.5$, where one-sided second derivatives exist (cf. Table 3). f can be approximated with any desired accuracy by a function that has a second derivative everywhere, bounded by the same constant as the original function. That is, change f'' by interpolating linearly between $f''(-1.3-\varepsilon)$ and $f''_+(-1.3)$ on $(-1.3 - \varepsilon, -1.3]$ and change f' and f accordingly to preserve continuity; this changes f on $(-1.3-\varepsilon, 3)$ by at most $36\varepsilon + 1.1\varepsilon^2$ (even if other points than -1.3 are considered). The decision not to use such a modification of f was taken to save time on computations, some of them very long.

The two methods, for finding a point of global minimum of a function defined on the k-dimensional Euclidean space R^k, are described by the same recursion formula

$$x_{n+1} = x_n - a_n \Delta(x_n) + c_n \xi_n, \tag{3.1}$$

where $\Delta(x)$ is the gradient of f at $x, \xi_1, \xi_2, \ldots$ are independent standard normal random variables and

$$a_n = \frac{a}{\log(n+2)}, \quad c_n = \frac{\sigma}{\log(n+2)} \qquad \text{for method K,} \tag{3.2}$$

$$a_n = \frac{a}{n}, \quad c_n = \frac{\sigma}{\sqrt{n \log\log(n+15)}} \qquad \text{for method GM,} \tag{3.3}$$

with positive numbers a and σ. These are specific cases of the methods originally proposed, the numbers 2 and 15 are specific choices to cause the fractions to make sense. (GM method has zero

instead of 15, but requires (3.3) for large n only; we think the slight change is inconsequential). Kushner [8] also considers the situation when the gradient cannot be computed, but is estimated by biased random variables (similar to stochastic approximation methods).

The two methods are stochastic approximation methods except that the steps a_n for method K and the variances of $c_n\xi_n$ for both the methods are larger than the optimal choices established for stochastic approximation methods. However, stochastic approximation methods do not guarantee convergence to a point of global minimum, except for a new modification in [16]. The modification is difficult to use unless the dimension of the domain is small.

We study the behavior of the two methods first by $E(x_n - 1)^2$, the expected squared deviation of x_n, for $n = 10,000$, from the point 1 of the global minimum, and its dependence on a, σ and the domain of the function. We then limit our considerations to method K and domain $[-3, 3]$ only, and obtain results on the density of x_n for several pairs $\langle a, \sigma \rangle$. We show a few paths of $x = \langle x_1, \ldots, x_n \rangle$. The results suggest a possible improvement z over x. We study this improvement for the same function and then for a multidimensional problem. The results show that the two methods K and GM are extremely slow, and that the improvement z is losing the advantage over x as the dimension increases.

3.2. The Expected Squared Error

We study the expected squared error $E(x_n - 1)^2$ for both K and GM methods, with the initial point $x_0 = -1$ of a local minimum of f, and with $n = 10,000$. Both methods depend on two constants a and σ.

For given a and σ, we estimate $E(x_n - 1)^2$ from 5,000 independent replicas Z_{is} of the random variable $Z = (x_n - 1)^2$ with $i = 1, \ldots, 100$ and $s = 1, \ldots, 50$. For ease of implementation, observations of $Z_i = (1/100)(Z_{i1} + \cdots Z_{i100})$ are obtained. The random variables Z_i can be expected to be normal to a high level of accuracy. We then estimate $E(x_n - 1)^2$ by the usual 0.99-interval estimate based on the normality assumption. The center of this interval estimate is $\bar{Z} = (1/50)(Z_1 + \cdots + Z_{50})$. The observation of $\bar{Z}$ will be called the mean-squared deviation and abbreviated to msd.

Table 4. msd for K, domain $[-3, 3]$, maximum half-width of the 0.99 interval estimates is 0.25, "best" $a = 0.3$, $\sigma = 1$.

	a					
σ	0.03	0.1	0.3	1	3	10
0.1	4.04	4.01	4.00	4.00	4.00	5.24
0.3	4.68	4.15	4.03	4.01	3.92	5.21
1	4.04	3.56	**2.24**	3.27	3.03	5.61
3	4.30	4.36	4.13	3.74	3.10	5.32
10	5.36	5.02	5.10	5.05	4.96	5.66
30	6.57	6.62	6.78	6.82	6.81	6.87

Table 5. msd for K, domain $[-8, 3]$, maximum half-width of the 0.99-interval estimates is 0.25, "best" $a = 3$, $\sigma = 1$.

	a					
σ	0.03	0.1	0.3	1	3	10
0.1	4.03	4.01	4.00	4.00	4.00	5.20
0.3	7.92	4.17	4.03	4.01	3.91	5.23
1	20.36	15.89	5.87	3.26	**3.09**	5.60
3	23.03	22.53	21.35	15.37	7.03	6.70
10	24.15	24.44	23.81	23.98	23.81	23.67
30	27.83	28.43	27.70	27.62	28.63	29.44

Table 6. msd for GM, domain $[-3,3]$, maximum half-width of the 0.99-interval estimates is 0.24, "best" $a = 1000$, $\sigma = 30$.

σ	a					
	30	100	300	1,000	3,000	10,000
1	2.81	3.50	3.95	4.00	4.00	4.76
3	2.62	3.11	3.71	3.99	4.00	4.84
10	3.63	2.30	2.72	3.39	3.97	5.03
30	4.13	4.06	3.73	2.41	2.77	5.33
100	4.63	4.60	4.74	4.70	4.59	5.19
300	5.96	5.93	5.96	5.88	6.04	6.00

Table 7. Gelfand-Mitter method, domain $[-8,3]$, maximum half-width of the 0.99-interval estimates is at most 10% of msd, "best" $a = 3,000$, $\sigma = 30$.

σ	a					
	30	100	300	1,000	3,000	10,000
1	9.95	3.89	3.99	4.00	4.00	4.72
3	6.16	4.00	3.92	4.00	4.00	4.80
10	18.11	7.46	2.99	3.47	3.98	5.00
30	22.57	20.68	17.21	6.27	2.89	5.37
100	23.76	23.47	22.93	21.80	20.01	18.31
300	26.43	25.27	26.01	25.22	25.49	26.37

Tables 4–7 give these estimates for the two methods, two functions, and several pairs of a and σ. The estimates have been computed for more values of a and σ; the tables give only values that surround the "best" a and σ.

Note that the expected squared deviations depend on the domain of the function minimized. Thus, for method K, the "best" $a = 0.3$, $\sigma = 1$ for domain $[-3,3]$ are rather a bad choice for domain $[-8,3]$ (cf. Table 5). For the "best" values of a and σ, Table 8 gives estimates of $E(x_n - 1)^2$ obtained anew from 20,000 independent observations in each of the cases and half-widths of the 0.99-intervals estimates with centers msd.

Table 8. msd and half-width of the 0.99-IE for the "best" a and σ.

method	domain	a	σ	msd	Half-width of the 0.99 IE
K	$[-3,3]$	0.3	1	2.23	0.05
K	$[-8,3]$	3	1	3.11	0.03
GM	$[-3,3]$	1000	30	2.40	0.06
GM	$[-8,3]$	3000	30	2.89	0.05

3.3. Density of x_n for $n = 10^m$ and $m = 1,2,3,4$

Estimates of these densities are relative frequencies in 5,000 independent observations. They are reported in graphs only. We report on method K only (similar results have been obtained for method GM), and on domain $[-3,3]$. We show the results for the "best" $a = 0.3$ and $\sigma = 1$, and then for four additional pairs in which the two parameters are changed by multiplication and division by 10. For σ small, mass stays close to -1; for σ large, close to the endpoints of the domain. The msd's from Table 4 are reported again in the description of Figures 3–7.

3.4. Paths

It may be of interest to see how the individual paths behave; this is not easy to show, and we show here 6 replicas of the process $\langle x_n \rangle$ evaluated at several n. This concerns method K, domain $[-3,3]$. In Figures 8 and 9, the unit on the horizontal axis is 1,000.

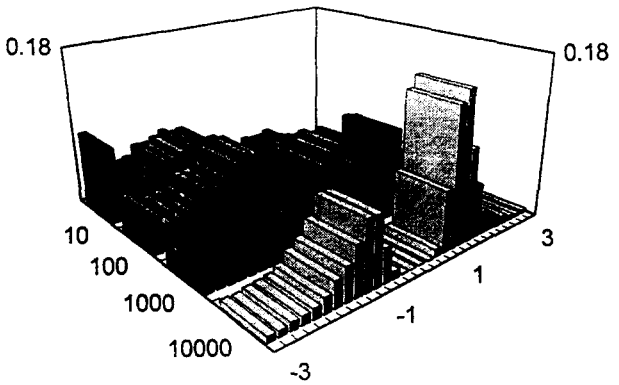

Figure 3. "Best" $a = 0.3$, $\sigma = 1$, msd= 2.24.

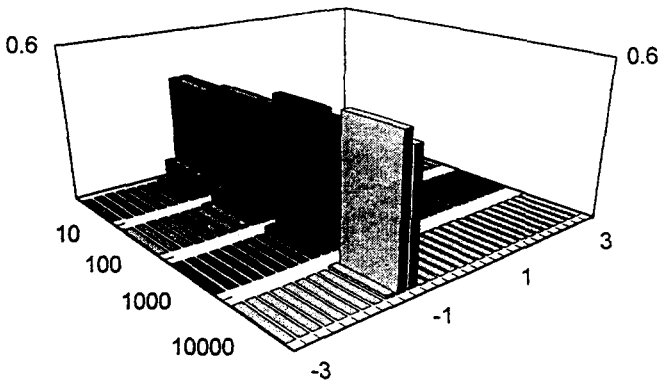

Figure 4. Small $a = 0.03$, small $\sigma = 0.1$, msd= 4.04 ("best" 2.24).

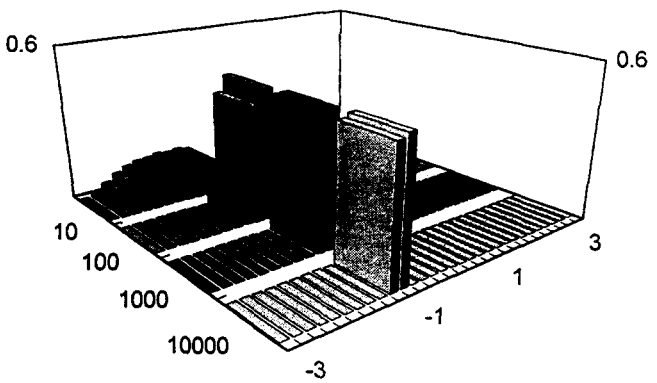

Figure 5. Large $a = 3$, small $\sigma = 0.1$, msd= 4.00 ("best" 2.24).

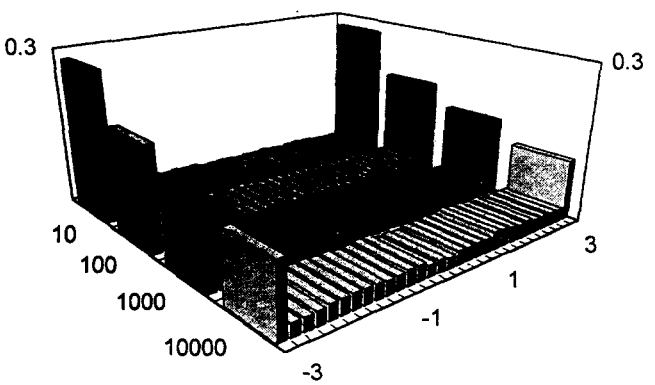

Figure 6. Large $a = 3$, large $\sigma = 10$, msd= 4.96 ("best" 2.24).

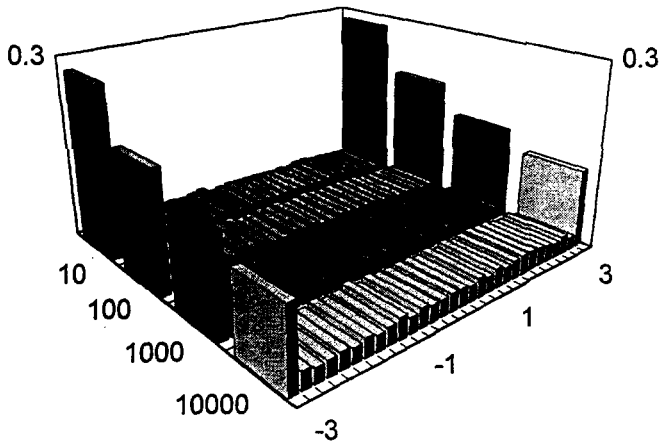

Figure 7. Small $a = 0.03$, large $\sigma = 10$, msd= 5.36 ("best" 2.24).

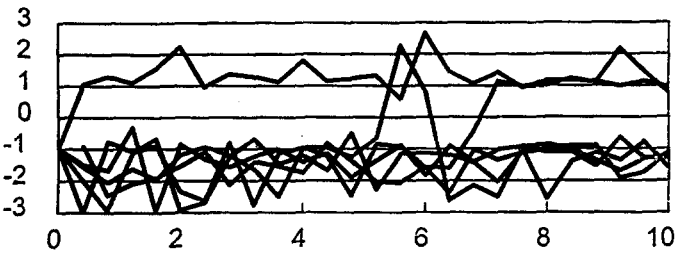

Figure 8. Six independent realizations of $\langle x_1, \ldots, x_n \rangle$ with $n = 10,000$.

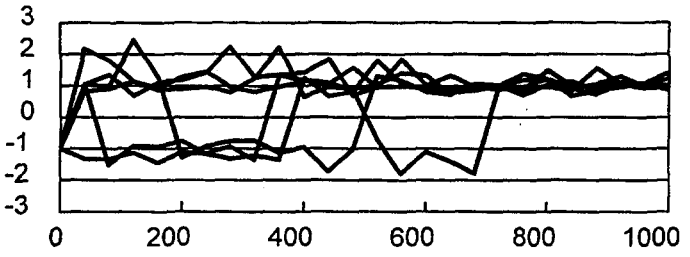

Figure 9. Six independent realizations of $\langle x_1, \ldots, x_n \rangle$ with $n = 1,000,000$.

3.5. An Improvement

An improvement suggests itself: since $\langle x_n \rangle$, for some a and σ, wanders over the domain, an improvement might be obtained by keeping track of x_i with the smallest function value. Formally, this is realized by observing also a sequence $\langle z_n \rangle$ defined by

$$z_0 = x_0; \quad z_i = x_i, \quad \text{if } f(x_i) \leq f(z_{i-1}), \quad z_i = z_{i-1}, \quad \text{otherwise.}$$

We have not seen this recommendation in the literature, perhaps because of strong belief by authors that an improvement to simulated annealing is not necessary. msd for z_n are given in Table 9. The half-widths are, as before, small in comparison with msd.

We see a drastic improvement over x_n for some, but not all a and σ. In particular, the values $a = 0.3$, $\sigma = 1$ are no longer the "best," the values $a = \sigma = 3$ are "best" now. Note that z_n is not necessarily closer to 1 than x_n. Generalization of this improvement to the case when function values and derivatives are observed with error would be not that simple.

Table 9. msd for z_n for $n = 10,000$, method K, domain $[-3,3]$.

σ	a 0.1	0.3	1	3	10
0.1	4	4	4	4	0.42
0.3	3.97	4	4	3.92	0.57
1	8×10^{-4}	0.57	3	2.74	0.25
3	2×10^{-7}	1×10^{-7}	7×10^{-8}	$\boxed{5 \times 10^{-8}}$	6×10^{-7}
10	3×10^{-7}	3×10^{-7}	3×10^{-7}	3×10^{-7}	4×10^{-7}
30	7×10^{-7}	7×10^{-7}	7×10^{-7}	7×10^{-7}	7×10^{-7}

4. THE MULTIDIMENSIONAL CASE

4.1. The Function

We shall consider the function F, defined on $[-3,3]^k$ by

$$F(x) = \sum_{i=1}^{k} f(x_i).$$

This is a very simple function and suitable to show unsatisfactory behavior of a method and much less suitable to show a good behavior. We shall apply the various methods to F without using the fact that, for the minimization of F, it is enough to minimize f restricted to $[-3,3]$. Table 10 shows the centers msd and half-widths of the 0.99-interval estimates for x_n, with the "best" $a = 0.3$, $\sigma = 1$ and z_n with the "best" $a = 3$, $\sigma = 3$; "best" as found in the one-dimensional case.

The expected squared deviation e_k for $\langle x_n \rangle$ in our k-dimensional problem is, of course, ke_1 and the optimal a and σ do not depend on the dimension. This is not true for $\langle z_n \rangle$; we use the "best" a and σ we found in the one-dimensional case (cf. Table 9). With increasing dimension k, the improved process $\langle z_n \rangle$ loses the advantage over $\langle x_n \rangle$.

Table 10. msd and half-widths for x_n and z_n, $n = 10,000$, and several dimensions k.

	x_n		z_n		ratio of the
k	msd	hw	msd	hw	two msd
1	2.23	0.06	5×10^{-8}	6×10^{-9}	4.49×10^{7}
2	4.45	0.11	2.9×10^{-4}	1.4×10^{-5}	3.18×10^{5}
3	6.68	0.17	0.020	0.009	330.8
4	8.90	0.22	0.324	0.038	27.5
5	11.13	0.28	1.428	0.085	7.79
6	13.36	0.33	2.970	0.092	4.50
7	15.58	0.39	4.727	0.100	3.30
8	17.81	0.44	6.503	0.131	2.74
9	20.03	0.50	8.609	0.153	2.33
10	22.26	0.56	10.546	0.184	2.11

4.2. Deterministic and Random Search

In the deterministic search, $[-3,3]$ was divided into m intervals such that

$$m^k \leq 9,500,$$

and the Cartesian products of these were used to cover $[-3,3]^k$. The function was evaluated at the center point of each subinterval. We have chosen 9,500 to have an option to add, after the deterministic search, a continuing search by a gradient method. For many k, the number n

Table 11. Values of m and $n = m^k$.

k	m	n
1	9,500	9,500
2	97	9,409
3	21	9,261
4	9	6,561
5	6	7,776
6	4	4,096
7	3	2,187
8	3	6,561
9	2	512
10	2	1,024

of function evaluations is considerably smaller than 9,500 (we put the gradient method at a disadvantage, because we consider the bound 9,500 irrespective of the set of integers of form m^k).

The deterministic search was completed by a gradient method, with the gradient multiplied by 0.2, starting from the deterministic search estimate and using at most 17 additional steps.

We also used random search: the estimate is equal to the point with the smallest function value among 10,000 points randomly selected from the domain. The random search performed worse than the deterministic search for our function F (see Section 4.3 for the overall comparison).

There is a difficulty with the deterministic search in that the covering of a k-dimensional cube by dividing the edges into m subintervals leads to a very large number m^k of function evaluations for all nontrivial $m \geq 2$ and even moderately large k. This is aggravated by the difficulty of covering uniformly, at least approximately, the cube by n points with n not of the form m^k. The first difficulty may, however, only reflect the difficulty of the problem itself, related to the entropy of the unit cube in the k-dimensional space: to cover the unit cube in R^k by n cubes of diameter u, we need at least $n \geq u^{-k}$; if u^{-1} is an integer; it is enough to have $n = u^{-k}$. If a global property of a function can be answered only by evaluating the function at a point in each of those n cubes, then we need n evaluations. (Of course, we may get a correct answer, sometimes, by merely guessing.)

Some methods, proposed as improvements over the discrete search seem not to be improvements, but only have their deficiencies better hidden, and their properties less well understood.

Is the worse performance of the random search in Table 12 an exception? Consider a positive integer k, a function H_k defined on $[0,1]^k$, and a subinterval I of $[0,1]$ of length 0.1 such that on I^k, H_k is negative and on the complement it is nonnegative. Assume also that H_k, restricted to I^k, has no other stationary points except the point of global minimum. In both the deterministic and the random search, it is desirable to select at least one point in I^k. For the deterministic method, that would be guaranteed if each edge is divided into 10 subintervals, resulting in $n = 10^k$ function evaluations. Next consider the random search. The probability that a point, selected at random from $[0,1]^k$, falls outside I^k, is $1 - (0.1)^k$ and the probability that at least one among n such independently selected points falls into I^k is $1 - [1 - (0.1)^k]^n$. To have the probability at least $1 - \varepsilon$, that a point will be selected in I^k, we need $[1 - (0.1)^k]^n \leq \varepsilon$ which is approximately (with a very good accuracy) equivalent to $n \geq -(\log \varepsilon)(10^k)$. Thus, n is required to be several times larger than in the deterministic case, and the desired event has only a large probability to occur whereas in the deterministic search it is a certainty.

Similarly, procedures have been proposed many times that start from a few randomly chosen points in the domain and follow with a search for a local extreme, or a simulated annealing method. We have never seen proofs of useful properties of such methods.

Table 12. Comparison of msd for several methods and dimensions.

k	x	z	random search	deterministic search	deterministic and gr.
1	2.23	5.0×10^{-8}	1.79×10^{-7}	1.11×10^{-8}	10^{-12}
2	4.45	0.00029	0.00114	0.00021	2×10^{-12}
3	6.68	0.0202	0.027	0.061	3×10^{-12}
4	8.90	0.324	1.088	0.040	4×10^{-12}
5	11.13	1.428	3.289	1.250	5×10^{-12}
6	13.36	2.970	5.892	0.375	6×10^{-12}
7	15.58	4.727	8.649	7.000	7×10^{-12}
8	17.81	6.503	12.378	8.000	8×10^{-12}
9	20.03	8.608	19.648	2.250	9×10^{-12}
10	22.26	10.546	29.155	2.500	10^{-11}

4.3. Comparisons

For $k = 1, \ldots, 10$, Table 12 compares the performance of the simulated annealing method x, its improvement z, the random search, and the deterministic search. Included are also results for the deterministic search, using at most 9,500 function evaluations, and the continuation by a small step gradient method.

The msd for x and z are copied here from Table 10. The results for the random search have been obtained in 5,000 independent replicas. The half-widths of 0.99 interval estimates of the expected squared deviations are all less than 10% of the corresponding values of msd. The properties of the deterministic search are also in our case easy to determine from the one-dimensional case and the value of m (cf. Table 11). The gradient method used 0.2 as the constant multiplying the gradient and was instructed to stop when the deviation from 1 was at most 10^{-6}. The number of these additional steps depends on k, because of the initial point of the gradient method, and was at most 17, in all cases. The results show that the simulated annealing sequence x has the worst behavior of all the methods, except for $k = 10$, when it is slightly better than the random search. Random search is second worst. The improved simulated annealing z and the deterministic search trade ranks depending on the dimension k with the deterministic search better for most k.

However, take the following into consideration.

(i) The improved simulated annealing method is favored here by preliminary computation to help select the "best" parameters a and σ; choosing these differently might influence seriously the performance (see Table 9). Trying to choose these parameters well in an actual application will substantially increase the number of function evaluations. We do not know of any theoretical results showing the behavior of x_n or z_n for a finite n. In contrast,

(ii) the deterministic search does not depend on parameters (depends on the function considered and on the number of evaluations). Its behavior is well known for many large classes of functions, where the behavior for the worst function in the class can be found. An example of such a class is, of course, the class of all Lipschitz (C) continuous functions on a bounded subset of R^k, with a known constant C. That does not mean the method can be applied only when C is known; but in that case, the bound for the error depends on C. For smoother functions, similar but more efficient methods are known.

The number of observations, 10,000, generously rich for $k = 1$, seems rather small for $k > 6$ and dimensions larger than 6 have been included mainly to compare x and z. For this reason, we obtained some results concerning the K method, domain $[-3, 3]$ and $n = 10^7$. Restricting the number of replicas this time to 500 only, we obtained (i) an estimate 0.722 of $P\{|x_n - 1| \leq 0.2\}$ for $k = 1$ (a 0.99-interval estimate for this probability is (0.667, 0.772)). For $k = 10$, we obtain an estimate $0.722^{10} = 0.0385$ of $P\{x_n \in [0.8, 1.2]^{10}\}$. For the same k and n, we obtained these

additional results: a 0.99 interval estimate $(9.03 - 2.26,\ 9.03 + 2.26)$ of the expected squared deviation of x_n from 1. With the deterministic search, with $m = 5$ and 9,765,625 evaluations, we obtained the point $\langle 1.2, 1.2, \ldots, 1.2\rangle$. Seven additional steps by the gradient method give a point with squared deviation from 1 less than 10^{-7}.

All the methods considered would perform differently for functions different from those considered here. For example, the performance of the deterministic search followed by the gradient method would not find, with the accuracy given in Table 12, the point of the global minimum, if the deterministic method leads close to a different point of local minimum, for example, in the case last discussed, with $m = 5$, if f had a smaller value at -1.2 than at 1.2. More difficult functions than F can be easily constructed. For example, for $k = 2$, a difficult function H is such that v minimizing $H(u, v)$ strongly depends on u, so that the method cannot really find the best v before it finds the best u and vice versa.

The small step gradient method is probably better than the steepest descent method (see [17]). In case of functions with nondiagonal matrix of the second-order derivatives, second-order gradient methods outperform the simple gradient method; a very simple version of such a method has been described by Fletcher and Powell [18] and studied often since (see, e.g., [19]).

REFERENCES

1. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Physics* **21**, 1087–1092 (1953).
2. V. Černý, A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm, Preprint, Inst. Phys. and Biophys., Comenius University, Bratislava, (1982).
3. V. Černý, A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm, *J. Opt. Theory and Applications* **45**, 41–51 (1985).
4. S. Kirpatrick, C.D. Gelatt Jr. and M.P. Vecchi, Optimization by simulated annealing, IBM Research Report RC 9355, (1982).
5. S. Kirpatrick, C.D. Gelatt Jr. and M.P. Vecchi, Optimization by simulated annealing, *Science* **220**, 671–680 (1983).
6. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Proceedings Pattern Analysis and Machine Intelligence PAMI-6*, 721–741 (1984).
7. R.Z. Hasminskij, Use of random noise in problems of optimization and learning (in Russian), *Problemy Peredaci Informacii* **1**, 113–17 (1965).
8. H.J. Kushner, Asymptotic global behavior for stochastic approximation and diffusions with slowly decreasing noise effects: Global minimization via Monte Carlo, *SIAM J. Appl. Math.* **47**, 169–185 (1987).
9. B.S. Gelfand and S.K. Mitter, Recursive stochastic algorithms for global optimization in R^d, *SIAM J. Control and Optimization* **29**, 999–1018 (1991).
10. P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, Reidel, (1987).
11. P.J.M. van Laarhoven, *Theoretical and Computational Aspects of Simulated Annealing*, Center for Mathematics and Computer Science, (1988).
12. G. Pflug, Application aspects of stochastic approximation, In Ljung, Pflug, Walk, *DMV Seminar,* Band 17, Birkhäuser, (1992).
13. J.M. Steel, A review of Laarhoven, 1988, *J. Amer. Statist. Ass.* **85**, 596 (1990).
14. G. Pflug, A talk at a meeting in Oberwolfach, (1990).
15. M. Lundy and A. Mees, Convergence of an annealing algorithm, *Mathematical Programming* **34** (1986).
16. Dippon and Fabian, Stochastic approximation of global minimum points, *Journal of Statistical Planning and Inference* (1992) (to appear).
17. G.E. Forsythe, Solving linear equations may be interesting, *Bull. Amer. Math. Soc.* **59**, 299–329 (1953).
18. R. Fletcher and M.J.D. Powell, A rapidly convergent descent method for minimization, *Comp. J.* **6**, 163–168 (1963).
19. E. Spedicato, Recent developments in the variable metric method for nonlinear unconstrained optimisation, In *Towards Global Optimization* (Edited by L.C.W. Dixon and G.P. Szegö), North-Holland and Elsevier, (1975).