AMERICAN JOURNAL OF MATHEMATICAL AND MANAGEMENT SCIENCES

NEIGHBORHOOD SIZE IN THE SIMULATED ANNEALING ALGORITHM

Larry Goldstein Michael Waterman

University of Southern California
Department of Mathematics
Los Angeles, CA 90089-1113

SYNOPTIC ABSTRACT

Simulated annealing is a probabilistic algorithm that has shown some promise when applied to combinatorially NP-hard problems. One advantage of the simulated annealing algorithm is that it is based on an analogy with statistical mechanics which is not problem-specific. However, any implementation of the algorithm for a given problem requires that several specific choices be made. The success or failure of the procedure may depend on these choices. In this study we explore the effect of choice of neighborhood size on the algorithm's performance when applied to the travelling salesman problem.

Key Words and Phrases: Simulated Annealing, travelling salesman, neighborhoods

1988, VOL. 8, NOS. 3 & 4, 409-423
0196-6324/88/030409-15 $20.00

1. INTRODUCTION.

Certain discrete versions of the simulated annealing algorithm are probabilistic approachs to combinatorially NP-hard problems, that is, to a class of problems for which no polynomial time algorithms are known. The key to the simulated annealing algorithm is an analogy with statistical mechanics which is not problem specific. In a typical discrete optimization problem, one is given a finite set S, (typically large), and a cost funtion f, and seeks $v \in S$ such that $f(v)$ is a minimum. One may regard the function f as the energy function of some physical system; if one could now simulate the cooling of this system, a state of minimum energy would be obtained. Although this parallel is universal, any implementation of the algorithm requires choices to be made that are specific to the problem at hand.

In order to simulate a physical system, the algorithm proceeds sequentially by moving from one state to another by a certain probabilistic mechanism. From any given state s, there are a set of states, say N_s, where transitions from s are allowed. We call N_s the set of neighbors of s.

It is with the choice of neighborhoods N_s that this study is concerned. It seems to be often overlooked that the performance of the simulated annealing algorithm depends critically on the choice of neighborhood structure, and more importantly, that one is free to choose a system that allows the algorithm to perform well. If the choice of neighborhood is too small, then the resulting simulated process will not be able to move around the set S quickly enough to reach the minimum in a

reasonable time. On the other hand, if the neighborhoods are too large, then the process essentially performs a random search through S, with next possible state chosen practically uniformly over S. The question now arises: what choice of neighborhoods N_s will allow the algorithm to converge quickly? Intuitively, it seems that a neighborhood system that strikes a compromise between these extremes would be best.

Neighborhood structure is not the only aspect of the simulated annealing algorithm that is free to be chosen in a way that improves the performance of the algorithm; the form of the energy function may also affect the behavior of the algorithm. For example, if f is nonnegative one may contrast an implementation of the simulated annealing algorithm that minimizes f with one that minimizes $\sqrt{f}$; this problem is not studied here.

2. THE SIMULATED ANNEALING ALGORITHM.

We now describe the simulated annealing algorithm in a general setting. We begin with the underlying neighborhood system.

Consider a finite set S. For each $s \in S$, suppose there is given a subset $N_s \subset S$ that satisfies

1. $\forall s \in S, s \in N_s$.

2. $\forall s, t \in S, s \in N_t$ if and only if $t \in N_s$.

3. $\forall s, t \in S, \mid N_s \mid = \mid N_t \mid$.

4. $\forall s, t \in S$, there exists an integer m and $u_1, u_2, \ldots, u_m$ in S such

that

$$s \in N_{u_1}, u_i \in N_{u_{i+1}}, u_m \in N_t,$$

for $i = 1, 2, \ldots, m-1$. (That is, we require the graph on S constructed by joining two elements of S with an edge whenever they lie in the same set N_s for some $s \in S$, to be connected.)

We call such an indexed system of subsets $\{N_s\}_{s \in S}$ a neighborhood system.

Now assume given a cost (or energy) function f, where $f : S \to R$, it is required to locate the element of S that minimizes f. For each neighborhood system, we can consider an associated "greedy" algorithm as follows:

Algorithm $G(\{N_s\}, s \in S)$: Begin at any point $s_1 \in S$. At stage n, choose s_{n+1} to satisfy

$$f(s_{n+1}) = min\{f(t) : t \in N_{s_n}\}.$$

It is clear that after a finite number of iterations of algorithm $G(N_s)$, the state will become trapped in a local minimum of f.

The simulated annealing algorithm is a probabilistic modification of the greedy algorithm $G(N_s)$ that does not get trapped in a local minimum. This is accomplished by occasionally accepting a new state that increases the energy function. The idea of a simulation of this type was first introduced by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953). For any given $T > 0$ the Gibbs distribution over S, assigns to $s \in S$ probability

$$\pi_T(s) = \frac{exp(-f(s)/T)}{Z_T}$$

where Z_T (the "partition function") is chosen so that the above probabilities sum to 1, that is

$$Z_T = \sum_{s \in S} \exp(-f(s)/T).$$

Note that for $T > 0$ small, the Gibbs distribution concentrates its mass on favorable states, that is, states s with small values of $f(s)$, and this effect is more pronounced the smaller the value of T. One may easily construct a Markov chain that has the above distribution as its stationary law. As the Markov chain converges in distribution to this law, one may run the simulation for a time and find a state of low energy with high probability. The greedy algorithm $G(N_s)$ is essentially this Markov chain run for the case of T fixed at 0, whereas in the limit of high T all states are essentially weighted with the same probability and one is moving from a state to its neighbors uniformly.

Of course, by the above mentioned analogy with statistical mechanics, T here is seen to play the role of temperature, and one may now suspect that T may be lowered as the simulation proceeds in order to force the system to a state of minimum energy. This idea is due to Kirkpatrick, Gelatt, and Vecchi (1983). As with a physical system, temperature may be lowered too rapidly and the system may become trapped in a local energy minimum, that is, the algorithm will too closely resemble the greedy algorithm $G(N_s)$. A theorem of Geman and Geman (1984) shows that if $T_n = c/\log n$, for c sufficiently large, then the system will in fact not be trapped. With this choice of T_n, the algorithm proceeds as follows.

Algorithm $SA(\{N_s\}, s \in S)$: Choose an initial point $s_1 \in S$, uni-

formly over S. At time n, assume s_n given. From the set N_{s_n}, choose a point uniformly, say t. Calculate

$$\Delta = f(t) - f(s_n).$$

Now, set $s_{n+1} = t$ with probability $p = exp(-\Delta^+/T_n)$, and set $s_{n+1} = s_n$ with the complementary probability $1 - p$, where

$$\Delta^+ = \begin{cases} \Delta & \text{if } \Delta > 0 \\ 0 & \text{otherwise.} \end{cases}$$

In order to implement the simulated annealing algorithm $SA(N_s)$ one is required to furnish a neighborhood system N_s and a cost function f. It is exactly these elements of the algorithm that are problem specific. We now turn to a specific problem, and a description of a neighborhood system for that problem.

3. LIN'S k-NEIGBORHOOD SYSTEM FOR THE TRAVELLING SALESMAN PROBLEM

The travelling salesman problem models the salesman who is required to visit a number of cities and return home covering minimal distance. Let the "cities" $c_1, c_2, \ldots, c_N$ be independent and uniformly distributed in the unit square $[0,1]^2$, and let $d_{i,j}$ denote the distance between city i and city j. The finite set S over which we seek a minimum is the set of all permutations of $\{1, 2, \ldots, N\}$; a given permutation gives the order in which the tour of the cities is to be taken. The cost (energy) function f that is to be minimized is the total length of the tour taken in the order dictated by the permutation $s \in S$ and can be written

$$f(s) = \sum_{i=1}^{N-1} d_{s(i),s(i+1)} + d_{s(N),s(1)}.$$

This problem belongs to the class of NP hard problems, hence no polynomial time algorithm for its solution is known (see for example Garey and Johnson (1979)).

In this particular problem, while there is a "natural" choice of an energy function, as stated in the introduction there is really no reason to believe that f will be prefered to some other function on S that attains its global minimum at the same optimal tour, such as $\sqrt{f}$ or f^2 for example. Given the function f as above, one is now only required to choose a neighborhood structure for the set of permutations S.

In a study of deterministic algorithms for the travelling salesman problem, Lin (1965) introduced the notion of k-optimality, which gives rise to a neighborhood structure for each k. In the terminology used here, a tour is k-optimal if it has the smallest cost of all tours in its neighborhood. The larger the value of k, the more neighbors any given tour will have. For $k = 1$ a tour is a neighbor of itself only and hence every tour is 1-opt; for $k = N$ every tour is a neighbor of every other tour, and hence only optimal tours are N-opt.

For fixed k we define a system of neighborhoods as follows. Imagine that there is a link between any two cities in a tour. We say that two tours are neighbors if one can break k or less links in the one tour and reassemble to obtain the other. From this definition it becomes clear that two tours are neighbors for $k = 2$ if and only if one tour can be obtained from the other by reversing the order of the cities in a portion of one of the tours. For k small relative to N, the number of k neighbors of any given tour is approximately $\binom{N}{k}\frac{(k-1)!}{2}2^k$. The factor $\binom{N}{k}$ counts the

number of ways k links may be broken from the N possible, $\frac{(k-1)!}{2}$ the number of ways the k components of the broken tour may be reassembled, and the factor 2^k counts the number of orientations possible for the k components. The formula is not exact since it ignores the possibility of having components of size 1, and so a factor of 2 should not be entered for this component; indeed, for $k = N$ all components are of size 1 and no factors of 2 enter yielding the correct answer of $\frac{(N-1)!}{2}$ for the total number of possible tours. We have the term $N-1$ as we are considering the tour to be in a loop, and we may consider it to begin at city 1; the factor of 2 takes care of the fact that a given tour and the same tour taken in reverse order are to be considered equivalent. For the cases of interest below, k is small relative to N and the formula above gives a reasonable approximation to the order of growth of the neighborhood size in k.

With the above ingredients, that is, with a cost function and a neighborhood structure now fixed by a choice of k, we can implement the simulated annealing algorithm $SA(N_s)$. Our interest below is to determine which value of k allows the simulated annealing algorithm of locate the minimum quickly.

4. EFFECT OF NEIGHBORHOOD SIZE ON SPEED OF CONVERGENCE.

Bonomi and Lutton (1984) implemented a version of the simulated annealing algorithm with Lin's 2-opt neighborhoods and reported positive results. In this study, we are interested in how different choices of neighborhood system, that is different values of k, affect the performance

of the algorithm.

In Bonomi and Lutton (1984), N points are laid down uniformly in the unit square $[0,1]^2$ as described. This area is then subdivided into many smaller subsquares; using a path that tends to a space filling curve in the limit, a short path is found that tours the subsquares and the algorithm is then run independently among a group of subsquares. We will call this procedure "modified simulated annealing" for the travelling salesman problem. This modified procedure will speed up convergence to the minimum.

In our study, we consider the unmodified version of the simulated annealing algorithm $SA(N_s)$. That is, we study the simulated annealing algorithm's performance as a function of k without the above modification that speeds convergence. We have three reasons for making such a study.

First, the heuristic used in Bonomi and Lutton (1984) is highly problem specific as it relies on the fact that points close together in $[0,1]^2$ are likely to be close together in the optimal tour. In fact, one takes advantage of knowing the average intercity distance in the optimal tour (see Bearwood, Halton, and Hammersley (1959) and the discussion below) and therefore, a priori, need only consider moves that result in intercity distances on this order. In many problems, among them even problems such as those in Goldstein and Waterman (1987) that bear significant resemblance to the travelling salesman problem, one does not have such a priori information about the solution, and therefore cannot build a heuristic that uses this information to advantage. Therefore we

retain more generality by considering the unmodified algorithm .

Second, it is preferable not to complicate the outcome of the simulation with the choice of some particular heuristic that may affect the results of the study in an unknown way. That is, with the modification, the choice of k is confounded with the choice of heuristic. In short, our second reason for making this study is that throughout, we are more interested in the simulated annealing algorithm in general than its performance for this problem in particular .

Lastly, even as applied to the problem at hand, if one were to adopt the subdivision approach for the travelling salesman problem, one would always be solving the unmodified version of the problem on subsquares anyway and would still like to be using the best value of k on each subproblem. (In any implementation designed to actually solve the travelling salesman problem it would certainly be advisable to adopt a heuristic such as the subdivision approach in order to speed convergence).

In most minimization problems, one is not usually given in advance the value of the cost function at the minimum. In fortuitous cases where this value is known, the information can be used to devise a stopping rule for a procedure to halt when it gets sufficiently close to the minimum. In addition, this value can be used to gauge how well an algorithm performs against such a standard.

The travelling salesman problem with a uniform city distribution is an example of a problem where the value of the cost function is known at the minimum (that is, the optimal tour length is known), in a probabilistic sense in the limit for many cities. (For an example where the value of

the cost function at the minimum is known deterministically for any size problem see Goldstein and Waterman (1987).) A result of Bearwood, Halton, and Hammersley (1959) shows that the length l_N of the optimal tour that connects N cities put down uniformly in $[0,1]^2$ obeys

$$\lim_{N\to\infty} \frac{l_N}{\sqrt{N}} = \beta, \text{with probability one}$$

where Monte-Carlo simulation puts the constant β at approximately 0.749 (Bonomi and Lutton (1984)). Using this result, given a particular tour, we can say how far away we are from the optimal tour.

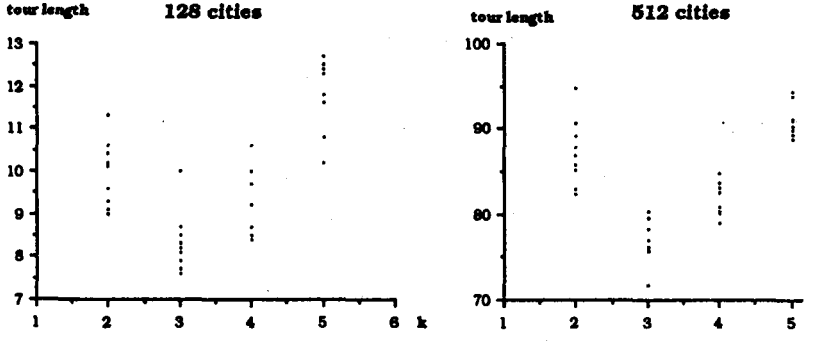
We now describe the simulation for the unmodified case. As alluded to above, the unmodified version of the algorithm is clearly an impractical way to solve the travelling salesman problem, particularly if N is large. However, the simulation described in this section is valuable as it yields a clear pattern for the choice of the optimal neighborhood size k.

As noted above, if the neighborhood size is small relative to the size of S, the Markov chain cannot move around the state space fast enough to find the minimum in a reasonable time. On the other hand, a neighborhood too large has the algorithm merely sampling randomly from a large portion of the state space; this is most clearly seen in the extreme case where $N_s = S$. It is therefore reasonable to expect that the best value of k furnish a compromise between these two conflicting extremes. Furthermore, regarding these considerations, it may be the case that the best value of k depends on N, as in fact we observe.

The simulation was run by random number generating N independent and uniformly distributed points in the unit square $[0,1]^2$ to serve

as locations for the N cities. Random numbers were generated using the generalized feedback shift register psuedorandom number algorithm of Lewis and Payne (1973). For various choices of k a random permutation was generated as the initial condition and the algorithm was run with temperature decreased as described in section 2.

As we do not expect to obtain the best tour for any large value of N, we gauge the effectiveness of a choice of k by running the algorithm for a fixed number of iterations and graphing l, the value of the best tour found by the Markov chain up to this time, versus the value of k used in the algorithm. For example with 128 cities independently and uniformly distributed in the unit square, we expect the shortest path to be of length $l_N = \sqrt{128}(0.749) = 8.47$. We find from Figure 1 that after 10,000 iterations and with $k = 2$ the best tour found was roughly 18.5 units long, about 10 units longer than the optimal tour, while using the value $k = 3$ we located tours roughly 16.5 units long, about 8 units longer than optimal. As the figure shows, using the value of $k = 4$ for this N is markedly worse than using $k = 3$.



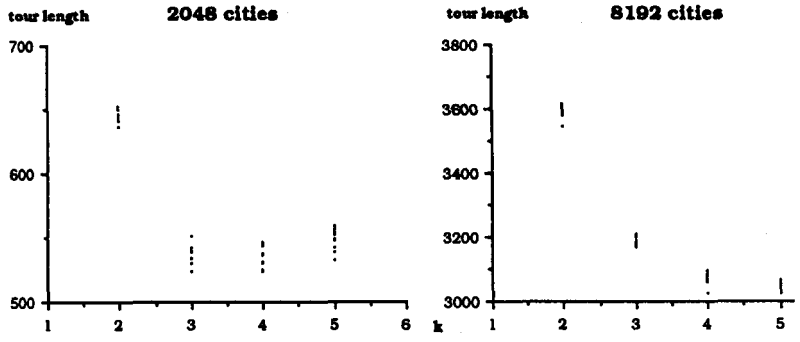


FIGURE 1. Length of shortest tour found in 10,000 iterations by the simulated annealing algorithm using k -opt neighborhoods for various problem sizes.

From Figure 1, we see that for N=128 or 512 cities the optimal value of k is 3, while for 2048 cities a k of 3 or 4 performs better than any other value of k, and for 8182 cities, the optimal value of k suggested by the figure is 5 (or, perhaps larger). It would of course be of much interest to quantify in more detail the behavior of the slowly growing optimal value of k as a function of N.

The results of the simulation above for the unmodified case recommend the choice k=3 even for problems of size 128. This suggests that an improvement can be made to the procedure in Bonomi and Lutton (1984) where the choice $k = 2$ is used throughout.

5. CONCLUDING REMARKS.

The simulated annealing algorithm can be a useful tool to apply to hard combinatorial problems, and although one appeal of the algorithm

is its apparent universality, one must keep in mind that some care must be taken in application as each implementation requires choices that essentially determine the actual efficacy of the procedure.

ACKNOWLEDGMENTS

The authors wish to thank Mark Eggert for valuable programming support.

REFERENCES

Bearwood, J., Halton, J.H., and Hammersley, J.M., (1959). The shortest path through many points. Proceedings of the Cambridge Philosophical Society, 55, 299-327.

Bonomi, E. and Lutton, J-L. (1984). The N-city travelling salesman problem: statistical mechanics and the Metropolis algorithm. Society for Industrial and Applied Mathematics Review, 26, 551-568.

Garey, M.R. and Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completness. W.H. Freeman, San Francisco.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 721-741.

Goldstein, L. and Waterman, M. (1987). Mapping DNA by stochastic relaxation. Advances in Applied Mathematics, 8, 194-207.

Hajek, B. (1985). Cooling schedules for optimal annealing. Mathematics of Operations Research. Submitted.

Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P. (1983). Optimization by simulated annealing. Science, 220, 671-681.

Lewis, T. and Payne, W. (1973) Generalized feedback shift register psuedorandom number algorithm. Journal of the Association for Computing Machinery, Vol. 29, no.3 pp. 456 - 468.

Lin, S. (1965). Computer solutions of the travelling salesman problem. Bell System Technical Journal , 44, 2245-2269.

Metropolis, M., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. Journal of Chemical Physics,21, 1087-1092.

Received 5/31/88; Revised 12/15/88.