JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 45, No. 1, JANUARY 1985

Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm[1]

V. ČERNÝ[2]

Communicated by S. E. Dreyfus

Abstract. We present a Monte Carlo algorithm to find approximate solutions of the traveling salesman problem. The algorithm generates randomly the permutations of the stations of the traveling salesman trip, with probability depending on the length of the corresponding route. Reasoning by analogy with statistical thermodynamics, we use the probability given by the Boltzmann–Gibbs distribution. Surprisingly enough, using this simple algorithm, one can get very close to the optimal solution of the problem or even find the true optimum. We demonstrate this on several examples.

We conjecture that the analogy with thermodynamics can offer a new insight into optimization problems and can suggest efficient algorithms for solving them.

Key Words. Traveling salesman problem, Monte Carlo optimization, importance sampling.

1. Introduction

In this paper, we present a Monte Carlo algorithm to find approximate solutions of the traveling salesman problem. It simulates the behavior of a statistical system which is equivalent to the traveling salesman problem in the following sense. Each permutation of the stations on the traveling salesman trip is considered to be the configuration of the statistical system. The corresponding length of the trip is then called the energy of the system in that particular configuration.

[1] The author acknowledges stimulating discussions with J. Pišút concerning the main ideas of the present paper. The author is also indebted to P. Brunovský, J. Černý, M. Hamala, Š. Peško, Š. Znám, and R. Zajac for useful comments.

[2] Staff Member, Institute of Physics and Biophysics, Comenius University, Mlynska Dolina, Bratislava, Czechoslovakia.

0022-3239/85/0100-0041$04.50/0

Statistical thermodynamics tells us that large systems at a given temperature approach spontaneously the so-called equilibrium state, characterized by a certain mean value of the energy, depending, of course, on the temperature. The temperature is a rather arbitrary parameter for the traveling salesman problem. However, simulating the transition to the equilibrium and decreasing the temperature, one can find smaller and smaller values of the mean energy of the system (= length of the trip). We shall demonstrate on several examples that, by this method, we obtain an efficient algorithm for getting close to the optimal solution of the traveling salesman problem or even finding the true optimum.

The algorithm is very simple, and it could perhaps be proposed without any reference to statistical physics. However, we feel that the thermodynamical analogy can be used as a general strategy to approach optimization problems characterized by a big combinatorial complexity. We therefore describe in the next section a few basic principles of statistical thermodynamics which are essential for our reasoning. The details of the algorithm are then presented in Section 3. In Section 4, we demonstrate the efficiency of the algorithm on a few examples. The last section contains the discussion and conclusions.

2. Brief Review of Basic Principles of Statistical Thermodynamics

We describe here basic principles of statistical thermodynamics (Ref. 1) on a simple example. We consider a one-dimensional lattice. In each point of the lattice, there is a small arrow (representing the spin of a particle), which can exist in two states: pointing up or down (Fig. 1). If the lattice has N points, there are 2^N different possibilities of how to arrange the orientation of the arrows. We say that there are 2^N possible configurations of the system. We further suppose that, in each configuration, the system has a definite energy (not necessarily different for different configurations). For example, one can have

$$E = B(n_+ - n_-),$$

where n_+ (n_-) is the number of arrows oriented up (down) in the configuration considered and B is some constant[3].

Practical experience tells us that large systems of this kind approach spontaneously the equilibrium state, in most cases irrespectively of the initial configuration. The equilibrium is, however, by no means a static situation. In equilibrium, the system randomly changes its state[4] from one

[3] Such a system represents the behavior of noninteracting spins in an external magnetic field.

[4] Because of interaction with the environment.

Fig. 1. One-dimensional spin system on a lattice.

possible configuration to another in such a way that the probability of finding the system in a particular configuration is given by the Boltzmann-Gibbs distribution

$$P(\text{conf}) = C \exp(-E_{\text{conf}}/T). \tag{1}$$

Here, E_{conf} is the energy of the configuration, T is the temperature, and C is a normalization constant. The mean energy of the system in the equilibrium is then

$$\bar{E} = \left[\sum_{\text{conf}} E_{\text{conf}} \exp(-E_{\text{conf}}/T)\right] \Big/ \left[\sum_{\text{conf}} \exp(-E_{\text{conf}}/T)\right]. \tag{2}$$

The numerical calculation of $\bar{E}$ might be quite difficult if the number of configurations is high. However, one can simulate the behavior of the system on a computer and, in many cases, one can in this way find experimentally the value $\bar{E}$ (and any other physical quantity of interest).

All that one needs is a Monte Carlo algorithm which simulates the random change of the state of the system from one configuration to another in such a way that in the equilibrium Eq. (1) holds. An algorithm of this kind is more or less standard (Ref. 2). We shall describe a particular example of such an algorithm in the next section.

Typically, if one starts the simulation with an arbitrary configuration, one reaches the equilibrium state after a reasonable number of Monte Carlo trials. One recognizes this by the fact that the energy of the configurations generated starts to fluctuate around some value, which corresponds, of course, to $\bar{E}$ (Fig. 2).

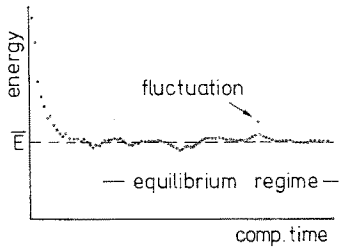

Fig. 2. Transition to equilibrium. Sketch of the typical results of computer simulation.

3. Algorithm

We start by introducing some notations. We consider the traveling salesman problem for N stations. Let D be a $N \times N$ matrix with the elements $D(i,j)$ giving the distance from the ith station to the jth station. Let $\{s_i\}_1^N$, $\{c_i\}_1^N$, $\{t_i\}_1^N$ denote permutations[5] of integers $1, 2, \ldots, N$. Then, the problem is to find a permutation c for which the total length

$$d = D(c_N, c_1) + \sum_{k=1}^{N-1} D(c_k, c_{k+1})$$

is minimal. We propose the following algorithm.

Step 0. Choose an arbitrary starting permutation $\{\hat{s}_i\}_1^N$. Choose a real number (temperature) T.

Step 1. Set $c_k = s_k$, for $k = 1, 2, \ldots, B$. Calculate the corresponding length

$$d = D(c_N, c_1) + \sum_{k=1}^{N-1} D(c_k, c_{k+1}). \tag{3}$$

Step 2. Set $i = 1$.

Step 3. Generate randomly an integer j, $1 \leq j \leq N, j \neq i$.

Step 4. Construct a trial permutation from the current permutation as follows.[6] Find

$$\tilde{i} = \min(i,j), \qquad \tilde{j} = \max(i,j).$$

Set

$$t_k = c_k, \qquad k = 1, 2, \ldots, i-1,$$

$$t_{\tilde{i}+k} = c_{\tilde{j}-k}, \qquad k = 0, 1, 2, \ldots, \tilde{j} - \tilde{i},$$

$$t_k = c_k, \qquad k = \tilde{j}+1, \tilde{j}+2, \ldots, N.$$

Step 5. Calculate the length corresponding to the trial permutation[7]

$$d' = D(t_N, t_1) + \sum_{k=1}^{N-1} D(t_k, t_{k+1}). \tag{4}$$

[5] A permutation of integers corresponds to a particular way to make the trip. For example, the permutation $\{4, 2, 1, 3, \ldots\}$ means that the traveling salesman has to go first to the fourth station, then to the second, then to the first, etc. We shall call the permutations s, c, t as starting, current, and trial permutations.

[6] By this construction, the stations in the ith and jth positions are interchanged and the route between them is taken with the reversed direction.

[7] Note that, if the matrix D is symmetric, then only two terms in the sum (4) are different from the terms in the sum (3). This fact can be used to speed up the calculations.

Step 6. If $d' < d$, go to Step 7 (see Footnote 8); otherwise[9], generate a random number x, $0 < x < 1$. Then, if $x < \exp[(d - d')/T]$, go to Step 7 (see Footnote 8); otherwise, go to Step 8 (see Footnote 10).

Step 7. Set $c_k = t_k$, $k = 1, 2, \ldots, N$. Set $d = d'$.

Step 8. Increase i by one. Then, if $i \leq N$, to to Step 3; otherwise, go to Step 2.

The algorithm, as it stands, is not finite. It generates an infinite sequence of current permutations. So, some comments are in order.

(i) For a given problem, one has to find an appropriate value of the temperature T. One can either make some educated guess[11] or one can simply find the right value experimentally by a trial-and-error method.

(ii) One has to run the calculation until an equilibrium seems to be reached.[12] Then, one should decrease the temperature and start the calculation again until a new equilibrium is reached. Repeating this, one should find smaller and smaller values of length d, and one has to decide where to stop the procedure according to some suitable criteria.

(iii) In order to increase the acceptance rate, it is sometimes useful to change Step 3 of the algorithm into the following step.

Step 3'. Generate randomly an integer j, such that $1 \leq m \leq n < N/2$, where n is some fixed number and

$$m = \min\{|j - i|, |j - i + N|, |j - i - N|\}.$$

Using this option, the change from the current permutation to the trial permutation is more local. Empirically, we found that this option is useful in situations near the equilibrium, where taking n of order $N/10$ increases the acceptance rate reasonably.[13]

4. Some Examples

Example 4.1. We consider 100 points uniformly distributed on a unit circle. We take the distances between the points to be Euclidean. The optimal

[8] The trial permutation is accepted as a new current permutation.

[9] It is here where the analogy with thermodynamics is used. The trial configuration will be accepted with probability given by the Boltzmann factor.

[10] The trial permutation is not accepted as a new current permutation.

[11] According to the analogy with thermodynamics, a typical value of the temperature for a particular problem should be related to the mean distance between the stations.

[12] See the examples of the next section.

[13] One can perhaps find other methods of increasing the acceptance rate, specific to a particular problem.

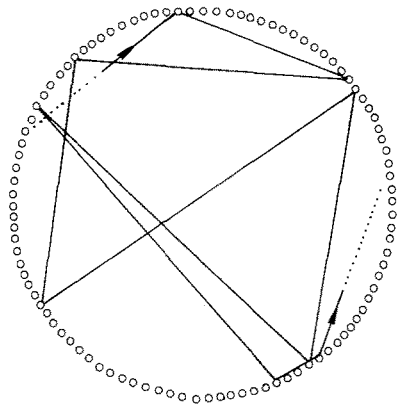

Fig. 3. Part of the route corresponding to the initial configuration, Example 4.1.

path is clearly that which goes around the circle, and its length is 6.28. We start with a path corresponding to a randomly generated permutation of the points. In our particular case, its length is 150.9 (Fig. 3). We applied the proposed algorithm using the value $T = 0.1$. The results are presented in Fig. 4, where we plot the length of the path corresponding to the current permutation after every 200 Monte Carlo trials. Qualitatively, the picture has the character expected by the thermodynamical analogy. The equilibrium regime seems to begin after some 4000 Monte Carlo trials, and the corresponding mean length is about 20. After 6600 trials we decided to decrease the temperature T to the value $T = 0.01$. Again, a new equilibrium was reached. We repeated the procedure again, choosing $T = 0.001$ and using the modified algorithm with Step 3′ instead of Step 3. After a total of 25,000 Monte Carlo trials a value 6.28 was reached, which means that

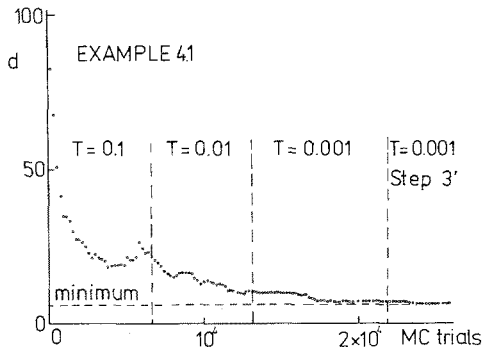

Fig. 4. Results of the computer simulation, Example 4.1. In the last part of the calculation, Step 3′ was used, instead of Step 3 of the algorithm.

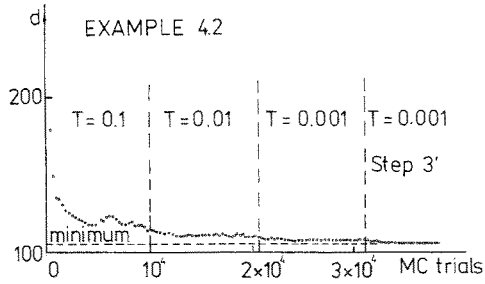

Fig. 5. Results of the computer simulation, Example 4.2.

the true optimum was found. It corresponds to finding one particular permutation out of 99! possible configurations. Since this might be due to a rather special geometry in this case, we investigated the following example.

Example 4.2. We consider again 100 points with the distances given by

$$D(i,j) = \tilde{D}(i,j) + P_i + P_j, \qquad i,j = 1, 2, \ldots, 100,$$

where $\tilde{D}(i,j)$ are the distances as given by the previous example and $\{P_i\}_1^{100}$ is a set of randomly chosen numbers.[14] It is easy to see that the optimal solution for this case is given by the same permutation as in the previous example, although the geometry of the problem is now completely different and, because of its Monte Carlo origin, in some sense unbiased.[15] We proceeded essentially in the same way as in the previous example. The results are presented in Fig. 5. It is really surprising that the true optimum was again found in this case after some 40,000 Monte Carlo trials.

Example 4.3. We investigated 100 points arranged in a rectangular lattice and considered the Euclidean distances between the points. After 50,000 Monte Carlo trials, we found the path presented in Fig. 6. Its length is 103.3 units of the lattice spacing, while the optimum value is clearly 100. However, one can proceed with the same algorithm and wait for even better results.

Example 4.4. To see whether the algorithm can get rid quickly of permutations which are obviously far from the optimum, we considered 200 points distributed randomly within two unit squares, which were 10 units apart (Fig. 7). The points numerated by odd numbers lied in the first square, and those numerated by even numbers lied in the second square.

[14] We used $P_i = i\sqrt{2} - [i\sqrt{2}]$.

[15] The author thanks Š. Peško for suggesting this example.

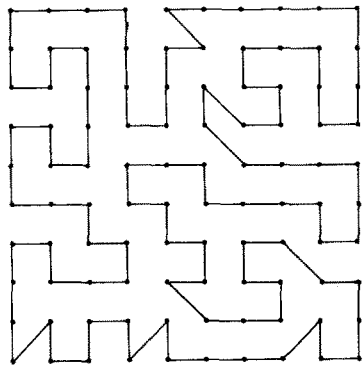

Fig. 6. Best path found for the problem of Example 4.3.

We started the minimization with the permutation 1, 2, 3, . . . , 199, 200. The corresponding path contains 200 jumps from one square to the other, instead of just two as it clearly happens in the otherwise unknown optimal path. We found that, after some 12,000 trials at the temperature $t = 0.01$, the algorithm got rid of all the 198 unnecessary jumps.

5. Comments and Conclusions

We demonstrated on several examples that, using the proposed algorithm, one can get quite close to the optimal solution of the traveling salesman problem. In some cases, we even found the true optimum. It would be nice if we could present here some rigorous theoretical results concerning our algorithm, the conditions for its convergence, etc. This is, however, not the case. We have, at least at present, nothing to add to our heuristic arguments based on the analogy with thermodynamics. Anyhow, from an utilitarian point of view, it seems that the proposed algorithm might be useful in solving practical problems, although larger computational experience is still necessary.

One can even speculate that the algorithm (or at least its main principles) might be used also as a general-purpose optimization procedure.[16]

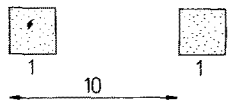

Fig. 7. Geometry of the problem, Example 4.4.

[16] We do not, however, try to specify the class of optimization problems which can be approached by this procedure.

This belief is based on the fact that the algorithm operates with only two concepts: the configuration (which we shall denote now as ξ) and a function $F(\xi)$ which is to be minimized. These two concepts are typical of any optimization problem. The other details, specific for the traveling salesman problem, are of marginal importance for our procedure. Of course, the configuration might mean quite different things for different problems. It might mean the set of particular values of some variables or, for example, a particular order of manipulations to assemble a certain product.

The optimization procedure should work as follows. Having a current configuration ξ, a trial configuration $\tilde{\xi}$ is to be generated, which, however, should be (in some sense) close to the configuration ξ. Then, $F(\tilde{\xi})$ is to be calculated, and one has to decide whether to accept $\tilde{\xi}$ as a new current configuration. Perhaps, the main idea of our traveling salesman algorithm is that, even in the case when $F(\tilde{\xi}) > F(\xi)$, the trial configuration $\tilde{\xi}$ should be considered for acceptance. Our proposal was to accept it with the probability given by the Boltzmann factor[17]

$$\exp((F(\xi) - F(\tilde{\xi}))/T).$$

The meaning of the term "$\tilde{\xi}$ close to ξ" should be specified with respect to the particular optimization problem. This is an important point, since it is here that the procedure differs from a crude Monte Carlo optimization, where one just randomly generates trial configurations independently of previous trials and, therefore, irrespectively of all the information about the problem gained by previous trials. Allowing only configurations $\tilde{\xi}$ which are close to the current configuration ξ, one introduces an effective importance sampling into the Monte Carlo method. However, this is a two-edged procedure. If one allows only configurations $\tilde{\xi}$ which are too close to ξ, one can get into trouble with shallow local minima. We demonstrated it on the traveling salesman problem.

Instead of Step 4 of our algorithm, we also tried to use the following step.

Step 4′. Construct the trial permutation as follows. Set

$$t_k = c_k, \qquad k = 1, 2, \ldots, N, k \neq i, j,$$

$$t_i = c_j,$$

$$t_j = c_i.$$

Here, only the stations in the ith and the jth positions are interchanged without reversing the route between them, as in Step 4. This looks

[17] The analogy with thermodynamics suggests that this acceptance criterion can perhaps be used universally, irrespective of the particular optimization problem. In principle, one could try to use other increasing functions, instead of exponential functions.

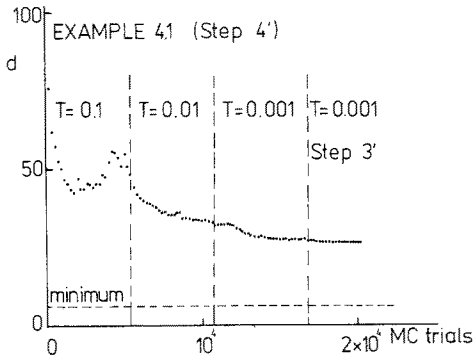

Fig. 8. Results of the computer simulation, Example 4.1. In the calculation, Step 4′ was used, instead of Step 4 of the algorithm.

reasonable since by this procedure one can reach (in several steps) any permutation.

Modifying the algorithm in this way, we obtained for Example 4.1 the results presented in Fig. 8. When compared to Fig. 4, the qualitative behavior is the same. Quantitatively, however, the results are different significantly. The mean energy found at corresponding temperatures is much higher when using Step 4′. This means that some metastable equilibrium was found in this way. In other words, the computation ended in a local minimum. The same behavior was observed in other examples.

We investigated the problem more closely, and our conjecture is that the algorithm using Step 4′ is not able to untie efficiently the loops in the route (such as those seen in Fig. 9). To undo such a loop, it might be necessary to go through many configurations of higher energy. This is in principle possible, but highly improbable. The construction of Step 4 is chosen with the aim of undoing such loops quickly.

One cannot avoid mentioning analogous situations in physics where various topologically stable objects, like magnetic vortex lines, are well known. Changing Step 4′ to Step 4 means, in this language, changing the type of interaction of the system with the environment. This can make such strange objects unstable.

We do not want to argue that the construction corresponding to Step 4 is a universal method for avoiding local minima in the traveling salesman

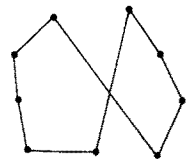

Fig. 9. Example of loop in the path, which causes difficulties when using Step 4′ instead of Step 4 of the algorithm.

problem. Our experience with nonlinear optimization suggests that it is impossible to find such a universal method. It is more practical to have a package of various optimization procedures and to choose the one which is empirically the most efficient for a particular problem. So, it might well happen that, in some cases, it would be better to use Step 4′, instead of Step 4, in the traveling salesman algorithm.

In this respect, one can also use the possibility of changing the temperature to get out of a local minimum. Increasing the temperature, one gets higher fluctuations; the configurations with higher energies become more probable, and this hopefully can help to get through the potential barrier around the local minimum. After decreasing the temperature again, one can hope to find a better approximation to the true optimum.

There is perhaps more poetry than mathematics in the speculations presented in this section. However, we believe that detailed mathematical studies of the proposed type of optimization algorithms are both possible and worth the labor, provided the experience shows their efficiency. The main purpose of the present paper is to show, on a few examples, that the algorithm, inspired by the way in which thermodynamical systems with many degrees of freedom really work, may be efficient for solving optimization problems in which many degrees of freedom make the situation rather complicated. These algorithms might be even more appropriate for the next generation of multiprocessing machines, where different degrees of freedom can be treated simultaneously.

In thermodynamics, it is well known that the equilibrium state is a state with maximum entropy (subject to some constraints specified by the external conditions). In this sense, the transition to an equilibrium is an optimization problem: the search for the maximum of the entropy. It might be surprising that our simple algorithm worked so well in the examples described above. We believe that this is caused by the fact that our algorithm simulates what Nature does in looking for the equilibrium of complex systems. And Nature often does its job quite efficiently.[18]

References

1. KITTEL, C., *Thermal Physics*, John Wiley and Sons, New York, New York, 1969.
2. METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., and TELLER, E., *Equation of State Calculations by Fast Computing Machines*, Journal of Chemical Physics, Vol. 21, pp. 1087–1092, 1953.

[18] After the paper was submitted for publication, we learned from news published in the May 1982 issue of *Physics Today* that a similar approach to the traveling salesman problem had recently been used by S. Kirkpatrick, D. Gelatt, and M. Vecchi and that essentially the same strategy for solving the optimization problems had been used by K. Wilson, D. Jacobs, and J. Prins and by M. Kalos.