Proc. 6th Ann. Allerton Conf. on Circuit and System Theory, Univ. of Ill., Urbana, Ill., Oct. 1968, pp. 814-821.

KENNETH STEIGLITZ

SOME IMPROVED ALGORITHMS FOR COMPUTER SOLUTION OF THE TRAVELING SALESMAN PROBLEM*

K. STEIGLITZ AND P. WEINER
Department of Electrical Engineering, Princeton University
Princeton, New Jersey 08540

ABSTRACT

Two techniques are described for improving Lin's method for obtaining locally optimal solutions to the traveling salesman problem. The first, called dynamic 3-opt, is inductive in nature and produces 3-opt solutions two to three times faster than the basic method described by Lin. The second technique, called linear 2-opt, takes advantage of distance ordering, and requires computation time linear in the number of cities, as opposed to the quadratic time dependence exhibited by the basic method. This approach can be extended to 3-opt and increases the size of problem which can practicably be solved by computers.

INTRODUCTION AND SUMMARY OF PREVIOUS WORK

The traveling salesman problem has received much attention. Briefly, the problem can be stated as follows: Given an nxn distance matrix (d_{ij}), determine a permutation of the integers from 1 to n, $(i_1,\ldots,i_n)$ which minimizes the sum $d_{i_1 i_2} + d_{i_2 i_3} + \ldots + d_{i_n i_1}$. We shall consider only the cases when (d_{ij}) is a symmetric matrix.

Previous work may be divided into two general categories: exact and heuristic.

Exact techniques

Although complete enumeration requires the evaluation of (n-1)!/2 tours, Held and Karp [1] and Bellman [2] have shown that the application of dynamic programming reduces the time required for solution from a factorial to an exponential function of n. Little, et.al. [3] and Agin [4] applied a branch and bound method which reduces the time for solution still further, but which still requires exponential time. Other exact techniques include linear programming [5,6], and the method of Croes [7]. The time required by all these algorithms grows so fast with the number of cities that they become impractical for moderate size problems (25-50 cities). For this reason a number of approximate techniques have been developed. Many of the papers referenced above concentrated on approximate applications of the exact techniques. On the other hand there are approximate methods which are not derived from exact techniques, and these will be discussed next.

Heuristic techniques

Karg and Thompson [8] combine two heuristic methods: The first is a fast way to obtain good but not necessarily optimum tours, which works well for convex problems. The second is a way of decomposing the original problem into convex pieces. The

*The research reported here was supported in part by the Bell Telephone Laboratories and in part by the U.S. Army Research Office - Durham under Contract DA-31-124-ARO-D-292.

work of Roberts and Flores [9] is notable in its application of statistical analysis to reduce the size of the problem to one small enough to be completed exhaustively.

In 1965 Reiter and Sherman [10] and Shen Lin [11] suggested the use of fast local optimization techniques applied repeatedly to random initial tours. It should be mentioned that the attainment of the absolute minimum length tour is not the only objective of these methods, which generate many distinct solutions near the optimum. This approach has provided by far the most dramatic reduction in the computation time required to obtain good solutions to large problems. In particular, the computation times documented by Lin [11] represent a great improvement over previously published results, and his approach provides a starting point for our study.

LIN'S METHOD - 3-OPTING

Lin's basic method is to apply a fast but powerful transformation repeatedly to a random tour until no further improvements are possible. The resulting stationary solutions are defined to be locally optimal with respect to the particular transformation. The power of the technique is derived in part from the careful selection of the particular transformation. Lin has also demonstrated that it is possible to extract common features from a collection of distinct locally optimum tours, and that this leads to a reduction technique which considerably reduces computation time.

The particularly successful transformation used by Lin is called 3-opting and is a generalization of the procedure introduced by Croes [7], which can be though of as 2-opting. In general a tour is called λ-opt [11] if it is not possible to obtain a tour with smaller cost by replacing any λ of its links by any other set of λ links. The transformation involves sequentially examining all ways of interchanging λ links of a given tour, and selecting the first one which produces an improvement.

The overall strategy proposed by Lin consists of starting with random initial tours and obtaining a number of distinct 3-opt tours. By examining common links of these tours, a reduction procedure is applied which commits certain links and effectively reduces the size of the problem for the next round of 3-opting. The reduced problem is attacked in the same way as the original. Although the technique of reduction introduces the danger of improperly committing suboptimum features to the tour at an early stage, it can significantly reduce the time spent on a particular problem.

DYNAMIC 3-OPTING

We shall now describe an alternate method for obtaining 3-opt tours, which we shall call <u>dynamic 3-opting</u>. Experimental results indicate that this method produces 3-opt tours two to three times faster than the basic method proposed by Lin. The method has the property that the 3-opt tours obtained are, on the average, as good as those produced by Lin's 3-opt procedure.

Dynamic 3-opting proceeds as follows, given a starting tour T_0. A starting city and an orientation are selected, so that T_0 can be characterized by the ordered list of indices $i_1, i_2, \ldots, i_n$. The basic idea is to obtain a 3-opt tour for the subproblem consisting of the cities $i_1, \ldots, i_{k+1}$ from a 3-opt tour for the subproblem consisting of the cities $i_1, \ldots, i_k$. This is done inductively as follows: There is only one tour for the subproblem consisting of the cities i_1, i_2, i_3, and this must be 3-opt. Now suppose a 3-opt tour has been obtained for the k-city subproblem. An initial tour for the (k+1)-city subproblem is obtained by inserting city i_{k+1} between two consecutive cities of the previously obtained 3-opt tour, and this is done by examining the incremental costs of the k possible tours and choosing the one which results in the minimum cost for the (k+1)-city subproblem. Although the tour obtained in this way is not necessarily 3-opt, experience indicates that it often is. In any event, we need only check those triples of links which include at least one of the two links connected to i_{k+1}. If and when an improvement is obtained by replacing a triple of links with another triple of links, it becomes necessary to check triples which include the newly added links. Thus a list of links is generated which has the property that we need only check triples of links which include members of this list. Proceeding in this way a 3-opt tour is obtained for the (k+1)-city subproblem.

Table 1 shows a comparison of the running times of Lin's 3-opt procedure and dynamic 3-opt on typical problems of size 10 to 57 cities. Both algorithms were coded in Fortran IV with equal sophistication, and run on the CDC 6600 computer. The times shown are the result of averaging 100 3-opt tours from random starts. Also shown are the average percent errors and the number of times correct out of one-hundred trials.

	Dynamic 3-opt			Lin's 3-opt		
Problem (# of cities)	average time in seconds	average % error	number correct per 100	average time in seconds	average % error	number correct per 100
10 ([12])	.038	.6	34	.057	.7	26
20 ([7])	.327	1.9	36	.517	1.5	46
25 ([1])	.378	.7	49	1.17	.75	56
29 ([13])	.683	.9	11	1.72	.9	23
33 ([8])	.956	.75	43	2.63	.6	51
42 ([5])	1.93	.9	31	6.93	1.1	35
48 ([1])	3.06	1.1	8	10.10	1.6	7
48 ([1], [14])	2.96	8.9	5	5.84	11.4	0
57 ([8])	5.31	1.5	0	18.23	1.9	1

Table 1. A comparison of the running times of Dynamic 3-opt and Lin's 3-opt.

During the course of computer experimentation, several heuristic techniques were developed which seemed to work very well on some problems and not so well on others. For example, an attempt was made to provide initial tours which were effective in combination with dynamic 3-opt. One such method was the nearest neighbor algorithm. In this method of generating an initial tour, a starting city is selected, the city nearest to that is chosen as the next city in the tour, and so on, until a complete tour is generated. Because the beginning of a nearest neighbor tour is more likely to be in the optimum tour than the end, the nearest neighbor tour was fed into the dynamic 3-opting procedure in an order reverse to the one in which it was generated. In this way it was hoped that the poor links would be corrected at an early stage in the optimization. One problem that was especially amenable to this combination of nearest neighbor and dynamic 3-opt was the 48 city Knight's tour, for which the optimum was found 11 times when the nearest neighbor algorithm was started from each of the possible 48 cities. On the other hand, the procedure never produced the correct answer for the 57 city problem although it did produce very good tours. On the 105 city problem [11] the nearest-neighbor-reversed-dynamic 3-opt procedure produced a tour of length 23076, which is lower than any reported previously.

Dynamic 3-opt has the interesting feature that it may be iterated; that is, the tour resulting from one application may be used as an initial permutation for another application, by selecting a starting city and direction. The following strategy was devised to exploit this feature: Once a dynamic 3-opt tour was obtained, it was fed into the dynamic 3-opt procedure with different starting points and directions until a tour with lower cost was found. By repeating this it was found that one could hill-climb to the global optimum. In point of fact, the conjectured global optimum was obtained in this manner for every problem tried, up to and including the 57 city problem. The method has the disadvantage that 2n local optima must be checked out at the final stage before it can be determined that no further improvement is possible. Experimental evidence indicates, moreover, that the expected time before obtaining the conjectured global optimum is not significantly less than the time taken by dynamic 3-opting from random starts. This fact is somewhat mitigated by the feature of having a definite termination point.

Another variation of some interest involves the concept of link denial. First a 3-opt tour is obtained. Then a link in the tour is temporarily assigned a high cost, effectively excluding it from future tours. The dynamic 3-opt procedure is then entered at its last stage, since most triplets of the present tour have already been tested. This quickly yields another tour which may or may not be 3-opt relative to the original matrix. This can be checked out by restoring the cost of the denied link to its original value, and by then applying the appropriate tests. The above method provides a very fast procedure for generating a large number of distinct solutions close to the optimum. In fact for the 57 city problem a solution with cost 12978 was found, which is neither globally optimum nor 3-opt, but is the third best solution known. Thus, this procedure may prove of value in engineering applications where non-analytical constraints may be present.

LINEAR 2-OPTING

Dynamic 3-opt, while two to three times faster than Lin's 3-opt, still requires computation time which is asymptotically cubic in the number of cities. If dramatic reduction in the time required to find a local optimum is to be made, means must be found for reducing the asymptotic dependence of time on the number of cities. Such a method will now be described for finding 2-opt tours. By using the concept of local improvement and by ordering the distance matrix, there results a method for 2-opting that takes time asymptotically proportional to n.

Consider a tour with node v_1 connected to branches with lengths a and b. A local improvement is said to exist at v_1 if there exists a link not in the tour terminating at v_1 with length c and if $c<a$ or $c<b$. It can be shown easily that if a 2-opt improvement exists then a local improvement exists. Furthermore, the local improvement determines a favorable 2-change. If now for each node v_1 the other nodes are listed in order of increasing distance from v_1, we need only search down the list for a local improvement until we have passed the two nodes adjacent to v_1 on the tour. Thus, if the nodes adjacent to v_1 are each marked on the list by a star, we need only search down to the second star. In a problem with a Euclidean metric, it is plausible that the average depth of the second star for a good tour is independent of the number of cities, since it depends only on the local situation at a given city. Indeed, it has been verified experimentally for the class of problems considered here that the average second star depth is 4 to 5. This implies then that a complete checkout for a 2-opt tour requires a time linear in the number of cities, since the search for a local improvement at each city is roughly constant.

The algorithm suggested above requires a starting tour with small average second star depth, for otherwise the initial search for local improvements would still take quadratic time. The following randomized start was used: The cities are ordered randomly; each city, proceeding in the random ordering, is then connected to its nearest neighbor, with the provisions that no city is connected to more than two others, and that no closed cycle is formed until the last step.

The linear 2-opt algorithm with randomized nearest neighbor start was implemented and compared with Lin's 2-opt algorithm with random start, with the results shown in Table 2. Graph 1 shows a plot of computation time vs. problem size, and verifies the linear dependence of the new 2-opt procedure.

Problem (# of cities)	Linear 2-opt average time in seconds	Linear 2-opt average % error	Lin's 2-opt average time in seconds	Lin's 2-opt average % error
10	.015	1.9	.013	1.6
20	.023	11.1	.038	11.8
25	.028	2.8	.060	3.6
29 ([13])	.035	2.3	.085	3.0
33	.028	2.6	.12	4.6
42	.038	4.5	.19	7.7
48 ([1])	.040	4.3	.25	6.8
48 ([1], [14])	.046	14.9	.12	30.1
57	.052	5.3	.39	6.2

Table 2. A comparison of Linear 2-opt and Lin's 2-opt.

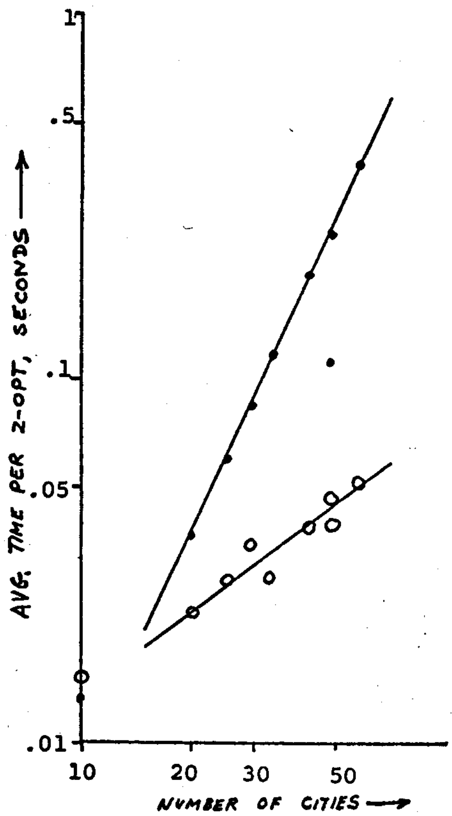

Fig. 1. A comparison of running times for Linear and Lin's 2-opt.

The approach described above has been extended to obtain a local optimization procedure almost as powerful as 3-opt which still takes only linear computation time. Experimental results will be reported in a future paper.

CONCLUSION

We have described two techniques for improving Lin's method for finding locally optimal solutions to the traveling salesman problem. The first, called dynamic 3-opt, is inductive in nature and at each stage adds one city to a 3-opt subproblem and checks only those triples of links which it is necessary to check. The second technique, called linear 2-opt, takes advantage of distance ordering in the search for 2-changes and requires computation time linear in the number of cities, as opposed to the quadratic dependence exhibited by the basic 2-opt algorithm. This approach can be extended to 3-opt and increases considerably the size of problem which can practicably be solved by existing computers.

REFERENCES

1. Held, M. and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems", SIAM, vol. 10, pp. 196-210 (1962).

2. Bellman, R., "Dynamic Programming Treatment of the Traveling Saleman Problem", J. ACM, vol. 9, pp. 61-63 (1962).

3. Little, J.D.C., K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem", Opns. Res., vol. 11, pp. 979-989 (1963).

4. Agin, N., "Optimum Seeking with Branch and Bound", Management Sci., vol. 13, no. 4 (December 1966).

5. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "Solution of a Large Scale Traveling Salesman Problem", Opns. Res., vol. 2, pp. 393-410 (1954).

6. ________, "On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem", Opns. Res., vol. 7, pp. 58-66 (1959).

7. Croes, G. A., "A Method for Solving Traveling Salesman Problems", Opns. Res., vol. 6, pp. 791-812 (1958).

8. Karg, L. L., and G. L. Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems", Management Sci., vol. 10, pp. 225-248 (1964).

9. Roberts, S. M., and B. Flores, "An Engineering Approach to the Traveling Salesman Problem", Management Sci., vol. 13, pp. 269-288 (1966).

10. Reiter, S. and G. Sherman, "Discrete Optimizing", SIAM, vol. 13, pp. 864-889 (1965).

11. Lin, S., "Computer Solution of the Traveling Salesman Problem", BSTJ, vol. 44, pp. 2245-2269 (1965).

12. Barachet, L. L., "Graphic Solution of the Traveling Salesman Problem", Opns. Res., vol. 5, pp. 841-845 (1957).

13. Lin, S., private communication.

14. Knight's Tour.