A Man-Machine Approach Toward Solving the Traveling Salesman Problem

Patrick Krolak and Wayne Felts
Vanderbilt University, Nashville, Tennessee
and
George Marble
Washington University, St. Louis, Missouri

The traveling salesman problem belongs to an important class of scheduling and routing problems. It is also a subproblem in solving others, such as the warehouse distribution problem. It has been attacked by many mathematical methods with but meager success. Only for special forms of the problem or for problems with a moderate number of points can it be solved exactly, even if very large amounts of computer time are used. Heuristic procedures have been proposed and tested with only slightly better results. This paper describes a computer aided heuristic technique which uses only a modest amount of computer time in real-time to solve large (100–200) point problems. This technique takes advantage of both the computer's and the human's problem-solving abilities. The computer is not asked to solve the problem in a brute force way as in many of today's heuristics, but it is asked to organize the data for the human so that the human can solve the problem easily.

The technique used in this paper seems to point to new directions in the field of man-machine interaction and in the field of artificial intelligence.

Key Words and Phrases: heuristic procedures, computer-aided heuristic technique, man-machine interaction, artificial intelligence, assignment problem, mask of the assignment, rubber band tour generator, interaction process, traveling salesman problem

CR Categories: 3.57, 3.66, 5.30

Introduction

The traveling salesman problem is easily stated in the following manner. Suppose a salesman is to visit n cities (nodes, customers, etc.), visiting each only once, how does he schedule his itinerary so that he travels only a minimum distance (or pays a minimum travel expense)? For the purposes of this paper, distances between cities are assumed to be the same in either direction.

This problem, which is related to numerous other problems such as board wiring, scheduling, and routing, has been the subject of intensive research for many years. To date, the efforts have been rather disappointing. No exact algorithm has ever claimed to solve problems of over 65 cities, and most heuristic algorithms require large amounts of computer time to provide satisfactory solutions to problems for that size or larger. In this paper the authors report on a man-machine approach that provides reasonable performance on very large problems (200 cities).

Careful studies have been made of currently used heuristics, of existing theory on the properties of the optimal solution, and of methods a man may use to solve this problem without either of the previously mentioned tools. The results indicate that giving a man a map of the cities he is to visit does not provide him with sufficient information to find reasonable solutions when the number of cities is large. On the other hand, giving a computer a distance matrix and generating a large number of random starting tours which are cleaned up by various heuristics which make local improvements produces excessively long computer run times before a satisfactory tour is produced and/or accepted. Clearly then, any successful approach to producing a tour whose distance is the minimum mileage, or very close to it, at

The research for this paper was supported in part by NSF grant GK-4975 L/C.

an acceptable cost in computer time and man power will have to be some combination of the above.

A division of labor seems best. The computer organizes the data and performs the numerous calculations. The human uses his eyes and problem-solving skills to create solutions and to direct the attempts at improvement. A key step in our process is the initial one of organizing the data so that a human can see the important features of the problem and integrate these pieces of information into a complete solution. Another feature of our approach is to have the computer provide the human with a number of alternative solutions so that he can see which regions the computer has probably correctly connected and which ones still need work.

Theory

In approaching the traveling salesman problem, previous authors have taken two distinct tacks. One group tried to create an exact algorithm through a branch-and-bound type of solution [1, 2]. Such an algorithm generally made heavy use of the fact that the traveling salesman problem is closely connected to another problem called the assignment problem. The other group tried to solve the problem by creating an initial tour by some simple rule (random, nearest neighbor, etc.) and then improving this tour by various methods [3, 4]. This generally involves improving a small section of the tour at a time. Unfortunately, if the initial tour is a very bad one, this local improvement will still provide only a slightly better result even after a great deal of effort. For small problems one can generate a large number of different initial tours, improve them, and use the best result with fair confidence that it is an optimal or at least a near optimal tour. For large problems this approach requires several hours of computer time, and the resulting cost often exceeds the worth of the study. One recent man-machine study [5] to which the authors' attention was called while this paper was in preparation attempted to speed up this heuristic process by turning the control of these local improvements over to the human. Our work attempts to go beyond that effort by bringing the human in much earlier and by giving him tools that expand his problem-solving ability. To do this it will be necessary: (1) to help focus the human's attention on those items which will probably be part of any near optimal tour; (2) to indicate regional and global features of any specific problem that might play a part in the development of any good tour; and (3) to provide alternative points of view so that the human can see areas for improvement in his solution and so that he can form an estimate of when he has reached the point of diminishing returns in his efforts.

Step 1. Organizing the Data To Suggest a Tour

The problem with simply giving someone a map marked with the cities he is supposed to connect is that in most cases his eyes will have difficulty in perceiving any order or pattern. The authors have found that the following technique is extremely helpful in solving this problem.

Suppose that you are a salesman who must visit a number of cities which are quite far apart and that in each city you have two or more customers who are fairly close together. In general, the time you spend in planning your tour yields the greatest savings in mileage reduction if you correctly decide on the order in which you visit the cities rather than spend the same time on deciding how to visit your customers within each city. This observation leads us to conclude that we would like to find the cluster centers or regions around which a high density of customers is located. How can we do this efficiently?

The Assignment Problem. Associated with every traveling salesman problem is another problem called the assignment problem. Let us denote the distance between city i and city j as d_{ij}. Then the associated assignment problem can be stated as

$$\min Z = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} ,$$

with

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, \cdots, n,$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1, \cdots, n,$$

where $x_{ij} = 0$ or 1. Here we interpret $x_{ij} = 1$ as meaning that we travel from city i to city j. Since it is necessary to forbid travel between a city and itself, we let $d_{ii} = \infty$. Although any feasible traveling salesman tour is necessarily a solution to the associated assignment problem, the optimal assignment is not necessarily a feasible solution to the traveling salesman problem. In fact, the optimal assignment is generally a collection of disjoint subtours frequently of length 2. This result leaves branch-and-bound algorithms very little to work with and hence leads to their poor performance in large problems [6]. However, Shapiro [2] notes that there is a high correspondence between the optimal traveling salesman's tour and the associated assignment solution. That is, it is not uncommon for the vast majority of assignment subtours to be the links of the traveling salesman's tour. Thus the assignment solution indicates local order.

Assignment of the Assignment. If we think of the subtours found in our solution of the assignment in our previous example as being the visits to customers within one city, we can now see how to proceed. Let us find the geometric means of all the subtours of the optimal assignment—that is, find the center of gravity of the cities on the subtour. These geometric means are now the location of the "city," and if we calculate the distance matrix for these "cities" and solve the resulting assignment problem referred to as the second level assignment or the assignment of the assignment, we find how the "cities" are regionally clustered. Carrying this operation to higher levels we get fewer and fewer "cities" in each level. At each level the resulting subtours locate, like a map, first the cities, then county seats, then state capitals, etc. This information next tells us which regions are likely to be connected to which regions. The correspondence between the observed traveling salesman tour and the second level assignment is also high, as is the third level, but reliability of the prediction falls as we go higher in level. The information is then presented to the problem solver as in Figure 1.

Mask of the Assignment. The assignment subtours have been observed to make up a large part of the traveling salesman optimal tour [2]. We will call these subtours primary links. The authors have found that solving another assignment problem gives many of the links that connect these subtours. This assignment problem is referred to as the mask of the assignment and is defined as similar to the assignment with the exception that all the $d_{ij} = \infty$ for those $x_{ij} = 1$ in the optimal solution of the original assignment. In effect, we are forbidding the assignment subtours in order to find the next best collection of subtours. The optimal subtours of the mask of the assignment are referred to as the secondary links.

The mask of the assignment is also drawn up (Figure 2) and, together with the assignment of the assignment's results, forms the input to our problem-solver.

Step 2. Human Generated Tour

At the end of Step 1, the problem-solver has before him data that has been found to be a major portion of other optimal traveling salesman tours. It is now up to him to integrate all of the data into a tour. The human problem-solver has one advantage that no computer has in that he can "see" the whole problem. This is not merely a comment on man's visual capabilities; rather it is a recognition of his ability to conceptualize. Thus a human can envision a solution which accounts for interactions between all of the regions that must be connected and can avoid the pitfalls of the typical nearest neighbor approaches.

The problem-solver then draws the tour and compares how well it uses the primary and secondary links of the assignment problem and its mask, and how well it connects regions grouped by the second and higher level assignment problems, etc. In general, the problem-solver will find that a good tour has a number of features: (1) it seldom has rapidly oscillating sawtooth curves; (2) it is smoother—more like a polygonal approximation to a curve, particularly in areas with a high density of points; (3) it appears to have a high ratio of enclosed area to perimeter. A tour with long narrow necks which wind back on themselves is to be avoided.

If two or more problem-solvers use this information to generate independent trial tours, it is worthwhile to compare the solutions. The comparison generally results in a composite tour that requires much less computer time to reduce the tour to final form. The cost of having two or more people draw a tour is relatively small since the data has already been prepared, and even with a 200-city problem a tour can be generated in half an hour.

The generation of a tour provides two benefits that should not be overlooked. The first is that by going through the thought process the problem-solver will become familiar with the features of the problem. In the event that the model and the real problem do not completely correspond, this familiarity should allow the problem-solver to adjust the model results to include these difficulties in a more intelligent manner. Since this frequently occurs in applications, this benefit should not be minimized as to its importance. Secondly, in generating a tour and making the comparison and evaluating the results, the problem-solver develops a feeling for places where his solution is weak and where it is in need of only a little work. This experience determines which of several heuristics will be used in the cleanup phase and where they will be applied. For some applications the process could very well stop here since frequently only a reduction of several percent in total length results from further effort.

Step 3. Computer Generated Initial Tours

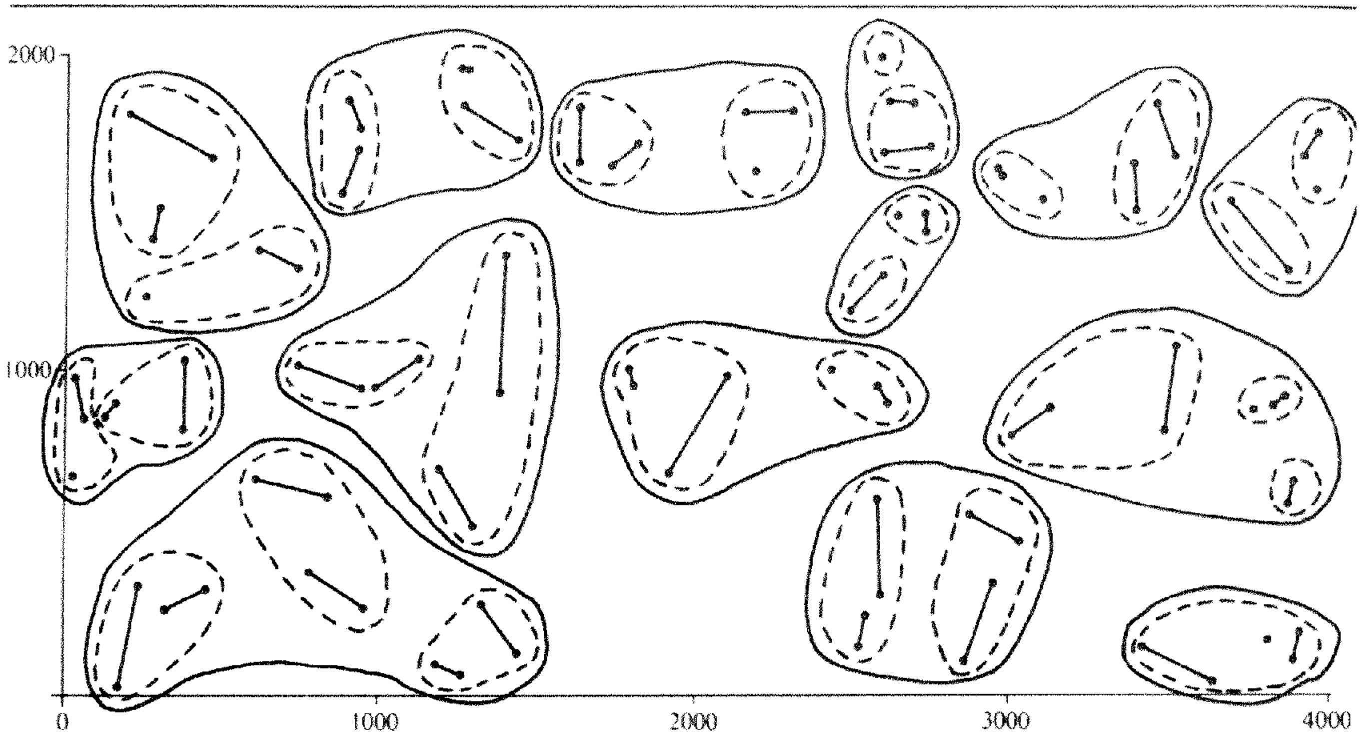
For the sake of comparison or to suggest to the problem-solver other approaches to solving the problem, a collection of heuristics is made available to produce initial tours. The authors have tried to generate several heuristics that produce a near optimal tour using only a minimum amount of computer time. They succeeded in generating a number of different heuristics, the details of which form the subject of another paper [7]. The authors call one of these heuristics the "Rubber Band Tour Generator" because the manner in which it generates a tour is similar to stretching a rubber band around the regional assignment data. These heuristics, however, have been found to produce tours that are very nearly optimal in computer times that are relatively short (1–2 minutes of Sigma VII time on a 100-city tour) and do not grow rapidly with an increase in problem size. Besides being fast and accurate the heuristic makes use of the data generated by Step 1.

Step 4. Tour Review

The problem-solver now has several proposed tours which are evaluated and compared with one another. The data from Step 1 is compared with each of the tours, and a decision is made whether to work with one of the proposed tours or a composite of the tours. Generally, after some experience the problem-solver is able to recognize some superior features or ideas in each of the tours generated in Steps 2 and 3 and finds it to his advantage to begin work with a composite.

At this point the reviewer is concerned with two types of problems. One type concerns the difficulties that arise because the eye is not able to resolve small trade-offs. These questionable areas are resolved by applications of point replacement, by the interchange of two or more links with another set, or by other routines that are commonly used in traveling salesman heuristics.

Fig. 1. Display of the first through third level assignment problems.



In general, the result of this kind of improvement is a very small reduction in total length. The other type of problem, much more difficult to see and to handle, occurs when the connection of one region to another has been made incorrectly. Locating regions where a major segment of the tour is incorrectly drawn can be done in a number of ways. The intersection set of the primary and secondary links and the current tour often shows many regions completely connected while other regions are very disjoint. These disjoint regions are often the areas that require major changes to be made in the tour. The comparison between the computer's heuristics, such as the "Rubber Band Tour Generator," and one or more solutions drawn by individual problem-solvers reveal regions, particularly those interior regions of very large problems, which can be interconnected in a variety of ways. The set of cities found in these regions is generally moderately large, and the number of combinations of the various ways to interconnect the various regions is also fairly large. Hence the need for a regional tour improvement routine is evident.

The result of this step is to pinpoint the problem areas and decide on the order of application of regional and local heuristics.

Step 5. Final Improvements in the Tour

The problem-solver now applies the local and re gional improvement routines to the proposed tour. Th local routines are simple routines such as the poir routine which removes a given point from the tour an inserts it between two adjacent cities if it finds that thi reduces the total distance. A heuristic such as describe by Linn [4] for the replacement or interchange of two c more links is also available. The regional improvemer routines are really traveling salesman optimization prc grams which have been shown to be able to solv problems of moderate sizes reliably and rapidly. Th problem-solver determines that a region of the probler has not been properly handled. The only problem abou using a traveling salesman optimization program is tha the final result must match up with the remainder of th tour. This match-up can be handled by replacing th true distances by a zero distance wherever the tour is cut This forces the resulting regional tour to include arti ficial links which will match up with the remainder of th tour. The regional routine has the sections of the tou being clipped off the original tour and the location o zero distance links as inputs. The regional improvemen routine currently being used by the authors is based oi

Fig. 2. Mask of the assignment problem.

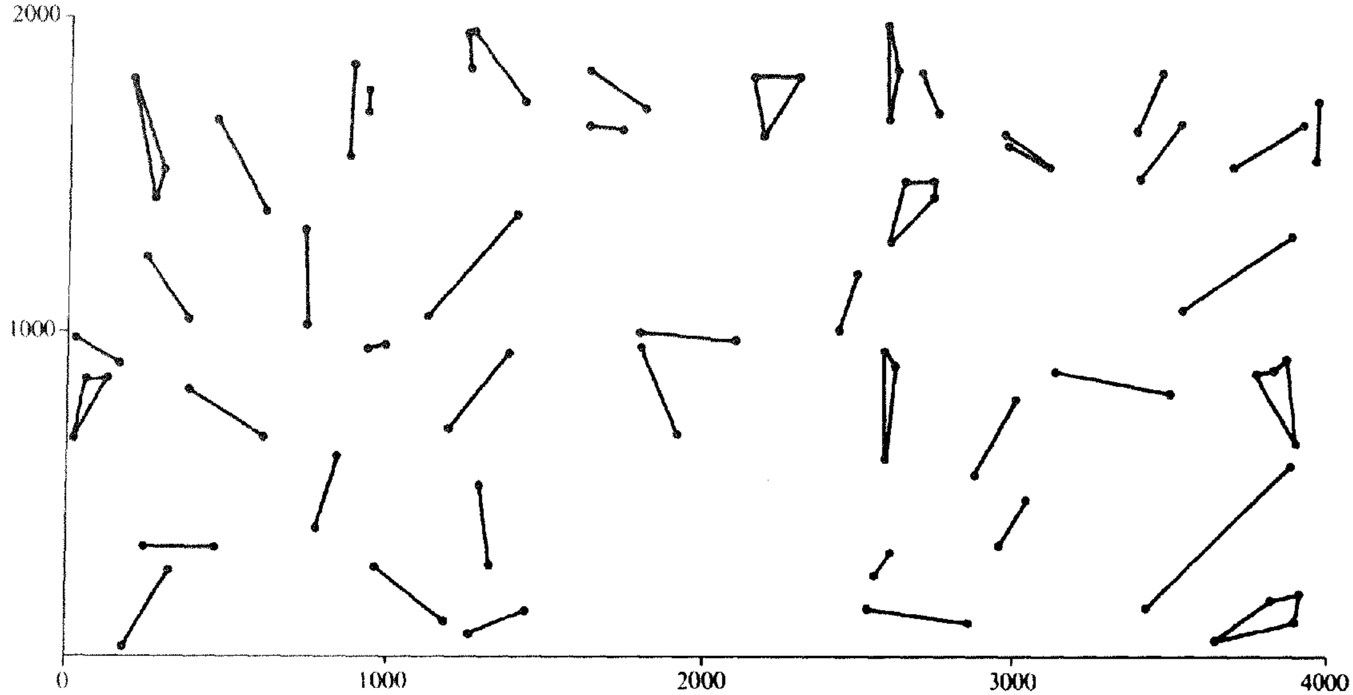


Fig. 3. Final tour = 21282 units, 5.7 minutes CPU time, 76 minutes terminal time.

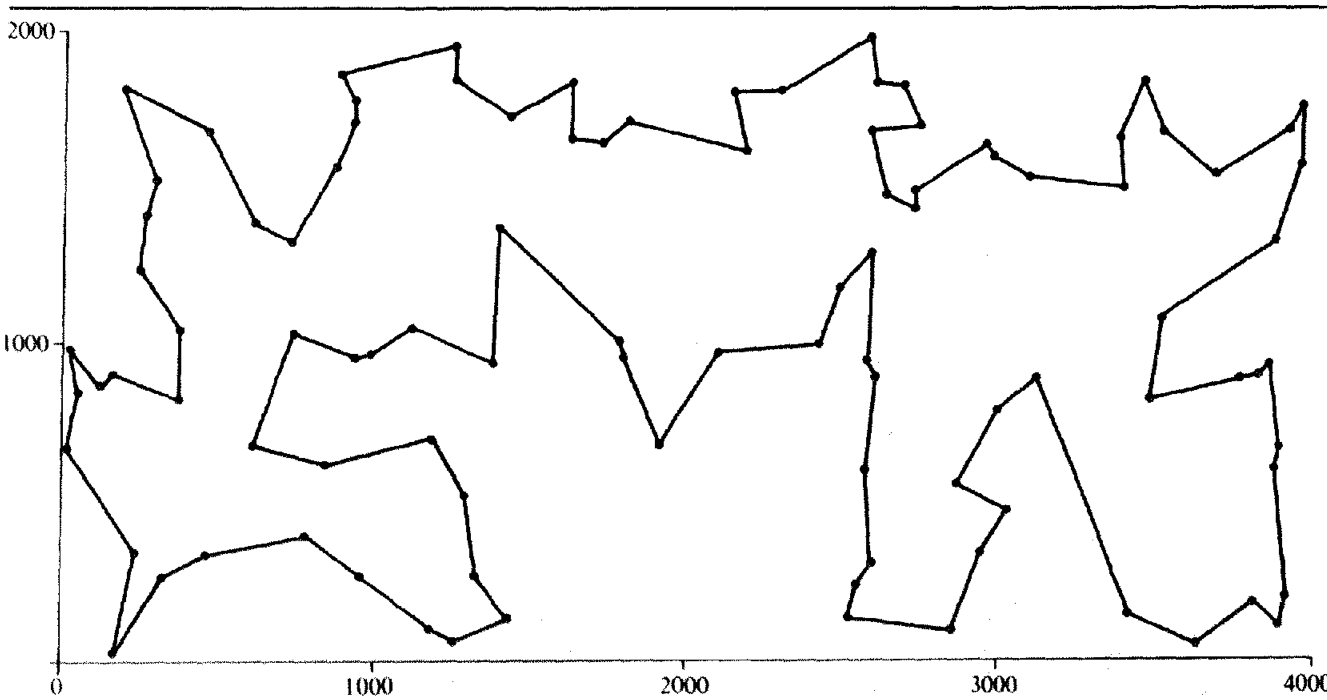


Table I. Comparative results

Number cities	Problem number	Rubber Band Time	Rubber Band Cost	Oberuc Time	Oberuc Cost	Man-Machine Time	Man-Machine Cost
100	24	1.312 min.	21282	22 min.	21473.7	4.87 (76 min.)	21282.0
100	25	1.097 min.	22523	23 min.	22549.3		22193.3
100	26	1.110 min.	21536	11 min.	21052.5		20852.3
100	27	1.373 min.	21410	22 min.	21527.8		21294.3
100	28	1.048 min.	22794	19 min.	22119.0		22115.6
150	30	2.681 min.	27419	37 min.	26794.0	3.06 (75 min.)	26761.2
150	31	2.750 min.	26509	43 min.	26320.0		26216.4
200	32	5.198 min.	30409	150 min.	29563.1		29823.1
200	33	4.612 min.	29966	150 min.	31424.5	8.83 (70 min.)	29678.2

Table II. Coordinates of the Example
Problem 24 (100 cities, Cartesian coordinates, randomly generated)

City Number	X Coord	Y Coord	City Number	X Coord	Y Coord
1	1380	939	51	2482	1183
2	2848	96	52	3854	923
3	3510	1671	53	376	825
4	457	334	54	2519	135
5	3888	666	55	2945	1622
6	984	965	56	953	268
7	2721	1482	57	2628	1479
8	1286	525	58	2097	981
9	2716	1432	59	890	1846
10	738	1325	60	2139	1806
11	1251	1832	61	2421	1007
12	2728	1698	62	2290	1810
13	3815	169	63	1115	1052
14	3683	1533	64	2588	302
15	1247	1945	65	327	265
16	123	862	66	241	341
17	1234	1946	67	1917	687
18	252	1240	68	2991	792
19	611	673	69	2573	599
20	2576	1676	70	19	674
21	928	1700	71	3911	1673
22	53	857	72	872	1559
23	1807	1711	73	2863	558
24	274	1420	74	929	1766
25	2574	946	75	839	620
26	178	24	76	3893	102
27	2678	1825	77	2178	1619
28	1795	962	78	3822	899
29	3384	1498	79	378	1048
30	3520	1079	80	1178	100
31	1256	61	81	2599	901
32	1424	1728	82	3416	143
33	3913	192	83	2961	1605
34	3085	1528	84	611	1384
35	2573	1969	85	3113	885
36	463	1670	86	2597	1830
37	3875	598	87	2586	1286
38	298	1513	88	161	906
39	3479	821	89	1429	134
40	2542	236	90	742	1025
41	3955	1743	91	1625	1651
42	1323	280	92	1187	706
43	3447	1830	93	1787	1009
44	2936	337	94	22	987
45	1621	1830	95	3640	43
46	3373	1646	96	3756	882
47	1393	1368	97	776	392
48	3874	1318	98	1724	1642
49	938	955	99	198	1810
50	3022	474	100	3950	1558

the work of Oberuc [3]. This routine can handle up to 30 cities at any one time. The authors have found the performance of this routine to be satisfactory for this size problem, but it becomes less reliable for larger problems. The routine requires only a small amount of core, which is an advantage for time-sharing.

Step 6. Final Review

At this point the problem-solver has one basic decision to make—he must decide whether the potential undiscovered reduction in tour length is worth the expense of his and the computer's time. Guha [8] claims a tight bound in the optimal traveling salesman solution based on the optimal assignment solution and what he calls the exits of the subtour. The bound is simple to calculate and can be used to estimate the maximum reduction. If the problem-solver has worked with similar problems, then by the time he reaches this point the problem should have a very good, if not optimal, solution. (See Figure 3.)

Actual Man-Machine Configuration. The first step of the process requires a large amount of memory and a transportation linear programming code (a special assignment code is not available to the authors); hence it is done under batch mode on an XDS Sigma VII. Since the authors have facilities which provide a 10- to 20-minute turnaround, this is not a major bottleneck.

The results are plotted on a CalComp plotter, and clear transparencies are made up to be used by the problem-solver with a light board so that by overlaying the various results he can make comparisons rapidly. (This would be a perfect application for a computer graphics system; however, such luxuries are outside the budget of the authors. The light board and overlays are adequate for the authors' purposes and only moderately bothersome to work with.)

The third and fifth steps are done under the Sigma VII's time-sharing monitor, and the data is handled through a teletype terminal. The programs are arranged so that the problem-solver can call in whatever heuristic is necessary to improve the current tour.

Handling Large Problems. One would expect the size and speed of this man-machine process to be limited by the requirement of the assignment solution. However, the authors have found that these limitations can be avoided by the following observation: that the assignment problem tends to group neighbors together, and that if one knows the optimal solution for a given problem, one can remove a subtour without changing the rest of the solution. While we are not able to envision the optimal subtours of the assignment solution, it is fairly common to find an open region which divides the cities into two or more distinct sets. If this open region or valley is wider than the average nearest neighbor distances in these distinct sets, then one can change the original assignment problem into one of solving the assignment problem on each of the distinct sets. This can be done with little fear of suboptimizing the original problem since, if the valley is of the assumed width, it is unlikely that any optimal subtour would have cities in more than one set. The authors have had no difficulty splitting 200-city assignment problems into two 100-city ones. The results, of course, can be recombined for the next higher level assignment if the resulting number of subtours is manageable. Even if the separation turns out to split a subtour, the resulting data would still be accurate enough to use to generate a tour.

If the splitting technique is used to break the assignment problem up into manageable pieces of, say, 100 cities, the limit on the size of the problem seems to be more a function of the problem-solver's patience and eye fatigue. Based on the authors' experience it would seem to be economically and physically possible to solve 300-city to 600-city problems.

Results

To date, the results have been very encouraging. The authors have not found the problems in the current literature to be very difficult because of obvious patterns. Most of the problems were generated from a map of the United States, and most of the cities lie on the west and east coasts. Since most of the problems in the literature are under 50 cities, they are being used as training exercises. To avoid having the results tainted by a foreknowledge of the literature, one 49-city problem based on road map distances was solved using great circle distances, and a new optimal tour of a very different character was found.

To test the performance of the method on a more challenging set of problems, Oberuc's 75-city, 86-city, and 100-city problems all using great circle distances were undertaken and all of the optimal solutions were found except for the 100-city problem where the man-machine process found a better solution. The 100-city problem is of interest for cost comparison. Oberuc [3] used 80 minutes of UNIVAC 1108 time (at least three times the speed of the Sigma VII) and the man-machine method required 10 minutes of CPU time and one hour of human time. The remainder of the large problems tested were randomly generated using Cartesian coordinates. The results would seem to indicate that the "Rubber Band Tour Generator" performs adequately and that the man-machine method produces answers at reasonable expenditures of both computer time and human time (see Table I).

The times given in Table I are for the Sigma VII computer. The man-machine times do not include the computer times for the assignment problems which take 0.8 minutes for the 100-city problem and 1.8 minutes for the 200-city problems. (CPU times are not always possible to get but certainly are under 10 minutes for any problem listed.) The time in parentheses is the amount of the human's time required to carry out the process. For those human times not given, no problems were worked on for more than 90 minutes. Based on current rates, one engineering technician man-hour is equivalent to one minute of Sigma VII time.

One further type of traveling salesman problem that would seem to have practical application is the three-dimensional extension. In particular, the design of a circuit on multiple boards or the laying of ducts or pipes through a ship or a building can be formulated as a traveling salesman problem where the distance metric in the former case is

$$d_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2 + k\,|z_i - z_j|\,,$$

or in the latter case is

$$d_{ij} = |x_i - x_j| + |y_i - y_j| + |z_i - z_j|\,.$$

This additional coordinate does not normally present any mathematical difficulty to any algorithm or heuristic but is an order of magnitude more difficult for the man-machine process. However, the authors' computational experience with Oberuc's heuristic and the man-machine process on 150-point problems (30 random points on 6 boards) indicates that the man-machine process is more economical and accurate than the results of Table I.

The process has been taught to five students and they have produced results that are similar to those in Table I (errors of 0.5 percent). Although the results of the above man-machine process are not perfectly reproducible, they are surely in satisfactory range for most potential users.

For those who would like to make a comparison of their heuristics with the authors' results, the coordinates for the problem in Figures 1–3 are given in Table II.

Conclusions and Extensions

The authors believe that so far their work has shown the man-machine approach to be an accurate and economical method of solving large traveling salesman problems. The interaction process is being carefully studied to seek ways of improving the problem-solver's ability to envision the important features of the problem and to increase his imagination so that he will not overlook regions that still need to be improved. A training manual is planned, and studies about the learning process of the problem-solver should produce interesting results.

Since the traveling salesman problem is akin to many other important problems such as the bottleneck problem and the coin collector (the salesman returns home each time he visits a certain number of cities), the question arises whether these results can be extended. The authors believe the answer is a qualified yes, if the problem has certain features. These features include a means of organizing the important data to be utilized in the solution, a problem that can be represented in the form of maps, diagrams, or charts, and a means of improving the results through the use of simple heuristics.

Acknowledgments. The authors would like to thank the computer centers of Vanderbilt University, Nashville, Tennessee, and of Washington University, St. Louis, Missouri, for providing the necessary computer time for the study. Discussions with Dr. Leon Cooper, Washington University, provided the original motivation for this study, and his comments were of great help.

Received July 1970; revised October 1970

References

1. Little, M. D. C., Murty, K. G., Sweeney, D. W., and Karel, C. An algorithm for the traveling salesman problem. *Operations Res. 11* (1963), 979–989.
2. Shapiro, D. Algorithms for the solution of the optimal cost traveling salesman problem. Sc.D. Th., Washington U., St. Louis, Mo., 1966.
3. Oberuc, Richard E. A practical algorithm for finding solutions to the traveling salesman problem. 34th ORSA Nat. Meeting, Denver, Col., Nov. 1968.
4. Bellmore, M., and Newhauser, G. L. The traveling salesman problem: A survey. *Operations Res. 17*, 3 (May 1968), 538–557.
5. Michie, D., Fleming, J. G., and Oldfield, J. V. A comparison of heuristic, interactive, and unaided methods of solving a shortest-route problem. In *Machine Intelligence*, D. Michie, (Ed.), American Elsevier, New York, 1968, p. 245.
6. Bellmore, M., and Malone, J. C. Pathology of traveling salesman algorithms. Johns Hopkins U. Rep., Baltimore, Md., 1969.
7. Krolak, P., Felts, W., and Marble, G. Heuristics for solving the traveling salesman problem and various related scheduling problems. 7th Mathematical Programming Symp., The Hague, The Netherlands, Sept., 1970.
8. Guha, D. K. A method for solving the traveling salesman problem. 34th ORSA Nat. Meeting, Denver, Col., 1968.