Comput. & Ops Res., Vol. 2, pp. 115–119. Pergamon Press, 1975. Printed in Great Britain

THE CLUSTERED TRAVELING SALESMAN PROBLEM*

JAMES A. CHISMAN†
Systems Engineering Program, Clemson University

Scope and purpose—The classical "traveling salesman problem" in operations research[3] involves finding the shortest closed path that a salesman can take to travel to N different cities, visiting each city only once. Besides scheduling "salesman," this type of formulation can be extended to find optimal tours of an order picker through a warehouse to satisfy a shipping order. It is a ramification of this latter application to which this article is addressed.

Abstract—The traveling salesman problem is expanded to include the situation where a group of cities (cluster) must be visited contiguously in an optimal, unspecified order. Given several sets of clusters within the problem, a method is developed for optimizing simultaneously the ordering of cities within each cluster and the ordering of clusters. A real-world warehousing problem is shown to fit this formulation.

INTRODUCTION TO THE TRAVELING SALESMAN PROBLEM

The classical "traveling salesman problem" involves finding the shortest or least costly path (tour) that a salesman can take to visit N cities, where each city is visited once and only once and the salesman must return to his arbitrary starting position (city). Obviously this type of problem and its solution has applications in other areas, such as the (1) scheduling of production, maintenance, and pickups and deliveries, (2) location of facilities and (3) the placement of inventories.

The standard distance or cost matrix for this problem is shown in Fig. 1, where d_{ij} is the distance or cost to go from city i to city j directly and d_{ii} is meaningless.

Using Fig. 1 the traveling salesman problem can be formulated as follows:

$$\text{Minimize } Z = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}x_{ij} \text{ for } i \neq j \tag{1}$$

subject to

$$\sum_{j=1}^{N} x_{ij} = 1 \text{ for } i = 1, 2, \ldots, N;\ i \neq j \text{ (one exit/city)} \tag{2}$$

$$\sum_{i=1}^{N} x_{ij} = 1 \text{ for } j = 1, 2, \ldots, N;\ i \neq j \text{ (one arrival/city)} \tag{3}$$

$$x_{ij} = \begin{Bmatrix} 1, \text{ if one goes from city } i \text{ to city } j \\ 0, \text{ otherwise} \end{Bmatrix} \text{ for all } i \text{ and } j \tag{4}$$

A complete tour must be made of all cities and no subtours can be allowed. (5)

Formulations (1)–(4) above are a statement of the classical "assignment problem," (see Wagner[3]) which, in general, will allow subtours, will not guarantee a complete tour and can be solved by linear programming algorithms. The traveling salesman problem requires formulations (1)–(5). Restriction (5) complicates the solution considerably and disallows the use of linear programming. Two main types of solution procedures are available: "integer linear programming" and "branch and bound."

*This work was supported by the J. P. Stevens & Co., Inc., a major textile manufacturer.

†James A. Chisman is the director of the Engineering Technology Program at Clemson University. He holds the B.S. in Electrical Engineering from Akron University; and the M.S. in Industrial Engineering and the Ph.D. in Management Engineering from Iowa University. He has published articles in the *Journal of Industrial Engineering, The Journal of the Environmental Engineering Division of the A.S.C.E.* and several other journals. His primary field of research has been simulation and mathematical programming applied to environmental problems and to production planning and control.

$$
\text{From}\;
\begin{array}{c|cccccc|}
 & \multicolumn{6}{c}{\text{To}} \\
 & 1 & 2 & 3 & \cdot\;\cdot\;\cdot & & N \\
\hline
1 & & d_{12} & d_{13} & \cdot\;\cdot\;\cdot & & d_{1N} \\
2 & d_{21} & & d_{23} & \cdot\;\cdot\;\cdot & & d_{2N} \\
3 & d_{31} & d_{32} & & \cdot\;\cdot\;\cdot & & d_{3N} \\
\cdot & \cdot & \cdot & & \cdot\;\cdot\;\cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\
N & d_{N1} & d_{N2} & d_{N3} & \cdot\;\cdot\;\cdot & & \\
\hline
\end{array}
$$

Fig. 1.

If Fig. 1 is symmetric (i.e., $d_{ij} = d_{ji}$), Dantzig *et al.*[1, 2] show how an integer linear programming formulation of the traveling salesman problem can be used to arrive at a solution. For the symmetric or asymmetric (general) case, Wagner[3] demonstrates an integer linear programming formulation and solution. Unfortunately, these approaches are practical for problems with only 10 or fewer cities because of the very large number of restriction equations that are required to be written.

The most practical approach to solving traveling salesmen problems with more than 10 cities is to use one of the branch and bound algorithms that are available. For the symmetric case, Held[4] and Lin[5] have developed fairly efficient algorithms. For the general case, Eastman[6] and Little[7] have developed usable algorithms. Using branch and bound algorithms, not only can an optimal solution be found, but if computer time is a limiting factor, a run can be terminated early with a "near optimal" solution. If more than 80 cities are involved, exact solutions, using the above methods, are usually too costly to find. In this case, it will be necessary to use a fast, approximating algorithm, such as Christofides[8] and Webb[9].

If it is necessary for the salesman, due to some load restriction, to make more than one trip from his home base to a subset of cities and return before he can complete the tour of all cities or if there is more than one salesman, each visiting a subset of the cities, then this problem becomes a "dispatching" or "delivery" problem. The routing can be "optimized" by using one of several available techniques, such as Christofides[10] and Tillman[11].

THE CLUSTERED TRAVELING SALESMAN PROBLEM

While studying a real-world warehousing problem for J. P. Stevens & Co., Inc. the author found a class of problems, which he shall coin the "clustered traveling salesman problem," for which there appears to be no description or solution in the literature. This problem can be stated as follows: In an N city traveling salesman problem, the salesman must not only visit each city once and only once, but a subset (cluster) of these cities must be visited contiguously. For example, in a particular problem, cities i, j, and k must be visited contiguously in one of the following orders $i-j-k$, $i-k-j$, $j-i-k$, $j-k-i$, $k-i-j$, or $k-j-i$. There may be more than one cluster within a given problem.

The order of the visits within each cluster is not specified; if it were, the problem would simply break down into finding the optimal route among the clusters. For the general clustered problem, however, it is necessary simultaneously to find the optimal order within each cluster and the optimal order among clusters, which are not independent problems. (A city not in a cluster of two or more cities can be looked on as a special limiting case of a cluster with only one city.)

The formulation of this problem is the same as for the traveling salesman problem, (1)–(5), with the additional restriction that follows:

The path through a cluster must be contiguous and only link with other clusters at the beginning and end of the path. (6)

Figure 2 shows how the distance or cost matrix of Fig. 1 can now be partitioned to identify the various clusters, C_i, and the various submatrices, L_{iq}, which links C_i with C_q.

The particular example which gave rise to this problem is stated as follows: In a warehousing system, an order for goods will arrive which will contain several suborders (bills-of-lading), each of which will call for several different stock numbers (SN). A motorized truck will be dispatched through the warehouse to pickup the SN's for each bill-of-lading (BL). The restriction is that a BL must be completely satisfied before the next BL is started. The order of picking SN's within each BL and the ordering of the BL's are to be simultaneously optimized. (It can be seen here that the location of each SN is analogous to a city and each BL is a cluster of SN's.)

A simple example of this warehouse dispatching problem is shown in Fig. 3. Position 1 represents the shipping dock (SD) from which the truck starts and to which it returns. BL1 is composed of SN's 2 and 3 and BL2 is composed of SN's 4, 5 and 6. Hence, Items 2 and 3 must be picked contiguously in some order and Items 4, 5, and 6 must likewise be picked contiguously in some order. The problem is to find the shortest path from the SD through the warehouse and return to the SD, considering the above restrictions. (The matrix is asymmetric since the truck can only travel one way in most aisles of the warehouse.) Multiple locations for the same SN are not considered. The quantity of each SN picked up at a given location is not required to be known since this does not affect distance traveled; of course, this implies that the truck is large enough to hold all items desired and there is an adequate supply at each location.

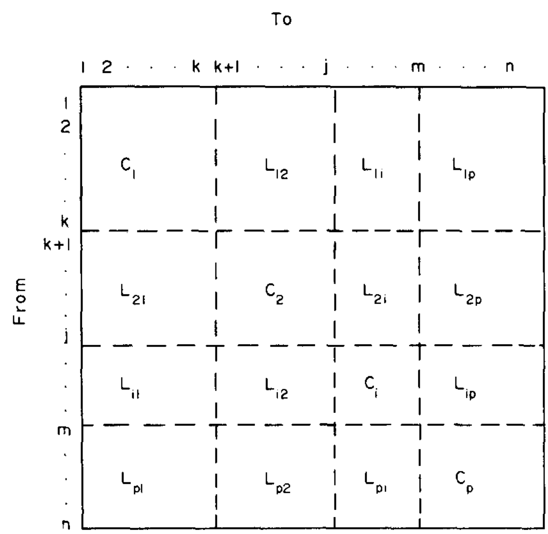

Fig. 2.

		SD	BL1		BL2		
	SN	1	2	3	4	5	6
SD	1	C_1	30	10	40	20	50
BL1	2	20	C_2	40	10	50	30
	3	50	20		30	10	40
BL2	4	30	40	50		60	10
	5	40	10	50	20	C_3	30
	6	10	50	20	30	40	

Fig. 3.

METHOD OF SOLUTION

To solve the general, clustered traveling salesman's problem of Fig. 2. The author developed the following novel approach:

(1) From all elements in each clustered partition, C_i, subtract a large number K ($K \gg \max d_{ij}$ for all i and j, say $K = 10[\max d_{ij}]$). Do not change the elements in the linkage partitions, L_{iq}.

(2) Using one of the branch and bound algorithms [4–7, or others], solve this revised problem.

The tour which is found using this method is optimal for the clustered traveling salesman problem. Once this optimal tour is found, the minimum distance or cost is found by evaluating this tour for the original matrix.

Without going into a lot of mathematical jargon, the reason that this method produces an optimal, clustered tour is as follows:

(1) By subtracting a large number, K, from each element in each cluster, the relative attractiveness is enhanced for making links within a cluster, C_i, rather than between two clusters using an L_{iq} element.

(2) Since each element within a cluster has had the same value, K, subtracted from it, no effect has been made on the relative attractiveness among these various elements (cities) of a cluster.

(3) Since eventually linkage among clusters must be made and since the relative attractiveness of cluster linkage partitions was not changed (K was *not* subtracted from each element in any L_{iq}), then optimal linkage among clusters will be forced.

To partially demonstrate the method's application, Fig. 4 shows the revised setup for the example shown in Fig. 3. ($K = 100$ was subtracted from each element in cluster C_2 and C_3; the only element in C_1 is d_{11} which does not exist.) Applying a branch and bound algorithm to the problem of Fig. 4 produces the optimal tour of 1-5-4-6-3-2-1 which, when priced according to Fig. 3, has a minimum $Z = 110$. (Incidentally, the optimal non-clustered traveling salesman solution is 1-3-5-2-4-6-1 with minimum $Z = 60$.)

For small problems (say less than 10 "cities"), the integer linear programming formulation of Dantzig *et al.* [1, 2] or Wagner [3] can be augmented by adding the following restriction for each cluster, C_k ($k = 1, 2, \ldots, p$) in the problem:

$$\sum_{i,j \to C_k} \sum x_{ij} = S_k - 1$$

where $i,j \to C_k$ indicates that all the x_{ij}'s in cluster partition C_k are to be summed, and S_k is the size or order of partition C_k (i.e., the number of rows in C_k).

For the example problem of Fig. 3, the following restrictions would have to be added:

$$x_{23} + x_{32} = 1 \quad \text{(for cluster } C_2\text{)}$$

$$x_{45} + x_{46} + x_{54} + x_{56} + x_{64} + x_{65} = 2 \quad \text{(for cluster } C_3\text{)}$$

(Note: There is no need to alter the distance or cost matrix if the above integer linear programming formulation is used.)

		SD	BL1		BL2		
	SN	1	2	3	4	5	6
SD	1	C_1'	30	10	40	20	50
BL1	2	20	C_2'	-60	10	50	30
	3	50	-80		30	10	40
BL2	4	30	40	50		-40	-90
	5	40	10	50	-80	C_3'	-70
	6	10	50	20	-70	-60	

Fig. 4.

CONCLUSIONS

The method presented in this article for solving the "clustered" traveling salesman problem was tried on several problems of various sizes and configurations for which the optimal solutions were known. The optimal solutions were found with no significant difference in the computer running times compared to those for the equivalent nonclustered problems, as would be expected.

An interesting ramification of the real-world type of problem that this method can handle is the case where several BL's within an order call for some similar SN's. This is easily handled by treating each SN as a separate location, as before; and in the distance or cost table making $x_{rs} = 0$, where r and s are the same SN within 2 different BL's. This will obviously tend to make r and s beginning or ending SN's on one cluster's (BL's) path so as to link up with similar SN's appearing at the end or beginning of an adjacent cluster's (BL's) path, whenever possible. This tendency was found to be the case in the problems solved.

Although multiple locations for each SN was not considered pertinent for this study, it is logical to assume that this could exist in another application. Also, for this study, it was assumed that each location had an adequate quantity of a particular SN to satisfy that order's needs. An extension of the problem presented in this article to include multiple, limited supply locations for a particular SN seriously complicates this problem and implies that a "transportation problem" (see Wagner [3]) is imbedded in the clustered traveling salesman problem. More work needs to be done to accomplish this goal.

REFERENCES

1. G. Dantzig, R. Fulkerson and S. Johnson, Solution of a large-scale traveling-salesman problem, *Ops Res.* **2**, 393–410 (1954).
2. G. Dantzig, R. Fulkerson and S. Johnson, On a linear-programming combinatorial approach to the traveling-salesman problem, *Ops Res.* 7(1), 58–66 (1959).
3. H. M. Wagner, *Principles of Operations Research with Applications to Managerial Decisions.* Prentice-Hall (1969).
4. M. Held, The traveling-salesman problem and minimum spanning trees: part II, *Mathematical Programming (Netherlands)* **1**(1), 6–25 (1971).
5. S. Lin and B. W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem, *Ops Res.* **21**(2), 498–515 (1973).
6. W. L. Eastman, *Linear Programming with Pattern Constraints*, Ph.D. Dissertation, Harvard University, July (1958).
7. J. D. C. Little, K. G. Murty, D. W. Sweeney and C. Karel, An algorithm for the traveling salesman problem, *Ops Res.* **11**(2), 972–989 (1963).
8. N. Christofides and S. Eilon, Algorithms for large-scale traveling salesman problems, *Opl Res. Q.* **23**(4), 511–518 (1972).
9. M. H. J. Webb, Some methods of producing approximate solutions to traveling salesman problems with hundreds or thousands of cities, *Opl Res. Q.* **22**(1), 49–66 (1971).
10. N. Christofides and S. Eilon, An algorithm for the vehicle-dispatching problem, *Opl Res. Q.* **20**(3), 309–318 (1969).
11. F. A. Tillman and T. M. Cain, An upperbound algorithm for the single and multiple terminal delivery problem, *Mgmt Sci.* **18**(11), 664–682 (1972).

(*Paper received* 3 *February* 1975)