Technical Note

Bounds for the Travelling-Salesman Problem

Nicos Christofides

Imperial College of Science and Technology, London, England

(Received March 25, 1971)

This paper concerns finding a tight lower bound to the travelling-salesman problem, with the hope that all the different branch-and-bound algorithms for this problem can benefit from it. The bound is calculated by an iterative procedure with guaranteed convergence and is shown to require a computation time only about 9 per cent greater than the time required to solve an equivalent assignment problem. This new bound was tested on 14 sample problems and, on the average, found to be only 4.7 per cent below the optimum for symmetrical, and 3.8 per cent below the optimum for asymmetrical problems.

LET US SUPPOSE that there are n cities and the distances (costs) between these cities are known. A travelling salesman starts from a city, visits each and every one of the remaining cities just once, and returns to the original starting city. The problem is to find the tour that involves covering the smallest distance. This well known travelling-salesman problem (TSP) is a combinatorial one; for the general case of an asymmetrical distance matrix, there exists a total of $(n-1)!$ tours.

There are a number of algorithms[5,9,10,11] in the literature, based on a variety of principles, that can solve the TSP without complete enumeration and evaluation of all the possible tours. However, even for the best of these algorithms, the computation times increase exponentially with the number of cities, and hence even moderately sized problems (more than a few tens of cities) cannot be solved optimally. Of these methods the most important are the ones belonging to the general class of branch-and-bound algorithms; they are based on a tree search in which, at each stage the problem is partitioned into two or more subproblems, represented by nodes in a decision tree, and lower bounds are calculated on the cost of the optimal tour contained in each subproblem. The partition of the problem into subproblems is chosen so that these subproblems are disjoint in the sense that every one of the total number of tours in the problem appears as a tour in one and only one of the subproblems. Successive partitions of the original problem will eventually yield a small enough problem whose solution is obvious. Once an initial solution is obtained, the lower bounds calculated at the various nodes are used to eliminate from further consideration whole parts of the tree that would otherwise have to be investigated. Thus, if the lower bound on a node is larger than the best solution already obtained, then no more branching from that node is necessary, since the existing solution can obviously not be improved.

From what has already been said about the nature of algorithms based on the branch-and-bound principle, it is apparent that the quality of the lower bounds will be a vital factor in determining the effectiveness of the method. It is with this point in mind that this paper has been devoted to establishing a tight lower bound to the optimal tour of the *TSP*.

There is a second, and equally important, area where a good lower bound on the *TSP* solution is useful: to establish a reference point against which to compare the results of approximate heuristic methods of solution to the problem.

SOME BOUNDS AND THEIR RELATION

SEVERAL DIFFERENT bounds to the optimal solution of the *TSP* have been proposed and evaluated in the past.[4,6,7] Of these, two are exact solutions to problems that are closely related to the *TSP*.

Bound from the Assignment Problem (*AP*)

The classical linear assignment problem can be stated as follows:

$$\text{Minimize } z=\sum_{i=1}^{i=n}\sum_{j=1}^{j=n} d_{ij}x_{ij}, \tag{1}$$

subject to

$$\sum_i x_{ij}=\sum_j x_{ij}=1, \qquad (\text{for all } i \text{ and } j=1,2,\cdots,n) \tag{2}$$

and

$$x_{ij}=0 \quad \text{or} \quad 1. \tag{3}$$

Equations (1) to (3), together with the added constraint that the solution must form a tour, can also represent a formulation of the *TSP* in which $x_{ij}=1$ means that link (i,j) is in the tour and $x_{ij}=0$ means that it is not. Thus, the minimum value of z for the *AP* in (1) is a lower bound to the optimal solution of the *TSP*.

Bound from the Shortest Spanning Tree (*SST*)

The *TSP* can be represented by a graph of n nodes completely interconnected by links whose 'weights' are given by the distance matrix D. Let us suppose that it is specified that link (i_1, i_2) is in the optimal tour. If this link is removed from the tour, a chain of $n-1$ links is obtained going through all the cities starting at i_1 and finishing at i_2. A spanning tree of a graph of n points is defined as any collection of $n-1$ links chosen so that every node can be reached from every other node, but the links need not necessarily form a single chain. Thus, the length of the shortest spanning tree $L(SST)$ is a lower bound to the length of the chain, and hence the length of the *SST* plus $d_{i_1i_2}$ is a lower bound to the length of the optimal tour of the *TSP*.

In general, no link (i_1, i_2) in the optimal tour will be known, but the longest link in the tour must be at least as long[4] as $\max_i (d_{is})$, where s is the second nearest node to node i. Thus, $d_{i_1 i_2} \geqq \max_i (d_{is})$ and $L(SST) + \max_i (d_{is})$ is a valid lower bound to the length of the solution to the TSP.

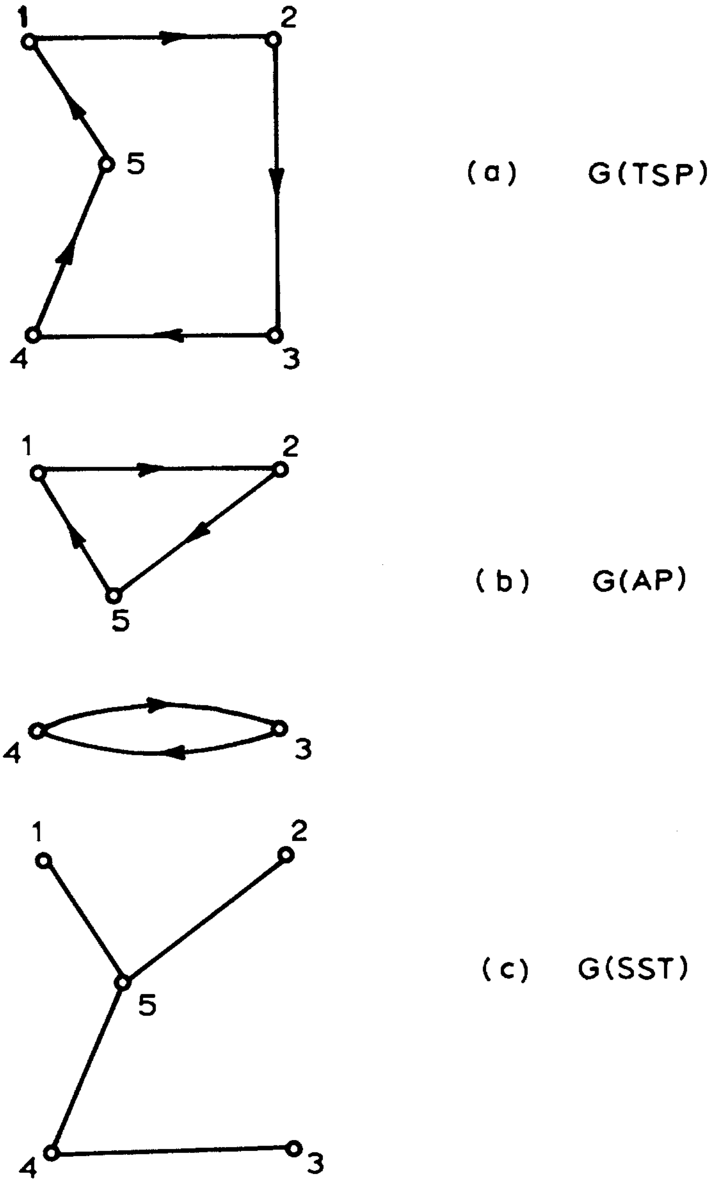


Fig. 1. Graphs for the five-city travelling-salesman problem (TSP), assignment problem (AP), and shortest spanning tree (SST).

Duality between the AP and SST

Let us define $G(TSP)$ as the graph formed by nodes corresponding to the cities of the TSP and the links that are used by the optimal tour. Thus, the graph $G(TSP)$ is a single circuit. Similarly, let us define the graphs $G(AP)$ and $G(SST)$ formed with the same nodes but having links that appear in the optimal solutions of the AP and SST respectively.

The graph $G(TSP)$, which is shown in Fig. 1(a) for the case of a five-city TSP, has the following properties: (i) The graph is connected, i.e., every node can reach every other node via a path using the links, and (ii) the 'degree' of every vertex is 2, i.e., there are two links joining at each node.

The graph $G(AP)$ does not necessarily have property (i), [as can be seen from the example of Fig 1(b)], but does, by definition, have property (ii). If, however, it happens that the solution to the AP does have property (i) as well, then it is also the solution to the TSP.

The graph $G(SST)$ has property (i), by definition, but does not have property (ii). If, however, it happens that the SST does have property (ii)—except for two 'end' vertices (i_1 and i_2, say), which by necessity must have a degree of unity—then the SST is the shortest chain passing through all the n points. If link (i_1, i_2) is in the optimal travelling-salesman tour, then the links of the SST plus link (i_1, i_2) will be the solution to the TSP; if it is not certain that (i_1, i_2) is in the optimal tour, then a small modification[2,3] is needed in order to obtain the solution to the TSP.

Thus, the solutions of the AP and SST are dual in the sense that they possess properties that are complementary with respect to the properties of the TSP. Two possible avenues of investigation now exist that can lead to solutions for the TSP.

(a) Use the solutions of the AP that possess property (ii) and force these solutions to conform to property (i). This could, for example, be done by a branch-and-bound algorithm,[11] with bounds calculated from the AP.

(b) Use the solutions of the SST that possess property (i) and force them to conform to property (ii), again using a branch-and-bound-like algorithm,[3,10] with bounds calculated from the SST.

However, since these algorithms will depend on bounds that may not be particularly tight,[4] the computation times necessary to produce an optimal answer may be prohibitive for even moderately sized problems; thus, these two methods are mentioned simply to illustrate their relation.

A BETTER LOWER BOUND

A SOLUTION TO the AP will, in general, produce a number of subtours (cycles), as shown, for example, in Fig 2(a). Let the ith of these subtours be called $S_{1,i}$ and let their number be n_1. (We will use the same symbol $S_{1,i}$ also to represent the set of nodes in subtour i.)

A *contraction* is defined as a replacement of a subtour by a single node, thus forming a contracted graph containing n_1 nodes $S_{1,i}$ $(i=1, 2, \cdots, n_1)$. The dis-

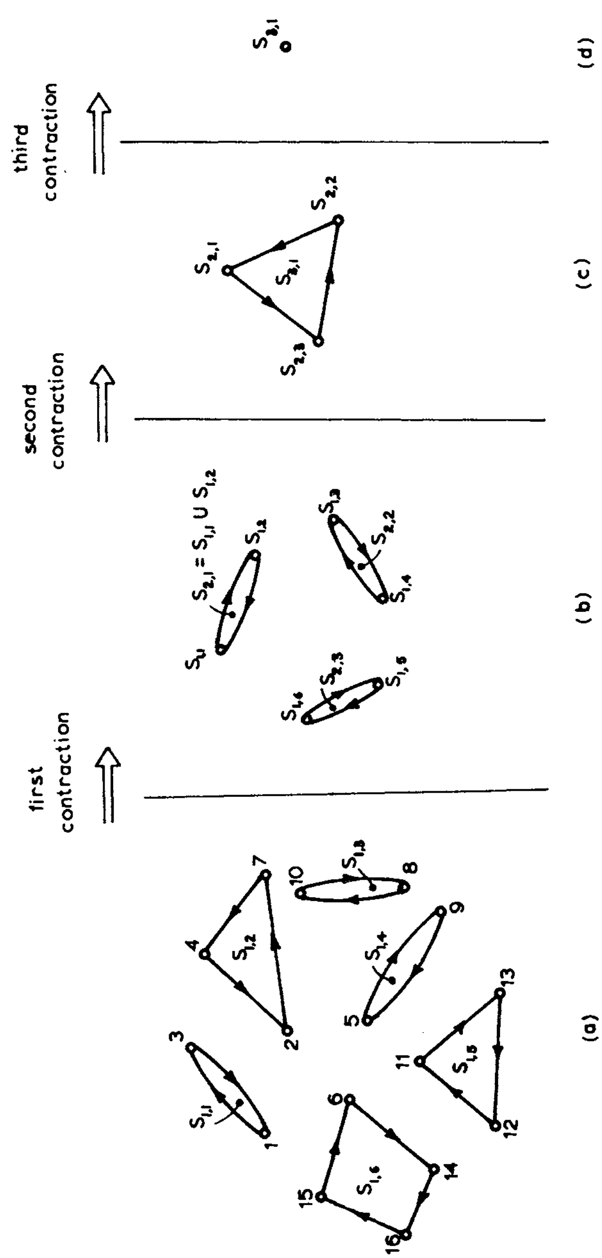


Fig. 2. The contraction process.

tance matrix $D_1 = [d_1(S_{1,i}, S_{1,j})]$ of the contracted graph is taken as:

$$d_1(S_{1,i}, S_{1,j}) = \min_{k_i \epsilon S_{1,i},\, k_j \epsilon S_{1,j}} [f_1(k_i, k_j)], \tag{4}$$

where $F_1 = [f_1(k_i, k_j)]$ is the resulting relative distance matrix at the end of the solution to the AP by (say) the Hungarian method. In Fig. 2(a), for example,

$$d_1(S_{1,5}, S_{1,6}) = \min_{k_5 \epsilon \{11,12,13\},\, k_6 \epsilon \{6,14,15,16\}} [f_1(k_5, k_6)].$$

A solution to the AP of this contracted problem under the matrix D_1 may still produce subtours having the previous subtours as nodes. Figure 2(b) shows one possible formation of the new subtours $S_{2,i}$ $(i = 1, 2, \cdots, n_2)$, where n_2 is their total number. $[n_2 = 3$ in Fig 2(b)]. These subtours may again be contracted into nodes to form a new problem, where the new distance matrix $D_2 = [d_2(S_{2,i}, S_{2,j})]$ is calculated from an equation similar to (4); i.e.,

$$d_2(S_{2,i}, S_{2,j}) = \min_{k_i \epsilon S_{2,i},\, k_j \epsilon S_{2,j}} [f_2(k_i, k_j)], \tag{5}$$

where the various S_2 are the union of all the sets S_1 forming the particular subtour, and where $F_2 = [f_2(k_i, k_j)]$ is the relative distance matrix at the end of the second solution to the AP.

A solution to the AP of the new doubly contracted graph may still produce subtours and the cyclic process of solution-contraction can be continued until the problem is reduced to a single node.

Compression is defined as the transformation of a matrix that does not satisfy the triangularity condition of metric space into one that does. Thus, to compress a matrix M, what is necessary is to replace every element $m_{i,j}$ for which $m_{ij} > m_{ik} + m_{kj}$ for some k by the value of $\min_k [m_{ik} + m_{kj}]$ and to continue this replacement until all $m_{ij} \leqq m_{ik} + m_{kj}$ for any k.

Description of the Algorithm

An algorithm for the calculation of the lower bound will now be stated; its validity is demonstrated by the theorem that follows.

Step 1. Set a matrix M equal to the initial distance matrix $[d_{ij}]$, and set $L = 0$.

Step 2. If the matrix M satisfies the triangularity condition of metric space, go to *step* 3; if not, compress M until $m_{ij} \leqq m_{ik} + m_{kj}$ for any k, as was explained earlier.

Step 3. Solve the assignment problem using matrix M and let $V(AP)$ be the value of this solution. Set $L = L + V(AP)$.

Step 4. Contract the matrix M by replacing subtours (formed as a result of the solution to the assignment problem at *step* 3) by single nodes. [*See* equations (4) and (5).]

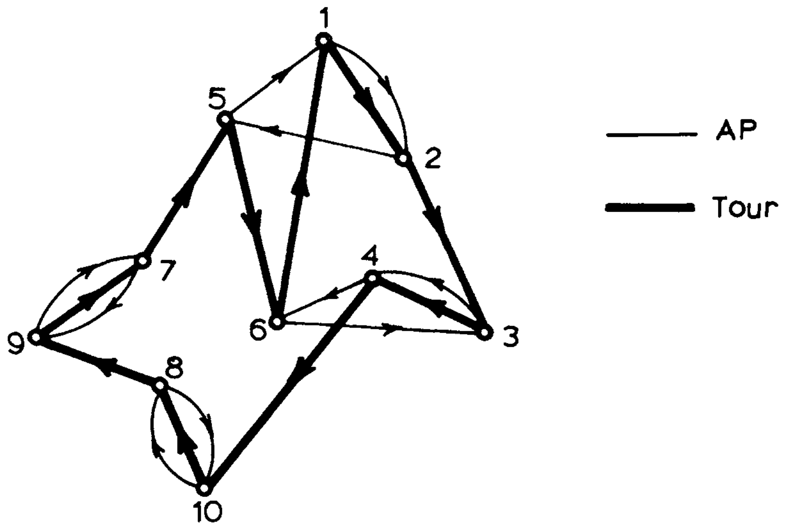
Step 5. If the contracted matrix M is a 1 by 1 matrix, go to *Step* 6; otherwise, return to *step* 2.

Step 6. End. The value of L is a lower bound to the value of the *TSP*.

THEOREM. *The sum of the values of the solution to the AP's obtained during the 'solution-contraction-compression' process (up to the stage when the contracted problem becomes a single point) is a valid lower bound to the TSP.*

Proof. It is well known that entry (i, j) in the relative distance matrix resulting at the end of the solution to an AP represents a lower bound on the extra distance that would result from including link (i, j) into the solution.

Consider subtour $S_{1,i}$ formed at the end of the solution to the first AP. Any tour passing through all the n points will have at least two links incident (one directed inward and the other outward) with one or more of the cities in $S_{1,i}$; and the actual number of such links is obviously even. In Fig 3(a) a tour is shown in heavy lines, whereas the solution to the AP is shown in light lines. After contrac-



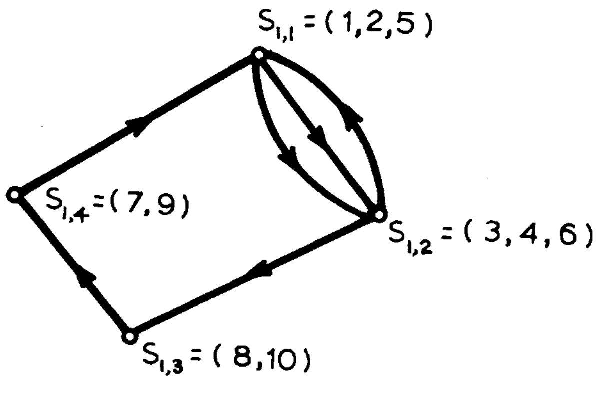


Fig. 3. Transformation of a tour under contraction.

tion the tour becomes the assignment of Fig 3(b) with two links incident on $S_{1,3}$ and $S_{1,4}$ but four links incident on $S_{1,1}$ and $S_{1,2}$.

Now, if the triangularity condition applies, then

$$d_1(S_{1,4}, S_{1,3}) \leqq d_1(S_{1,4}, S_{1,1}) + d_1(S_{1,1}, S_{1,2}) + d_1(S_{1,2}, S_{1,3}), \qquad (6)$$

and hence the assignment of Fig. 4—in which links $(S_{1,4}, S_{1,1})$, $(S_{1,1}, S_{1,2})$, and $(S_{1,2}, S_{1,3})$ have been replaced by link $(S_{1,4}, S_{1,3})$—has a value less than or equal to the value of the assignment of Fig. 3(b). Since the assignment of Fig. 4 has two incident links per node, and the solution to the AP of the contracted problem is the shortest such assignment, the value of the solution to the AP, $V(AP_1)$, is a lower bound on the value of the assignment of Fig. 3(b), i.e., on the value of the increment in cost that would be necessary in order to link the various subtours together. It is apparent that, since the graph of the assignment obtained by any tour after the first contraction [i.e., the assignment corresponding to Fig. 3(b)] is Eulerian [i.e., contains an Euler cycle], it is always possible[1] to transform it into

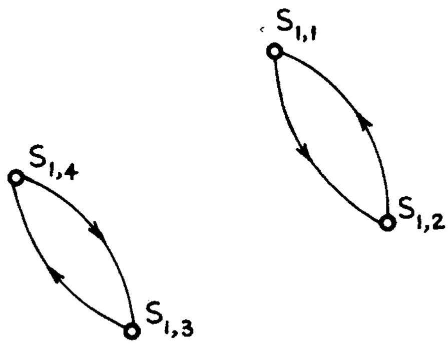


Fig. 4. The assignment corresponding to Fig. 3(b) with two links per node.

an assignment with only two incident links per node, and having lower (or equal) cost, by replacing a chain of links from one node to another by a single link between the nodes, as illustrated above.

If, on the other hand, the relative cost matrix D_1 does not satisfy the triangularity condition, then (6) may not apply. In this case, the matrix D_1 may be compressed first, and the value of the solution to the AP under the compressed matrix will then be the lower bound on the value of the increment in cost that would be necessary in order to link the subtours together. This is so, because compressing a matrix can only reduce (or leave unchanged) the cost of any assignment under the original matrix.

Similarly, $V(AP_2)$, the cost of the solution to the AP after the second contraction, is a lower bound on the cost of linking the subtours resulting from this contraction, and so on for the third, fourth, and later contractions. Therefore,

$$L = \sum_{i=0}^{i=k} V(AP_i), \qquad (7)$$

where $V(AP_0)$ is the value of the initial solution to the AP under the initial matrix, and k is the number of contractions necessary to reduce the graph of the problem to

TABLE I

THE INITIAL DISTANCE MATRIX FOR THE EXAMPLE

	1	2	3	4	5	6	7	8	9
2	32								
3	41	22							
4	22	30	63						
5	20	42	41	36					
6	57	51	30	78	45				
7	54	61	45	72	36	22			
8	32	20	10	54	32	32	41		
9	22	54	60	20	22	67	57	50	
10	45	31	36	64	28	20	10	32	50

a single node] is a valid lower bound to the cost of the solution of the TSP. Hence the theorem.

One should perhaps note here that, even if the initial distance matrix satisfies the triangularity condition, the consequent relative cost matrices may not, and their compression may be necessary at any one stage.

EXAMPLE

CONSIDER THE 10-CITY TSP whose symmetrical distance matrix is given in Table I.

The solution to the initial AP gives the value $V(AP_0)=184$. The resulting relative cost matrix is given in Table II, and the solution is given by the graph of Fig. 5.

The contraction of the graph of Fig. 5 produces a four-node graph whose distance matrix can be calculated, according to (4), to be as shown in Table III(a). This distance matrix does not satisfy the triangularity condition and is therefore compressed into the matrix shown in Table III(b).

The solution to the AP under the matrix of Table III(b) gives the value $V(AP_1)=20$. The resulting relative cost matrix is given in Table IV, and the solution to this AP is given by the graph of Fig. 6.

TABLE II

THE RELATIVE COST MATRIX RESULTING FROM THE SOLUTION TO THE INITIAL ASSIGNMENT PROBLEM

	1	2	3	4	5	6	7	8	9	10
1	X	12	31	2	0	37	44	24	2	37
2	0	X	0	18	10	19	39	0	22	31
3	19	0	X	41	19	8	33	0	38	26
4	2	30	53	X	16	58	62	46	0	56
5	0	22	31	16	X	25	26	24	2	20
6	25	19	8	46	13	X	0	12	35	0
7	32	39	33	50	14	0	X	31	35	0
8	12	0	0	34	12	12	31	X	30	24
9	2	34	50	0	2	47	47	42	X	42
10	25	31	26	44	8	0	0	24	30	X

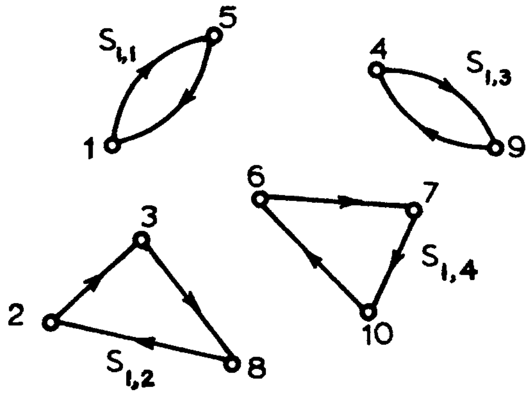


Fig. 5. The first solution to the AP for the example.

The contraction of the graph of Fig. 6 produces a two-node graph whose distance matrix is calculated from (5) to be as shown in Table V. This trivial 2×2 matrix needs no compression, since it satisfies the triangularity condition.

The solution to the trivial AP under the matrix of Table V has a value $V(AP_2) =$

TABLE III

Results for the Four-Node Graph Produced by Contracting the Graph of Fig. 5

(a) *Matrix of contracted graph.*

	1	2	3	4
1	X	12	2	20
2	0	X	18	8
3	2	30	X	42
4	8	8	30	X

(b) *Compressed matrix.*

	1	2	3	4
1	X	12	2	20
2	0	X	2	8
3	2	14	X	22
4	8	8	10	X

10, and the solution is given by the graph of Fig. 7, which becomes a single node after the next contraction.

Thus, a lower bound on the value of the TSP under the matrix of Table I is given by $L = V(AP_0) + V(AP_1) + V(AP_2) = 214$, compared to the value of the optimal solution to the TSP of 216, an error of only 0.93 per cent.

TABLE IV

The Relative Cost Matrix Resulting from the Solution to the Assignment Problem under the Matrix of Table III(b)

	1	2	3	4
1	X	10	0	10
2	0	X	2	0
3	0	12	X	12
4	0	0	2	X

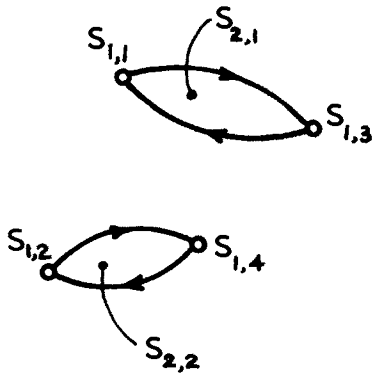


Fig. 6. Solution of the example after the first contraction.

RESULTS

TABLE VI shows experimental results for a number of randomly generated problems varying in size from 10 to 100 cities. The bounds calculated from the above method are compared with the value of the optimal solution to the TSP for the problems, where $n \leqq 40$ cities and with near-optimal solutions for the problems where $n > 40$ cities.

TABLE V

DISTANCE MATRIX FOR THE CONTRACTION OF THE GRAPH OF FIG. 6

	1	2
1	X	0
2	10	X

The optimal solutions to the TSP's with $n \leqq 40$ are derived from a branch-and-bound algorithm[12] similar to that of LITTLE ET AL.,[9] and the near optimal solutions to the TSP's with $n > 40$ are the best of 10 runs (with random starting tours) of the three-optimal procedure of LIN.[8]

Table VI shows that the mean value of the difference between the optimal solution and this bound is only 4.73 per cent for symmetrical and 3.76 per cent for asymmetrical travelling salesman problems. This difference is much smaller (less

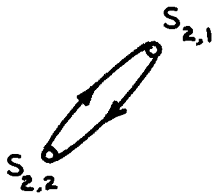


Fig. 7. Solution of the example after the second contraction.

than half) than that obtained by the previously available bounds,[4] except for a recently published bound [7] which is much tighter than the present one, but which requires a considerably greater computational effort.

The computation time to obtain the bound on the TSP is, on the average, only 9 per cent greater than the time required to solve the AP under the same cost matrix. The computation time for the solution to the AP using the Hungarian method varies as kN^3, where k is a constant and N is the size of the matrix. The worst possible case for the computation of the bound, as far as the computation times are concerned, appears when all the subtours at each contraction contain only two cities each, in which case the total computing time spent on solving assignment problems would be

$$kN^3+k(N/2)^3+k(N/4)^3+\cdots=(8/7)kN^3=1.143\,kN^3. \tag{8}$$

Hence, from (8) it can be seen that, at worst, the time required to calculate the

TABLE VI

EXPERIMENTAL RESULTS FOR RANDOM PROBLEMS

Number of cities	Symmetrical problems			Asymmetrical problems		
	Lower bound	Solution to TSP	% difference	Lower bound	Solution to TSP	% difference
10	182	189	3.70	148	152	2.63
20	234[a]	246[a]	4.87[a]	200	211	5.21
30	309	338	8.57	282	293	3.75
40	404	411	1.70	345	360	4.16
60	552	586	6.11	478	495	3.43
80	667	692	3.61	570	591	3.55
100	769	806	4.59	720	747	3.61
		Average	4.73		Average	3.76

[a] This is the problem of Croes.[8]

suggested bound to the TSP is only 14.3 per cent greater than the time required to solve an AP of the same size. (It should be noted here that the computing times required for the contraction and compression parts of the process vary only as N^2.)

REFERENCES

1. C. BERGE, *The Theory of Graphs*, Methuen, London, 1962.
2. N. CHRISTOFIDES, "A New Algorithm for the Travelling-Salesman Problem," Imperial College Report 69/11, London, 1969.
3. ———, "The Shortest Hamiltonian Chain of a Graph," *J. SIAM*, **19,** 689–696 (1970).
4. S. EILON, C. D. T. WATSON-GANDY, AND N. CHRISTOFIDES, *Distribution Management*, Griffin, London, 1971.
5. M. HELD AND R. KARP, "A Dynamic Programming Approach to Sequencing Problems," *J. SIAM*, **10,** 196–210 (1962).

6. ——— AND ———, "The Travelling-Salesman Problem and Minimum Spanning Trees," *Opns. Res.* **18,** 1138–1162 (1970).
7. ——— AND ———, "The Travelling-Salesman Problem and Minimum Spanning Trees—Part II," *Mathematical Programming,* **1,** 6–25 (1971).
8. S. LIN, "Computer Solutions of the Travelling-Salesman Problem," *Bell. Syst. Tech. J.* **44,** 2245–2269 (1965).
9. J. D. C. LITTLE, K. G. MURTY, D. W. SWEENEY, AND C. KAREL, "An Algorithm for the Travelling-Salesman Problem," *Opns. Res.* **11,** 972–989 (1963).
10. A. K. OBRUCA, "Spanning-Tree Manipulation and the Travelling-Salesman Problem," *Computer J.* **10,** 374–377 (1968).
11. D. SHAPIRO, "Algorithms for the Solution of the Optimal Cost Travelling-Salesman Problem," Sc.D. Thesis, Washington Univ., St. Louis, 1966.
12. R. SPEARS AND N. CHRISTOFIDES, "A Branch-and-Bound Algorithm for the Travelling-Salesman Problem," (to appear).