Travelling Salesperson Problem

Let G = (V, E) be a directed graph with edge costs c_{ij}. The variable c_{ij} is defined such that $c_{ij} > 0$ for all i and j and $c_{ij} = \infty$ if (i, j) $\notin$ E. Let |V| = n and assume n > 1.A tour of G is a directed simple cycle that includes every vertex in V. The cost of a tour is the sum of the cost of the edges on the tour. The travelling salesperson problem is to find a tour of minimum cost.

a tour to be a simple path that starts and ends at vertex 1. Every tour consists of an edge(1, k)for some k Є V – {1} and a path from vertex k to vertex 1. The path from vertex k to vertex1 goes through each vertex in V - {1, k} exactly once. It is easy to see that if the tour is optimal, then the path from k to 1 must be a shortest k to 1 path going through all vertices in V – {1, k}. Hence, the principle of optimality holds. Let g(i, S)be the length of a shortest path starting at vertex i, going through all vertices in S, and terminating at vertex 1.The function g(1, V – {1}) is the length of

an optimal salesperson tour. From the principal of optimality it follows that

$$\mathbf{g(1, V - \{1\}) = \min_{2 \le k \le n} \{c_{1k} + g(k, V - \{1, k\})\}} \qquad (1)$$

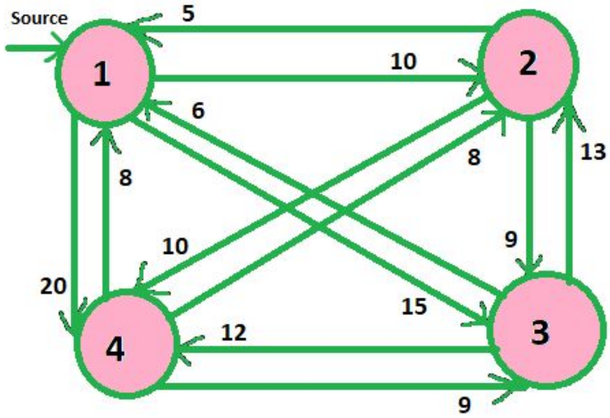
Generalizing(1) we obtain for (i $\notin$ S)

$$\mathbf{g(i, S) = \min_{j \in S}\{c_{ij} + g(j, S - \{j\})\}} \qquad (2)$$

Equation 1 can be solved for g(1, V – {1}) if we know g(k, V - {1, k}) for all choices of k. The g values can be obtained by using 2. g(i, Φ) = c_{i1},1 ≤ i ≤ n. Hence, we can use (2) to obtain g(i, S) for all S of size 1.Then we can obtain g(i, S) for S with |S|= 2, and soon. When |S| < n – 1, the values of i and S for which g(i, S) is needed are such that i ≠ 1,1 $\notin$ S, and i $\notin$ S.

Problem

Consider the following digraph to solve the TSP problem where vertex 1 is the source vertex.



Adjacency matrix or **cost matrix** of the above graph given below

	1	2	3	4
1	0	10	15	20
2	5	0	9	10
3	6	13	0	12
4	8	8	9	0

Thus g(2, Φ)= c21 = 5, g(3, Φ)= c31 = 6, and g(4, Φ) = c41 = 8. Using (2), we obtain

g(2,{3})= c23+g(3, Φ)= 15 g(2,{4})= c24+g(4, Φ)= 18

g(3,{2})= c32+g(2, Φ)= 18 g(3,{4})= c34+g(4, Φ)= 20

g(4,{2})= c42+g(2, Φ)= 13 g(4,{3})= c43+g(3, Φ)= 15

Next, we compute g(i, S) with |S| =2, i≠1,1∉S and i ∉S.

g(2,{3,4})= min {c23+g(3,{4}), c24+g(4, {3})}= min{29, 25}= 25

g(3,{2,4})= min {c32+g(2,{4}), c34+g(4, {2})}= min{31, 25}= 25

g(4,{2,3})= min {c42+g(2,{3}), c43+g(3, {2})}= min{23, 27}= 23

Finally, from (1) we obtain

g(1,{2, 3, 4})= min {c12 + g(2,{3,4}), c13+g(3,{2,4}), c14+g(4,{2, 3})}

= min {35,40,43}

= 35

So the optimal tour of the graph has length 35. A tour of this length can be constructed by considering the minimum value return by the above solution. So J(I,{2,3,4})= 2. Thus the tour starts from 1 and goes to 2. The remaining tour can be obtained from g(2,{3,4}). So J(2,{3,4})= 4.Thus the next edge is (2, 4).The remaining tour is for g(4, {3}). So J(4,{3})= 3.The optimal tour is 1, 2, 4, 3, 1.

Algorithm

C ({1}, 1) = 0

for s = 2 to n do

for all subsets S Є {1, 2, 3, … , n} of size s and containing 1

C (S, 1) = ∞

for all j Є S and j ≠ 1

C (S, j) = min {C (S – {j}, i) + d(i, j) for i Є S and i ≠ j}

Return min C ({1, 2, 3, …, n}, j) + d(j, i)

Time Complexity

If we solve recursive equation we will get total (n-1) $2^{(n-2)}$ sub-problems, which is $O(n2^n)$.

Each sub-problem will take O(n) time (finding path to remaining (n-1) nodes).

Therefore total time complexity is $O(n2^n) * O(n) = O(n^2 2^n)$