

## Experiment No. 11

**Title:** Write a program to find solution to travelling salesperson problem using dynamic programming

### Theory:

Travelling Salesman Problem is defined as “Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?”

Let say there are some villages (1, 2, 3, 4, 5). To work with worst case let assume each villages connected with every other villages. And there is a Salesman living in village 1 and he has to sell his things in all villages by travelling and he has to come back to own village 1.

He has to travel each village exactly once, because it is waste of time and energy that revisiting same village. This is same as visiting each node exactly once, which is Hamiltonian Circuit. But our problem is bigger than Hamiltonian cycle because this is not only just finding Hamiltonian path, but also we have to find shortest path.

Finally the problem is we have to visit each vertex exactly once with minimum edge cost in a graph.

Brute Force Approach takes  $O(n^n)$  time, because we have to check  $(n-1)!$  paths (i.e all permutations) and have to find minimum among them.

The correct approach for this problem is solving using Dynamic Programming.

Dynamic Programming can be applied only if main problem can be divided into sub-problems.

	1	2	3	4
1	0	4	1	3
2	4	0	2	1
3	1	2	0	5
4	3	1	5	0

Here problem is travelling salesman wants to find out his tour with minimum cost.

Say it is  $T(1, \{2,3,4\})$ , means, initially he is at village 1 and then he can go to any of  $\{2,3,4\}$ . From there to reach non-visited vertices (villages) becomes a new problem.

Here we can observe that main problem spitted into sub-problem, this is property of dynamic programming.

$$T(1, \{2,3,4\}) = \text{minimum of } \{ (1,2) + T(2, \{3,4\}) \quad 4+\textcolor{red}{6}=10$$

$$= \{ (1,3) + T(3, \{2,4\}) \quad 1+\textcolor{red}{3}=4$$

$$= \{ (1,4) + T(4, \{2,3\}) \quad 3+\textcolor{red}{3}=6$$

Here minimum of above 3 paths is answer but we know only values of  $(1,2)$  ,  $(1,3)$  ,  $(1,4)$  remaining thing which is  $T(2, \{3,4\})$  ...are new problems now. First we have to solve those and substitute here.

$$T(2, \{3,4\}) = \text{minimum of } \{ (2,3) + T(3, \{4\}) \quad 2+\textcolor{red}{5}=7$$

$$= \{ (2,4) + T(4, \{3\}) \quad 1+\textcolor{red}{5}=6$$

$$T(3, \{2,4\}) = \text{minimum of}$$

$$= \{ (3,2) + T(2, \{4\}) \quad 2+\textcolor{red}{1}=3$$

$$= \{ (3,4) + T(4, \{2\}) \quad 5+\textcolor{red}{1}=6$$

$$T(4, \{2,3\}) = \text{minimum of}$$

$$= \{ (4,2) + T(2, \{3\}) \quad 1+\textcolor{red}{2}=3$$

$$= \{ (4,3) + T(3, \{2\}) \quad 5+\textcolor{red}{2}=7$$

$$T(3, \{4\}) = (3,4) + T(4, \{\}) \quad 5+0=5$$

$$T(4, \{3\}) = (4,3) + T(3, \{\}) \quad 5+0=5$$

$$T(2, \{4\}) = (2,4) + T(4, \{\}) \quad 1+0=1$$

$$T(4, \{2\}) = (4,2) + T(2, \{\}) \quad 1+0 = 1$$

$$T(2, \{3\}) = (2,3) + T(3, \{\}) \quad 2+0 = 2$$

$$T(3, \{2\}) = (3,2) + T(2, \{\}) \quad 2+0=2$$

Here  $T(4, \{\})$  is reaching base condition in recursion, which returns 0 (zero) distance. This is where we can find final answer,

$$T(1, \{2,3,4\}) = \text{minimum of } \{ (1,2) + T(2, \{3,4\}) \quad 4+6=10$$

in this path we have to add +1 because this path ends with 3. From there we have to reach 1 so 3->1 distance 1 will be added total distance is 10+1=11

=  $\{ (1,3) + T(3, \{2,4\}) \quad 1+3=4$  in this path we have to add +3 because this path ends with 3. From there we have to reach 1 so 4->1 distance 3 will be added total distance is 4+3=7

=  $\{ (1,4) + T(4, \{2,3\}) \quad 3+3=6$  in this path we have to add +1 because this path ends with 3. From there we have to reach 1 so 3->1 distance 1 will be added total distance is 6+1=7

Minimum distance is 7 which includes path 1->3->2->4->1.

After solving example problem we can easily write recursive equation.

## Recursive Equation

$$T(i, s) = \min ( (i, j) + T(j, S - \{j\}) ); S \neq \emptyset ; j \in S ;$$

S is set that contains non visited vertices

$= (i, 1) ; S = \emptyset$ , This is base condition for this recursive equation.

Here,

$T(i, S)$  means We are travelling from a vertex “i” and have to visit set of non-visited vertices “S” and have to go back to vertex 1 (let we started from vertex 1).

$(i, j)$  means cost of path from node i to node j

If we observe the first recursive equation from a node we are finding cost to all other nodes  $(i, j)$  and from that node to remaining using recursion  $(T(j, \{S-j\}))$

But it is not guarantee that every vertex is connected to other vertex then we take that cost as infinity. After that we are taking minimum among all so the path which is not connected get infinity in calculation and won't be consider.

If S is empty that means we visited all nodes, we take distance from that last visited node to node 1 (first node). Because after visiting all he has to go back to initial node.

## Time Complexity

Since we are solving this using Dynamic Programming, we know that Dynamic Programming approach contains sub-problems.

Here after reaching ith node finding remaining minimum distance to that ith node is a sub-problem.

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  $O(n2^n) * O(n) = O(n^2 2^n)$

Space complexity is also number of sub-problems which is  $O(n2^n)$

### **Algorithm**

**Students are expected to write their own algorithm for Travelling salesman Problem**