Roll No: 33231
Name: Prajakta Keer

ASSIGNMENT 5

~~Problem Statement~~ Title: Travelling Salesman Problem

Problem statement: Write a program to solve the travelling salesman problem and to print the path and the cost using DP.

Objective: To understand and implement dynamic programming algorithm ~~by~~ for solving Travelling salesperson problem and study dynamic programming

Theory: A travelling salesman is getting ready for a big sales tour. Starting at his hometown, suitcase in hand, he will conduct a journey in which each of his target cities is visited exactly once before he returns home. Given the pairwise costs / distances between cities, what is the best order in which to visit them, so as to minimize ~~to~~ the overall cost / distance travelled?

Let $G = (V, E)$ be a directed graph defining an instance of the travelling salesperson problem. Let c_{ij} be the cost of edge $<i,j>$, $c_{ij} = \infty$ if $<i,j>$ does not belong to E and let $|V| = n$. Without loss of generality, we may assume that every tour starts and ends at vertex 1. So, the solution space S is given by $S = \{1, \pi, 1 \mid \pi$ is a permutation of $(2, 3, \ldots, n)\}$. $|S| = (n-1)!$ The size of S may be reduced by restricting S so that $(1, i_1, i_2, \ldots, i_{n-1}, 1) \in S$ iff $<i_j, i_{j+1}> \in E$, $0 \le j \le n-1$, $i_0 = i_n = 1$. A tour of G is a directed 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.

Dynamic Programming Strategy

Without loss of generality, regard a tour to be simple path that starts and ends at vertex 1. Every tour consists of an

33231

edge $<1,k>$ for some $k \in V-\{1\}$ and a path from vertex k to vertex 1. The path from vertex k to vertex 1 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. $g(1, V-\{1\})$ is the length of an optimal salesperson tour. From the principal of optimality it follows that

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

Generalizing eq^n ① we obtain

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

Eq^n ① may be solved for $g(1, V-\{1\})$ if we know $g(k, V-\{1,k\})$ for all choices of k. The g values may be obtained by using eq^n ②. Clearly, $g(i,\phi) = c_{i1}$, $1\le i\le n$. Hence we may use eq^n ② to obtain $g(i,S)$ for all S of size 1. Then we can obtain $g(i,S)$ for S with $|S|=2$ etc. When $|S|<n-1$, the values of i and S for which $g(i,S)$ is needed are such that $i\ne 1$; 1 and i do not belong to S

<u>Algorithm</u> <u>for</u> <u>TSP</u> <u>using</u> <u>DP</u>:

```
g(1, {1}) = 0
for s = 2 to n do
    for all subsets S of size s and containg 1:
        g(1,S) = ∞
        for all j ∈ S ; j ≠ 1 do
            g(i,S) = min j∈S {cij + g(j, S-{j})}
return min g(i,S)
```

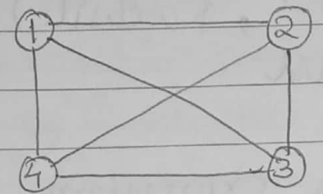
<u>Analysis</u> : The brute-force approach is to evaluate every possible tour and return the best one. Since there are

33231

(n-1)! possiblities of visiting the n points, which results in a runtime of $O(n!)$

By DP: There are atmost $2^n * n$ subproblems and each one takes linear time to solve. The total running time is therefore $O(n^2 2^n)$.

Example:



$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

$g(2, \phi) = C_{21} = 5$; $g(3, \phi) = C_{31} = 6$; $g(4, \phi) = C_{41} = 8$

$g(2, \{3\}) = C_{23} + g(3, \phi) = 15$; $g(2, \{4\}) = 18$

$g(3, \{2\}) = 18$; $g(3, \{4\}) = 20$

$g(4, \{2\}) = 13$ $g(4, \{3\}) = 15$

$g(2, \{3,4\}) = \min(C_{23} + g(3, \{4\}), C_{24} + g(4, \{3\})\} = 25$

$g(3, \{2,4\}) = \min(C_{32} + g(2, \{4\}), C_{34} + g(4, \{2\})\} = 25$

$g(4, \{2,3\}) = \min(C_{42} + g(2, \{3\}), C_{43} + g(3, \{2\})\} = 23$

Finally,

$g(1, \{2,3,4\}) = \min\{C_{12} + g(2, \{3,4\}), C_{13} + g(3, \{2,4\}) + g(4, \{2,3\}\}$

$= \min\{35, 40, 43\}$

$= 35$

$J(1, \{2,3,4\}) = 2$

$J(2, \{3,4\}) = 4$

$J(4, \{3\}) = 3$

Final optimal tour = 1 - 2 - 4 - 3 - 1

Cost of tour = 35

Input and Output :

Input = Cost matrix TSP graph

Output = Minimum cost of tour and its path

Conclusion :

The dynamic programming strategy for TSP is studied and implemented.