DESIGN AND ANALYSIS OF ALGORITHMS

Session -22

TRAVELLING SALESPERSON PROBLEM

THE TRAVELLING SALESPERSON PROBLEM - INTRODUCTION

- A **traveler** needs to **visit all the cities from a list**, where **distances** between all the cities are known and each city should be **visited just once**.

- What is the **shortest possible route** that he visits each city exactly once and **returns to the origin city**?

- Travelling salesman problem is the most **notorious computational problem**. We can use **brute-force approach** to evaluate **every possible tour** and **select the best one**. For **n number of vertices** in a graph, there are **(n - 1)! number of possibilities.**

- Instead of brute-force using **dynamic programming approach**, **the solution can be obtained in lesser time**, though there **is no polynomial time algorithm.**

Problem Definition:

- Let **G (V, E)** be a directed graph with **edge cost $c_{i,j}$** is defined such **that $c_{i,j}$ >0 for all i and j and $c_{i,j} = \propto$, if <i, j> $\notin$ E.**
- Let **V = n** and assume n>1.
- The traveling salesman problem is to find a tour of **minimum cost.**
- A tour of **Graph G is** a directed cycle that include **every vertex in V.**
- The **cost of the tour** is **the sum of cost of the edges** on the tour.
- The tour is the shortest path that **starts and ends at the same vertex (i.e.) 1.**

APPLICATION

- Suppose we have to route a postal van to pick up mail from the mail boxes located at ‘n’ different sites.
- An n+1 vertex graph can be used to represent the situation.
- One vertex represent the post office from which the postal van starts and return.
- Edge <i,j> is assigned a cost equal to the distance from site ‘i’ to site ‘j’.
- the route taken by the postal van is a tour and we are finding a tour of minimum length.

DYNAMIC PROGRAMMING APPROACH

- every tour consists of an edge <1,k> for some k $\in$ V-{} 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.
- the function which is used to find the path I

$$g(1,V-\{1\}) = \min\{ cij + g(j,s-\{j\})\}$$

- 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.

DYNAMIC PROGRAMMING APPROACH

The function **g(1,v-{1})** is the **length of an optimal tour.**

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

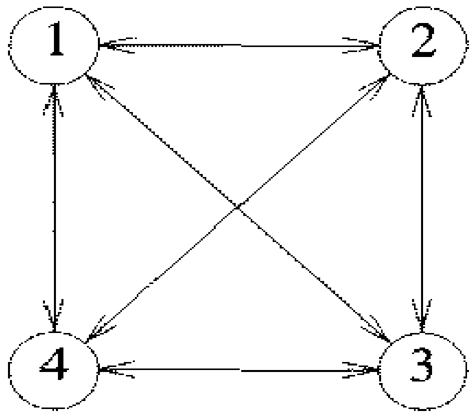
Generalizing (1) ,we obtain

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

Equation (1) can be solved for g(1, V-{1}) if we know g(k, V-{1,k})

for all choices of k.

EXAMPLE-FINDING MINIMUM COST TOUR FOR TSP



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

- $|\ s\ | = 0$
- $g(2,\Phi) = c_{21} \Rightarrow 5$
- $g(3,\Phi) = c_{31} \Rightarrow 6$
- $g(4,\Phi) = c_{41} \Rightarrow 8$

$|S| = 1$

$g(2,\{3\}) = c_{23} + g(3,\Phi) = 9+6 = 15$

$g(2,\{4\}) = c_{24} + g(4,\Phi) = 10+8 = 18$

$g(3,\{2\}) = c_{32} + g(2,\Phi) = 13+5 = 18$

$g(3,\{4\}) = c_{34} + g(4,\Phi) = 12+8 = 20$

$g(4,\{2\}) = c_{42} + g(2,\Phi) = 8+5 = 13$

$g(4,\{3\}) = c_{43} + g(3,\Phi) = 9+6 = 15$

$|S| = 2$

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

$\min\{ 9+20,\ 10+15\}$

$\min\{29, 25\} = 25$

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

$\min\{13+18, 12+13\}$

$\min\{31,25\} = 25$

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

$\min\{8+15, 9+18\}$

$\min\{23, 27\} = 23$

$|S| = 3$

$$g(1, \{2, 3, 4\}) = \min\{c_{12} + g(2, \{3, 4\}),$$
$$c_{13} + g(3, \{2, 4\}),$$
$$c_{14} + g(4, \{2, 3\}) \}$$
$$\min\{10+25, 15+25, 20+23\}$$
$$\min\{35, 40, 43\} = 35$$

Optimal cost is 35

The shortest path is

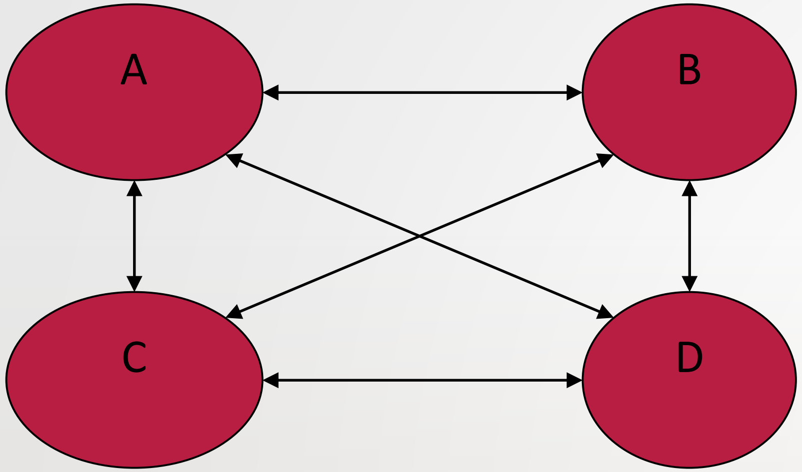
$g(1,\{2, 3, 4\}) = c_{12} + g(2, \{3, 4\})$ => 1 -> 2
$g(2,\{3, 4\}) = c_{24} + g(4,\{3\})$ => 1->2->4
$g(4,\{3\}) = c_{43} + g(3,\{\Phi\})$ => 1->2->4->3->1

The Optimal Tour is :1 → 2 → 4 →3 → 1

ANALYSIS

- An algorithm that proceeds to find an optimal by using(1)and (2) will require **$\theta(n^2 2^n)$ time** as the computation of g(i,S) with |S|= k requires k= 1 comparisons when solving (2).
- This is better than enumerating all n! different tours to find the best one. The most serious drawback of this dynamic programming solution is the space needed, **$0(n2^n)$.**
- This is too large even for modest values of n.

TRAVELLING SALESMAN PROBLEM



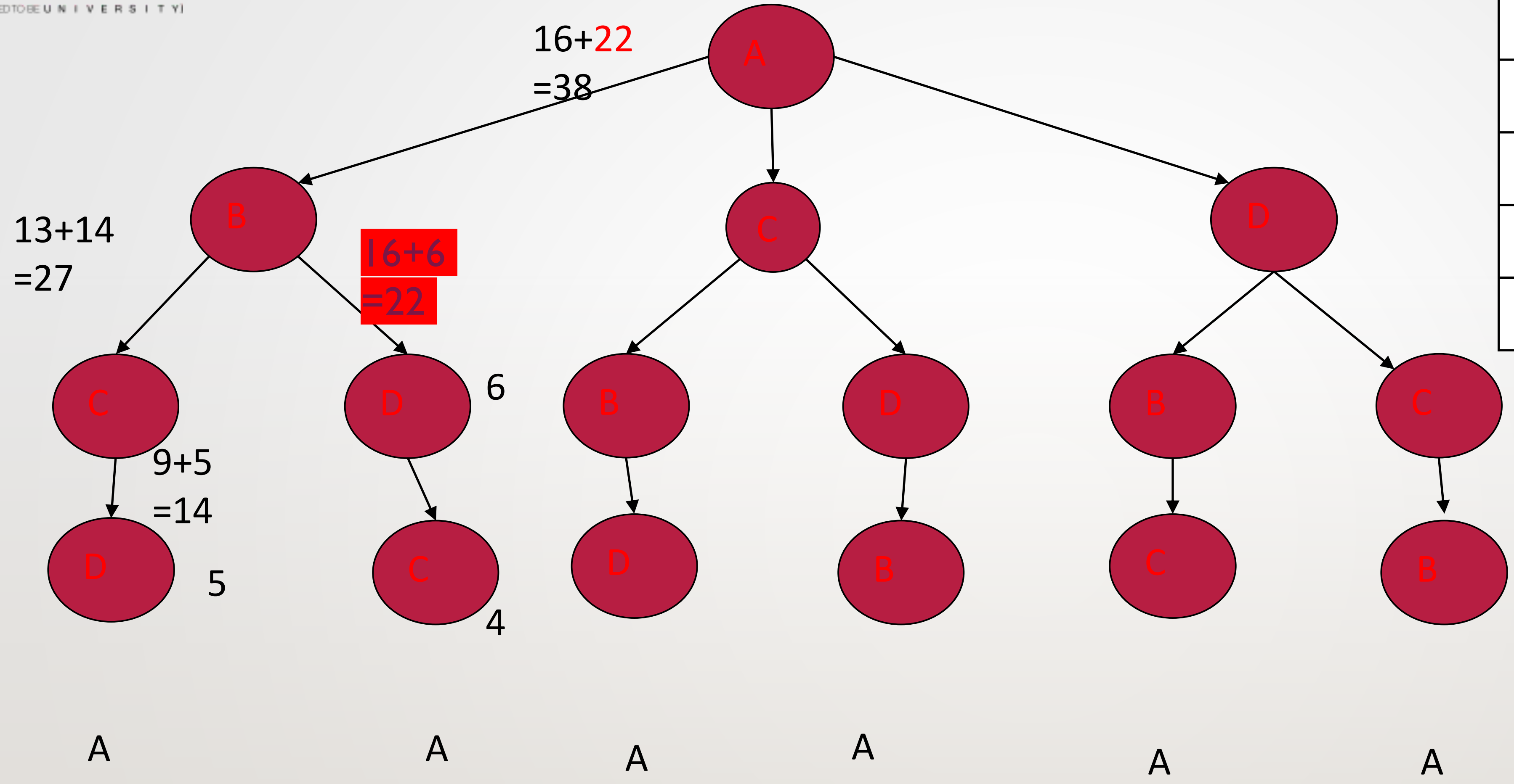
	A	B	C	D
A	0	16	11	6
B	8	0	13	16
C	4	7	0	9
D	5	12	2	0

Graph represents the places to travel

Distance between vertex

A traveler needs to visit all the cities from a list, where distances between all the cities are known and each city should be visited just once. What is the shortest possible route that he visits each city exactly once and returns to the origin city?

Travelling Distance Calculation



	A	B	C	D
A	0	16	11	6
B	8	0	13	16
C	4	7	0	9
D	5	12	2	0

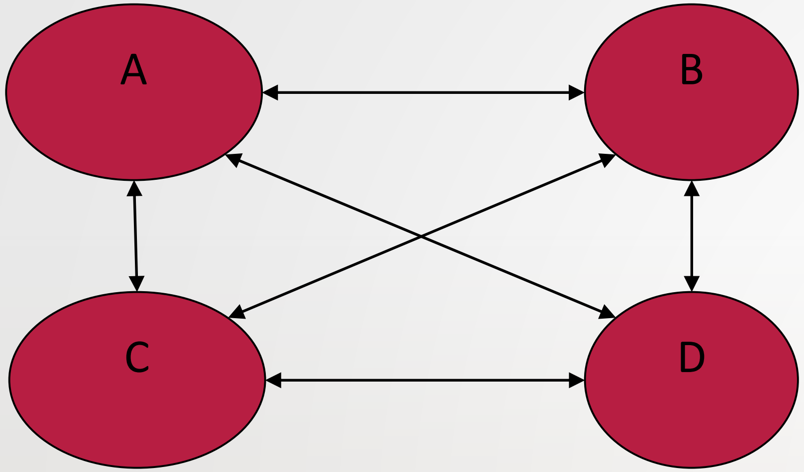
$$g(i,S)=\min[w(i,j)+g(j,\{S-j\})$$

i=starting vertex

S=set of vertex to travel

$j \in S$

Minimum cost Finding from vertex A



g(i, Ø)=CiA

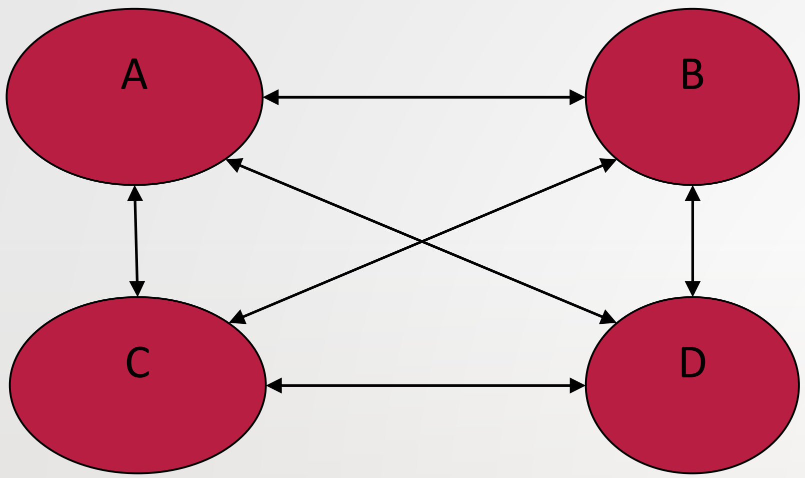
g(B, Ø)=8

g(C, Ø)=4

g(D, Ø)=5

	A	B	C	D
A	0	16	11	6
B	8	0	13	16
C	4	7	0	9
D	5	12	2	0

Minimum cost Finding from vertex A



	A	B	C	D
A	0	16	11	6
B	8	0	13	16
C	4	7	0	9
D	5	12	2	0

$g(B,\{C\})=\min[w(B,C)+g(C, \emptyset)]=13+4=17$

$g(C,\{B\})=\min[w(C,B)+g(B, \emptyset)]=7+8=15$

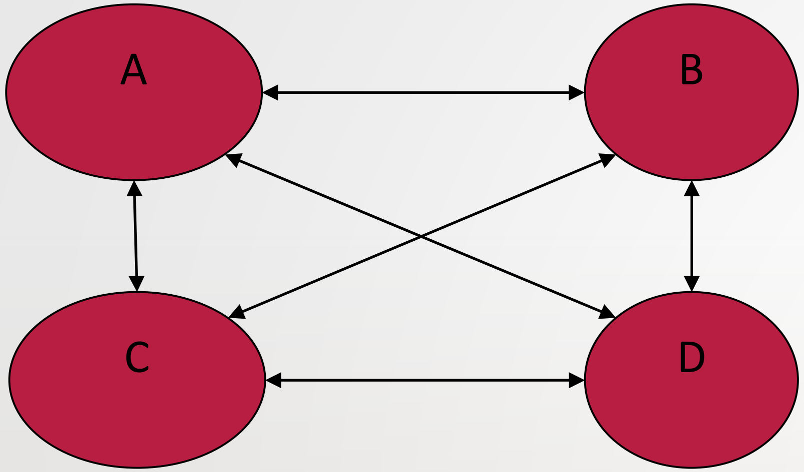
$g(B,\{D\})=\min[w(B,D)+g(D, \emptyset)]=16+5=21$

$g(D,\{B\})=\min[w(D,B)+g(B,V \emptyset\})]=12+8=20$

$g(C,\{D\})=\min[w(C,D)+g(D, \emptyset)]=9+5=14$

$g(D,\{C\})=\min[w(D,C)+g(C, \emptyset)]=2+4=6$

Minimum cost Finding from vertex A



	A	B	C	D
A	0	16	11	6
B	8	0	13	16
C	4	7	0	9
D	5	12	2	0

$g(C,\{B,D\})=\min[w(C,B)+g(B,\{D\})]=7+21=28$

$w(C,D)+g(D,\{B\})=9+20=29$

$g(D,\{B,C\})=\min[w(D,B)+g(B,\{C\})]=12+17=29$

$w(D,C)+g(C,\{B\})=2+15=17$

$g(B,\{C,D\})=\min[w(B,C)+g(C,\{D\})]=13+14=27$

$w(B,D)+g(D,\{C\})=16+6=22$

$g(A,\{B,C,D\})=\min[w(A,B)+g(B,\{C,D\}]=16+22=38$

$w(A,C)+g(C,\{B,D\}=11+28=39$

$w(A,D)+g(D,\{B,C\}=6+17=23$

The shortest path is,

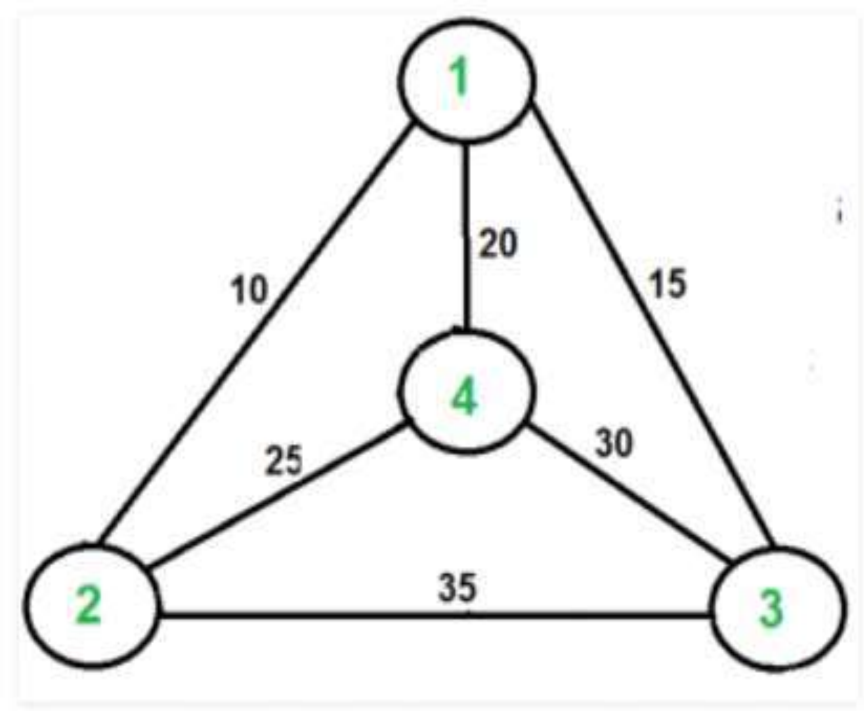
g(A,{B,C,D}) = w(A,D)+g(D,{B,C})=> A->D

g(D,{B,C}) = w(D,C)+g(C,{B})= => A->D->C

g(C,{B}) = min[w(C,B)+g(B, Ø)=> A->D->C->B->A

The Optimal Tour is : A → D → C →B → A

FIND MINIMUM COST TOUR FOR THE FOLLOWING GRAPH



SAMPLE QUESTIONS

- State TSP Problem
- What are the objectives and constraints in solving this problem
- What is the complexity class of the TSP
- Describe the problem statement of the TSP