VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

DISCRETE STRUCTURE FOR COMPUTING

Report
Assignment

Lecturer: Dr. Nguyen An Khuong
Student: Dai Ngoc Quoc Trung 2053537

HO CHI MINH CITY, JUNE 2024

Contents

Chapter 1

Traveling Salesman Problem

For Traveling Salesman Problem, dynamic approach was used to solve the problem.

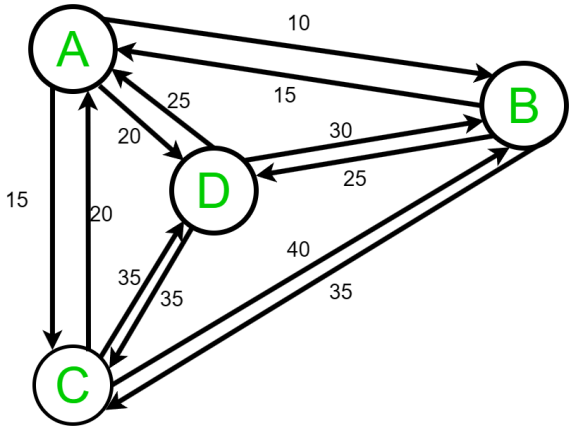

Figure 1.1: A given graph

The solution is to find a shortest path from a start node to every other nodes with possible routes. For example, I can find the path of $1 \to 2 \to 3 \to 4$, or $1 \to 3 \to 2 \to 4$, etc. Then when I have all calculations, I just simply compare the minimum sum of one the path and the distance between the last vertex to the source node with other routes and obtain the result.

To achieve this solution, I create two same sized tables, one for storing the minimum cost of every subsets that a source node traversing to every other nodes, and one for storing the previous vertices. If the problem has n vertices, it will have 2^n subsets that represents n-bit number (a bit mask) where each bit indicates whether a vertex is included in the subset or not. Therefore the size for both matrices must be $2^n \times n$.

For each bit mask, it represents a subset that whether a vertex is included or not, for example 0101 means only vertex A and C are visited. To find out the final result, I need to ensure all vertices are visited, which means the result of the case 1111. At first, all values will be initialized with infinite numbers meaning they are unreached yet, and the

start vertex is initialized with 0. That's look like this:

$$\begin{bmatrix} & A & B & C & D \\ 0000 & \infty & \infty & \infty & \infty \\ 0001 & \mathbf{0} & \infty & \infty & \infty \\ 0010 & \infty & \infty & \infty & \infty \\ 0011 & \infty & \infty & \infty & \infty \\ 0100 & \infty & \infty & \infty & \infty \\ 0101 & \infty & \infty & \infty & \infty \\ 0110 & \infty & \infty & \infty & \infty \\ 0111 & \infty & \infty & \infty & \infty \\ 1000 & \infty & \infty & \infty & \infty \\ 1001 & \infty & \infty & \infty & \infty \\ 1010 & \infty & \infty & \infty & \infty \\ 1011 & \infty & \infty & \infty & \infty \\ 1100 & \infty & \infty & \infty & \infty \\ 1101 & \infty & \infty & \infty & \infty \\ 1110 & \infty & \infty & \infty & \infty \\ 1111 & \infty & \infty & \infty & \infty \end{bmatrix}$$

The value of the table will be updated as following formula:

$$D(i, S) = min(D(K, S - \{K\})) + cost(i, K)$$

Where S is the subset, i is starting vertex, K is destination vertex, and cost(i, K) is the distance that go from i to K.

Let's give an example that a traveler want to from city A to every other cities and back to A. For first iteration, traveler can go to cities B, C, and D. If he goes to either of these cities, it would be marked as visited and determined the value. For example, the distance from A to B is 10, and B is visited, then the shortest distance from A to B is updated as 10.

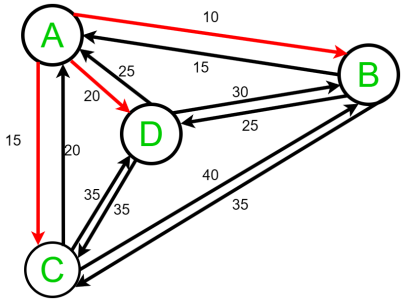

Figure 1.2: Starting from A

	A	B	C	D
0000	∞	∞	∞	∞
0001	**0**	∞	∞	∞
0010	∞	∞	∞	∞
0011	∞	**10**	∞	∞
0100	∞	∞	∞	∞
0101	∞	∞	**15**	∞
0110	∞	∞	∞	∞
0111	∞	∞	∞	∞
1000	∞	∞	∞	∞
1001	∞	∞	∞	**20**
1010	∞	∞	∞	∞
1011	∞	∞	∞	∞
1100	∞	∞	∞	∞
1101	∞	∞	∞	∞
1110	∞	∞	∞	∞
1111	∞	∞	∞	∞

The solution begins with second iteration, from now the traveler can have multiple choices to travel to other cities. After travelling to B, he may travel to city C or D, and the path will be updated according to his preferences. The value on the table will be updated if there is a new shortest distance value. Then the next iterations go on back and forth until the traveler have visited all of these cities based on different subsets. Then the matrix would be like this:

	A	B	C	D
0000	∞	∞	∞	∞
0001	**0**	∞	∞	∞
0010	∞	∞	∞	∞
0011	∞	**10**	∞	∞
0100	∞	∞	∞	∞
0101	∞	∞	**15**	∞
0110	∞	∞	∞	∞
0111	∞	**55**	**45**	∞
1000	∞	∞	∞	∞
1001	∞	∞	∞	**20**
1010	∞	∞	∞	∞
1011	∞	**50**	∞	**35**
1100	∞	∞	∞	∞
1101	∞	∞	**55**	**50**
1110	∞	∞	∞	∞
1111	∞	**80**	**70**	**80**

Now I have the table the filled in with the shortest path going from A to B, C and D. To find out the final result, the values from the last line of the table will be added with the distance the last vertex go to the start vertex. For example, the path of A $\rightarrow$ D $\rightarrow$ B $\rightarrow$ C produce 70, while the distance from C to A is 20. Then the result is 90 and this is also the need result. To obtain the path, I must use the table of previous vertex to backtrack and retrieve the value.