The Traveling Salesman Problem (TSP)

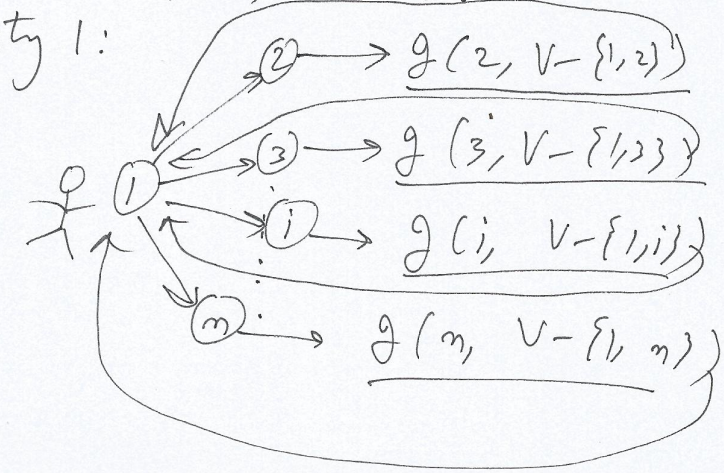
n cities are given: $V = \{1, 2, \ldots, n\}$
distance between city i and city j is given as $c_{ij} = c_{ji}$ $(i \neq j)$ if the traffic on the road is 2-way, otherwise distance from city i to city j is given as c_{ij} and distance from city j to city i is given as c_{ji}. For the 2-way traffic case, we can represent the problem as a weighted undirected complete graph K_n having n vertices and $\binom{n}{2}$ edges $\left(= \frac{n(n-1)}{2}\right)$.

For the 1-way traffic case, we can represent the problem as a weighted ~~un~~directed complete graph K_n having n vertices and $P(n,2) = n(n-1)$ edges.

A salesman will start the journey from city 1, visit each city exactly once, and return back to city 1. Our problem is to find a salesman tour with minimum distance.

Optimal Substructure of TSP

Let $g(i, S)$ denote the minimum distance tour starting from city i, visiting intermediate cities in S $(S \subseteq V - \{1, i\})$, and finally returning back to city 1. Our objective is to find $g(1, V-\{1\})$. Consider the salesman at starting city 1:



Optimal sub tours

Salesman has a choice of moving to 2, 3, ..., i, ..., n. The best option should be selected by the salesman.

Suppose the optimal tour $g(1, V-\{1\})$ is via city i, then the sub tour $g(i, V-\{1,i\})$ should be optimal, otherwise we will not have an optimal solution.

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

Generalizing this for a general i and S:

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

A Dynamic Programming Approach for Solving TSP

We observe that for computing $g(i,S)$ we require values of $g(i, S')$ where $|S'| = |S| - 1$. We follow the Bottom-up approach: start with $S = \phi$, find the values of $g(i, \phi)$. Now using these values, we can compute $g(i,S)$ for all $|S| = 1$. $\Rightarrow g(i,S)\ \forall\ |S| = 2 \Rightarrow \ldots$

$\Rightarrow g(1, V-\{1\})$.

Algorithm TSP$(V, (c_{ij})_{n \times n})$

```
for i = 2 to n
    g(i, φ) ← c_i1

for k = 1 to n-2
    for i = 2 to n
        for all S ⊆ V-{1,i} with |S| = k
            g(i,S) ← min_{j∈S} { c_ij + g(j, S-{j}) }

g(1, V-{1}) ← min_{2≤i≤n} { c_1i + g(i, V-{1,i}) }

return g(1, V-{1})
```

Time Complexity of TSP()

The for loop of lines 1 to 2 has complexity $n-1$

Line 6 has complexity $2|S| = 2k$

The forall loop of line 5 executes for $\binom{n-2}{k}$ steps

The for loop of line 4 executes for $n-1$ steps

The for loop of line 3 goes from $k=1$ to $k=n-2$

Therefore, total complexity of lines 3 to 6 is:

$$\sum_{k=1}^{n-2} (n-1)\binom{n-2}{k}(2k)$$

Complexity of line 7 is $2(n-1)$

Complexity of line 8 is 1

$$\Rightarrow T(n) = (n-1) + \left(\sum_{k=1}^{n-2} (n-1)\binom{n-2}{k}(2k)\right) + 2(n-1) + 1$$

$$= (3n-2) + 2(n-1)\sum_{k=1}^{n-2} \frac{k\,(n-2)!}{k!\,(n-2-k)!}$$

$$= (3n-2) + 2(n-1)(n-2)\sum_{k=1}^{n-2} \frac{(n-3)!}{(k-1)!\,((n-3)-(k-1))!}$$

$$= (3n-2) + 2(n-1)(n-2)\sum_{k'=0}^{n-3} \frac{(n-3)!}{k'!\,((n-3)-k')!}$$

$$= (3n-2) + 2(n-1)(n-2)\sum_{k'=0}^{n-3} \binom{n-3}{k'}$$

$$= 3(n-2) + 2(n-1)(n-2)\cdot 2^{n-3}$$

$$= 3(n-2) + (n-1)(n-2)\cdot 2^{n-2} = \boxed{\Theta(n^2 2^n)}$$

This is much better than the brute-force algorithm taking time $\Theta(n!) = \Theta\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)$

Space Complexity of TSP()

$$S(n) = |\{ g(i,S) \mid S \subseteq V - \{1,i\} \}|$$

$$= \sum_{k=0}^{n-2} (n-1)\binom{n-2}{k} = (n-1)\sum_{k=0}^{n-2}\binom{n-2}{k} = (n-1)2^{n-2}$$

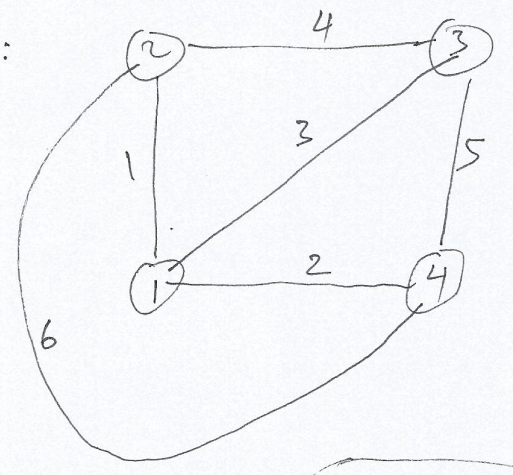
$$= \Theta(n2^n)$$

Time-Space Tradeoff for the TSP problem

BruteForce() Takes $\Theta(n!)$ time and $\Theta(n)$ space

TSP() Takes $\Theta(n^2 2^n)$ time and $\Theta(n2^n)$ space.

Example 1 :



$g(2,\phi) = 1$

$g(3,\phi) = 3$

$g(4,\phi) = 2$

$g(2,\{3\}) = 4 + g(3,\phi) = 7$

$g(2,\{4\}) = 6 + g(4,\phi) = 8$

$g(3,\{2\}) = 4 + g(2,\phi) = 5$

$g(3,\{4\}) = 5 + g(4,\phi) = 7$

$g(4,\{2\}) = 6 + g(2,\phi) = 7$

$g(4,\{3\}) = 5 + g(3,\phi) = 8$

$g(2,\{3\})=7$ $\quad g(2,\{4\})=8$ $\quad g(3,\{2\})=5$ $\quad g(3,\{4\})=7$ $\quad g(4,\{2\})=7$ $\quad g(4,\{3\})=8$

$$g(2,\{3,4\}) = \min\{4+g(3,\{4\}),\ 6+g(4,\{3\})\} = \min\{4+7,\ 6+8\} = 11$$

$$g(3,\{2,4\}) = \min\{4+g(2,\{4\}),\ 5+g(4,\{2\})\} = \min\{4+8,\ 5+7\} = 12$$

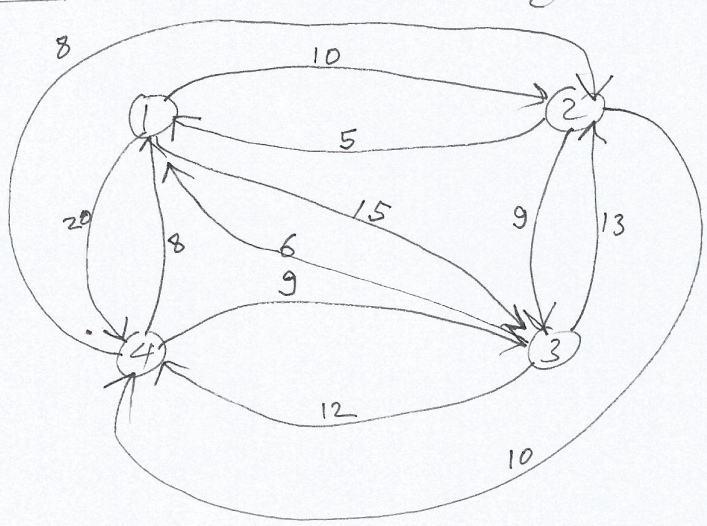
$$g(4,\{2,3\}) = \min\{6+g(2,\{3\}),\ 5+g(3,\{2\})\} = \min\{6+7,\ 5+5\} = 10$$

$g(2,\{3,4\})=11$ $\quad g(3,\{2,4\})=12$ $\quad g(4,\{2,3\})=10$

$$g(1,\{2,3,4\}) = \min\{1+g(2,\{3,4\}),\ 3+g(3,\{2,4\}),\ 2+g(4,\{2,3\})\}$$
$$= \min\{1+11,\ 3+12,\ 2+10\} = 12$$

We can find the tour by tracing back the minimum choice:

$g(1,\{2,3,4\}) \leftarrow g(2,\{3,4\}) \leftarrow g(3,\{4\}) \leftarrow g(4,\phi)$

The salesman tour is: $1 \to 2 \to 3 \to 4 \to 1$ having distance 12.

<u>Example 2</u>: Solve the following TSP instance: