The optimal choice of routing path
Mobile Internet Homework1

April 5, 2020

Abstract

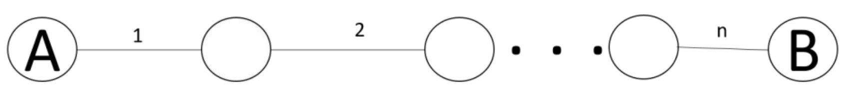
The optimal choice of routing path. I use greedy algorithm to solve this problem. And in the paper, I will give an easy-to-understand proof.

Key Words: Routing Path, Greedy Algorithm, Optimal Proof.

1 Problem

To focus more on the algorithm and method, I briefly summarize the description of the problem:

Considering there are n links between A and B, what we want to do is to find a method checking the the path between A and B with an optimal cost. We have the probability p_i and the cost c_i for each link i.



2 Algorithm

To solve this problem, I choose to use **greedy algorithm**.

The detailed method shows as follow:

1. Sort these links with the value $\frac{c_i}{1-p_i}$ as an ascending order.
2. Check these links as the sorted order one by one.

By this algorithm, we can have an optimal cost.

3 Proof

The following is the proof using exchange argument technique:

Considering there are two links i and j, which have the property that $\frac{c_i}{1-p_i} \leq \frac{c_j}{1-p_j}$.

From our algorithm, we will check i^{th} path, then wo check j^{th} path.

In the exchange argument technique, I will give the proof that if we check j before i, we will introduce another cost.

Probability and Cost Table

Exist Condition	Probability	Cost for Optimal Method	Cost for Exchange Method
i and j both exist	$p_i p_j$	$c_i + c_j$	$c_i + c_j$
i exists and j not exists	$p_i(1 - p_j)$	$c_i + c_j$	c_j
i not exists and j exist	$(1 - p_i)p_j$	c_i	$c_i + c_j$
i and j both not exist	$(1 - p_i)(1 - p_j)$	c_i	c_j

Set ΔC as the extra cost from exchangement, then we can have the following expression:

$$\Delta C = \sum Probability \cdot \Delta costForEachSmallCondtion \tag{1}$$

Introducing the concreate value:

$$\Delta C = \Delta C1 + \Delta C2 + \Delta C3 + \Delta C4 \tag{2}$$

$$\Delta C1 = p_i p_j \cdot [(c_i + c_j) - (c_i + c_j)] \tag{3}$$

$$\Delta C2 = p_i(1 - p_j) \cdot [(c_j) - (c_i + c_j)] \tag{4}$$

$$\Delta C3 = (1 - p_i)p_j \cdot [(c_i + c_j) - (c_i)] \tag{5}$$

$$\Delta C4 = (1 - p_i)(1 - p_j) \cdot [(c_j) - (c_i)] \tag{6}$$

Finally

$$\Delta C = c_j(1 - p_i) - c_i(1 - p_i) \tag{7}$$

By Condition $\frac{c_i}{1-p_i} \le \frac{c_j}{1-p_j}$, we have $\Delta C \ge 0$.

In this case, we can conclude that our optimal algorithm will not increase any cost comparing any other algorithm. It means it is the optimal algorithm.