EL7363 Communication Networks II:

Design and Algorithms

FINAL PROJECT REPORT

NEW YORK UNIVERSITY

TANDON SCHOOL OF ENGINEERING

TIANPEI CAI [tc1996@nyu.edu](mailto:tc1996@nyu.edu)

YUANZHI YAO yz1776@nyu.edu

**Part 1. Project Definition**

1.1 Problem Definition

This project tries to generate a solution to a common topic we have met through most of our network allocation and dimension problem – **Shortest Path Routing**. So finding out all the shortest paths for certain demand pair is an important step before we moving to next stage.

Our project aims to find out the shortest path for a given topology that generated by BRITE (https://www.cs.bu.edu/brite/), which can generates network topology randomly in any size.

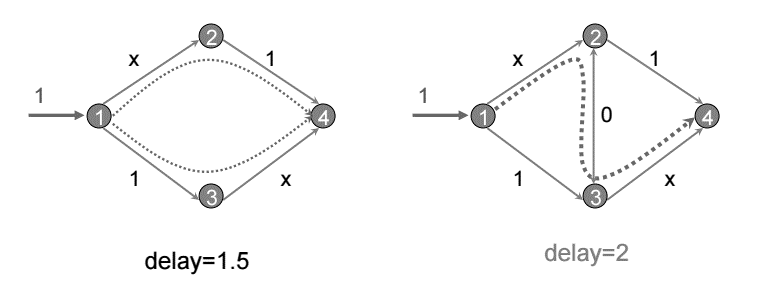
1.2 Motivation

In the first stage, we were trying to figure out a feasible solution to the Braess’ Paradox Problem by adding tax.

The Braess’ Paradox Problem can be define in short as:

*Adding extra capacity to a*[*network*](https://en.wikipedia.org/wiki/Network_(mathematics))*when the moving entities selfishly choose their route can in some cases reduce overall performance.*

This concept is somehow against human intuition. As showing below, after we add an ideal link to this network, the overall delay increasing.



Taking it into our daily life, a new high-way opened, people all want to leverage it, however, due to the selfish routing strategy, congestion became heavily on the connection roads. All of this problem is created by lacking of order. So we introducing a tax or say tolls – cars or flows have to pay a tax equals to the total delay (for all other cars or flows) that triggered by their joining. A paper (http://www.cs.cornell.edu/timr/papers/cost.pdf) points out that the taxes may not help this problem, however, our opinion is that it’s depend on how we treat the tax. In a common sense, tax will benefit for people in some patterns – opening more roads, transportation facilities. So how to set the rate of tax and how it effect this problem becomes our future topic. As mentioned above, this problem also need to figure out the question – which path to choose? For selfish routing, everyone wants to go through the shortest path, so how could we find out the shortest path in a large network would be the first problem we need to figure out.

1.3 Tools

Topology Generator: BRITE (<https://www.cs.bu.edu/brite/>)

Programming Language: Java

Collaborate Platform: GitHub (https://github.com/Tempay/ShortestPathFinding)

**Part 2. Design Models**

For shortest path routing, we considered the COST, which contains: propagation delays, unit costs, interface delays…, in our problem, the topology has already be defined by BRITE, so here we considered about the propagation delay, and in future when we deal with the Braess Paradox, the arbitrary tax would also be considered.

***LP:* D/SDP Link-Path Formulation**

**Simple Design Problem**

**indices**

*d* = 1*,* 2*, ...,D* demands

*p* = 1*,* 2*, ..., Pd* candidate paths for flows realizing demand *d*

*e* = 1*,* 2*, ...,E* links

**constants**

*δedp* = 1, if link *e* belongs to path *p* realizing demand *d*; 0, otherwise

*hd* volume of demand *d*

*ξe* unit (marginal) cost of link *e*

**variables**

*xdp* flow allocated to path *p* of demand *d* (continuous non-negative)

*ye* capacity of link *e* (continuous non-negative)

**objective**

minimize ***F*** = \_*eξeye* (bandwidth cost) (4.1.1a)

**constraints**

\_*pxdp* = *hd, d*= 1*,* 2*, ...,D* (demand constraints) (4.1.1b)

\_*d*\_*pδedpxdp ≤ ye, e*= 1*,* 2*, ...,E* (capacity constraints). (4.1.1c)

**Part 3. Solution Techniques**

**Part 4. Performance Evaluation**