ITS Assignment 2 description

Under supervision of

Dr. Zahra Amini

Teaching Assistants:

Ehsan Hassanzadeh

Armin Sahebi

November 2021

In this problem, you will be required to design an algorithm for finding the shortest path between two points with specific criteria. Python is the recommended programming language for this assignment.

Part 1

This part aims to reach node 8 from node S in the network shown in Fig. 1 with minimum cost (minimum travel time). In this network, all links are in both directions, and numbers above each link denote the link's travel time (seconds).

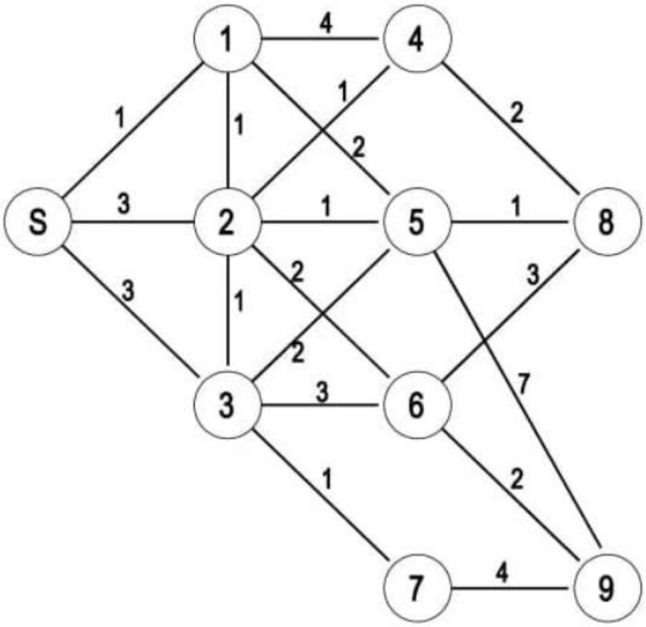

Fig. 1: Sample Network

- **Shortest Path Algorithm:**
  1. Design an algorithm for finding the shortest path and describe the logic behind it. (Describe your assumptions and required steps)
  2. Plot the shortest path from node S to node 8 using your designed algorithm and plot each step.
  3. Plot the search tree graph for the shortest path.
  4. Calculate the total cost for the shortest path.

- **Incremental Traffic Assignment:**
  1. Assume that the travel time for each link is calculated using the equation below:

$$T_f = T_0(1 + 2\left[\frac{V}{40}\right]^2)$$

Where:

T_f = Final link travel time
T_0 = Initial link travel time in free-flow mode (travel times in Fig. 1)
V = Assigned flow to the link

2. Given that 100 vehicles are about to travel from node S to node 8 on 20 vehicle increments, modify your shortest path algorithm to update links' travel times after each increment.
    - In other words, you should find the shortest path for 20 vehicles and update links travel times after the first step assignment. Then, you should find the new shortest path based on the updated travel times and assign 20 more vehicles to the new shortest path. Continue these steps to assign all 100 vehicles from node S to node 8
    - Each link has a **capacity of 40 vehicles**. Consider this limitation in traffic assignment.

3. Explain required modifications and describe each step.
4. Plot the shortest path in each step and plot the search tree.
5. Calculate the total cost in (*veh*s)*

Part 2.

In this part, you ought to write Python code for your algorithm in part 1. Same as the previous part, you have to write a script for finding the shortest path in the dataset network and then assign traffic flow incrementally.

- **Dataset**

  The dataset is a text file named "Network.txt," including the network information. Each row of the dataset represents a link in the network containing start node, end node, and travel time (seconds), respectively. Table 1 shows sample data from the dataset.

Table 1: Links data sample

FromNodeId	ToNodeId	TravelTime(s)
0	1	376
0	2	486
0	29	401
1	0	298
1	23	249
1	32	145
2	0	320
2	26	299
2	34	189

Tip: This network has one-way links with different initial travel times. As a result, each node may have several outputs and inputs. In contrast, some nodes may have no outputs or no inputs. These scenarios are shown in Fig. 2.

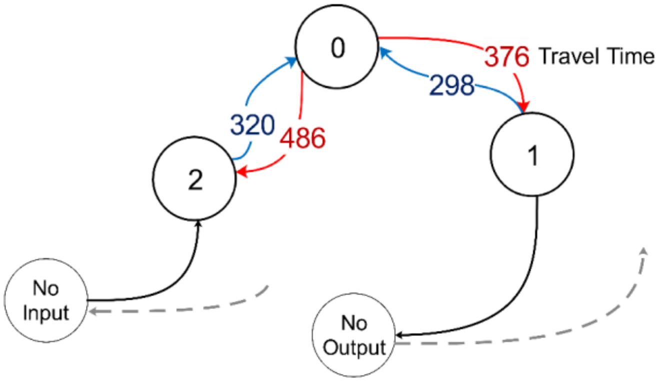

Fig. 2: Different scenarios for network's nodes

- **Incremental Traffic Assignment:**

1. Assume that the travel time for each link is calculated using the equation below:

$$T_f = T_0(1 + 0.3\left[\frac{V}{4000}\right]^2)$$

Where:

T_f = Final link travel time
T_0 = Initial link travel time in free-flow mode (travel times in the dataset)
V = Assigned flow to the link

2. Find the source node and sink node from Table 2.

Table 2: Centroid Coordinates

Last digit of Student ID	Source Node	Sink Node
1	723	6856
2	2148	10123
3	985	3654
7	68	6575
8	3859	11
9	357	4896

3. Given that 8000 vehicles are about to travel from the source node to the sink node on 1000 vehicle increments, modify your shortest path algorithm to update links' travel times after each increment. Note that each link has a **capacity of 4000 vehicles**.

4. Calculate the total travel time for accommodating all 8000 vehicles to the shortest paths in ***veh*hr***. (e.g., Travel Time = 5000 veh*hr)

5. Find the order of the shortest paths' nodes in each increment (e.g., 0→1→2→3→4)

6. Find the number of nodes in the shortest paths' nodes in each increment (e.g., 0→1→2→3→4: 5 Nodes)

7. Calculate the algorithm running time.

Important Points:

1. You have to write your own script for the shortest path algorithm. Do not use shortest path algorithms from Python libraries.

2. You should submit a PDF report file (including required descriptions in each part, a description of your Python script, and final results), and your Python script.

3. Please write comments in your script for each part of the code.