Public Transport as a Network Flow Problem
Mathematics 441 – portfolio object 2

Mauricio Soroco
60785250

September 2022

Contents

1 Problem Description **2**

2 Problem Formulation **3**
2.1 Graph Visualization 3
2.2 Network Flow Problem Formulation 4
2.3 Integrality Theorem 4
2.4 Linear Programming Formulation of Network Flow 4

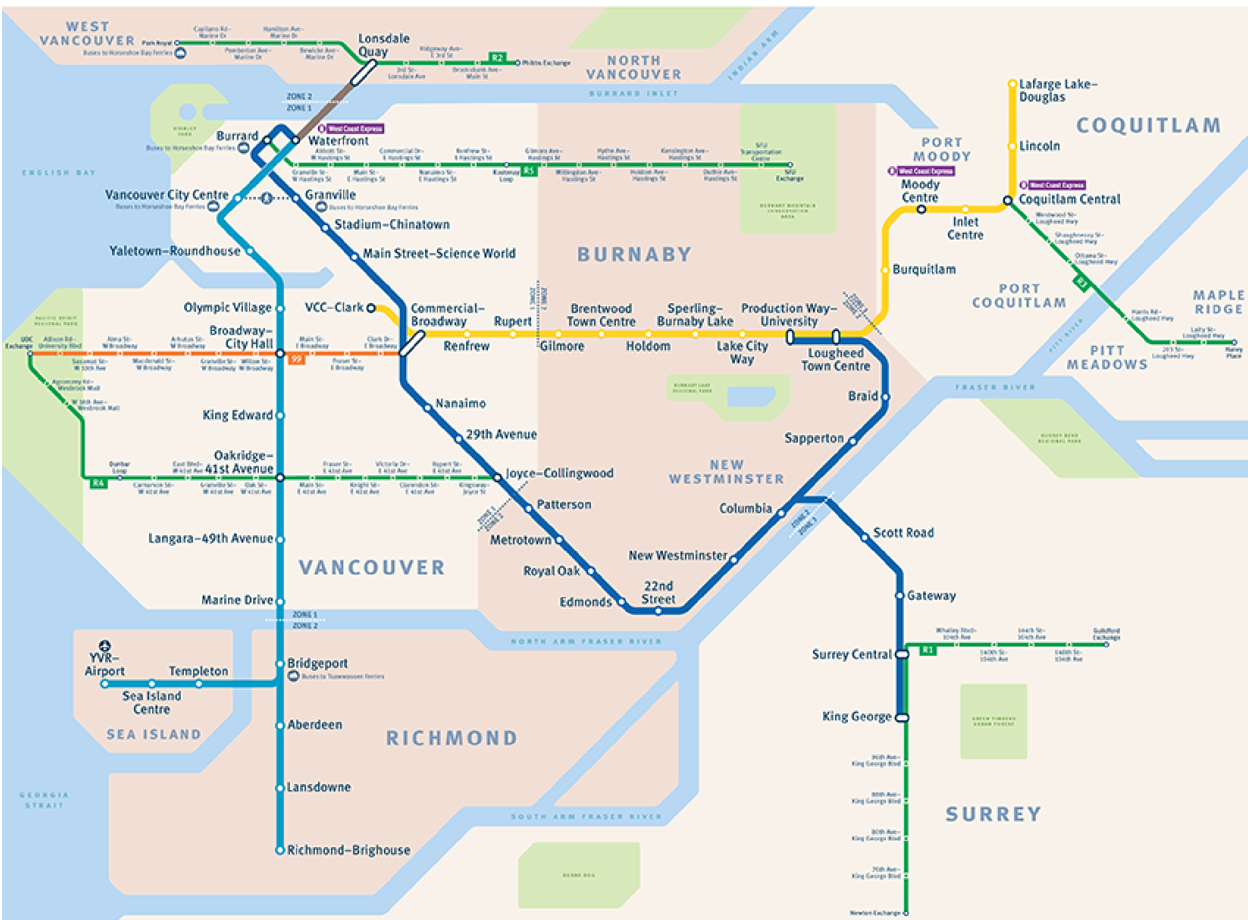
3 Final Words and Further Problems **7**

1 Problem Description

Public transport involves passengers trying to commute around the lower mainland on buses, trains and boats that arrive at set intervals and travel along a known route. We will formulate the problem of determining how to allocate an efficient amount of transportation at a particular time, minimizing costs to the bus company, to transport people from a common starting destination to their end destinations (or vice versa) given a choice of multiple bus routes.

Suppose in this problem that UBC students are wanting to go to their homes after a day of school.

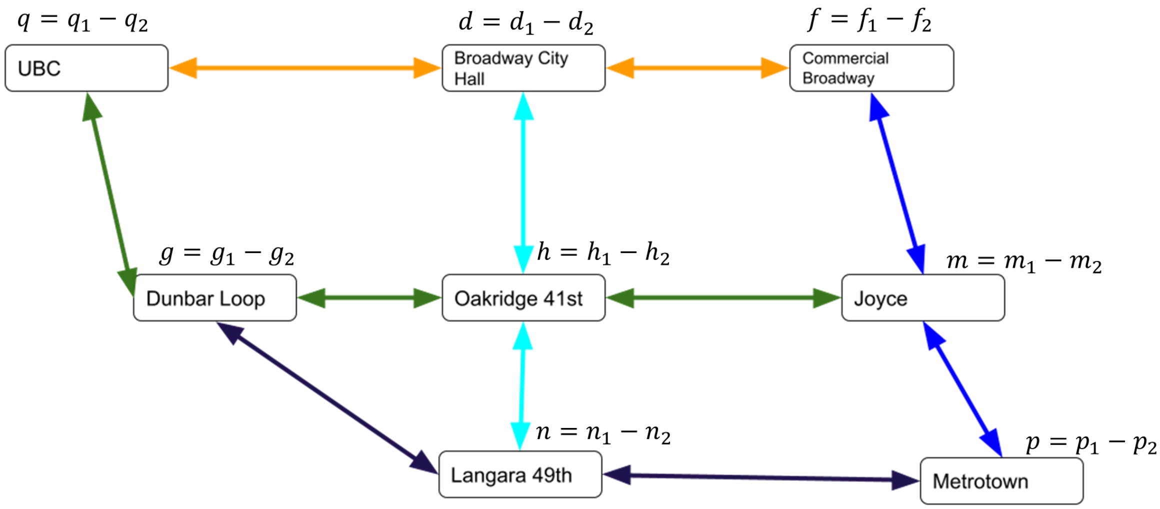
We formulate an example problem based on some popular transit routes.



2 Problem Formulation

2.1 Graph Visualization

Consider the following graph representing a simplified model of the transportation problem. Below we show a graph of possible transit routes:



$$G = (V, E)$$

Where:

The stations "UBC", "Dunbar Loop", etc $\in V$.
The bus/train routes e.g., $\langle$ "UBC" ,"Dunbar Loop" $\rangle \in E$.

Each node above represents a station (in this case for either busses or trains or both) and each edge represents the route that at least one bus or train may travel. For example, the Canada line (in light blue) travels from Langara to Oakridge and then eventually reaches Broadway City Hall. Due to the directed arrows, they may similarly travel along the reverse direction. At each stop (node) there is a certain quantity of travelers that will either board or exit the vehicle. The number of travelers boarding is represented by θ_1 and the number of travellers exiting is represented by θ_2. Hence the net number of passengers boarding a vehicle is given by:

$$\theta = \theta_1 - \theta_2 = (\text{\# boarding})\text{-}(\text{\# exiting}) \tag{1}$$

Note that the θ_i are positive magnitudes while the net change of θ can be either positive (to indicate a supply—i.e. passengers wanting to board) or can be negative (to indicate a demand—i.e. passengers wanting to disembark). For example, at 2 pm of a weekday we can expect more students leaving UBC than arriving, hence $q > 0$ and $q_1 > q_2$. On the flip side, at 8 am when a large proportion of students are arriving to UBC, we would expect $q < 0$ and $q_1 < q_2$.

2.2 Network Flow Problem Formulation

One of the assumptions for a feasible solution of the network flow problem is:

$$\sum_{i \in V} b_i = 0 \tag{2}$$

Hence, in our toy model:

$$q + d + f + g + h + m + n + p = 0 \tag{3}$$

For each route (edge) there is a cost of transporting a passenger from point i to point j. There are a few possible cost functions. Some examples may involve letting c_{ij} take into account the revenue per passenger, the distance, and the cost associated with time it takes travel (to account for traffic etc.). For example, we could let $c_{ij} < 0$ be the average revenue per passenger for boarding the vehicle traveling from node i to j.Thus by minimizing the cost, we are maximizing revenue for the transit company.

2.3 Integrality Theorem

One important constraint for this problem is that the solution presented be purely integers. This is because we cannot have half a fraction of a student going to point A and another fraction of a student going to point B. For this we apply the Integrality Theorem which states: "For a network flow problem with integral data, every basic feasible solution and in particular every basic optimal solution assigns integer flows to every arc" (Vanderbei p 247) Hence as a consequence of this theorem, the costs c_{ij} must be all integers (note that b and A are already integral).

2.4 Linear Programming Formulation of Network Flow

A network flow is a matrix $X = [x_{ij}]$ where x_{ij} is the amount of passengers who travel from node i to node j. The balance equation at each node specifies that the number of passengers who board the vehicle at station k minus the number of passengers who disembark from the vehicle at station k equals the supply/demand. Hence the balance equation at node n is:

$$\sum_i x_{ik} - \sum_j x_{kj} = -b_k \tag{4}$$

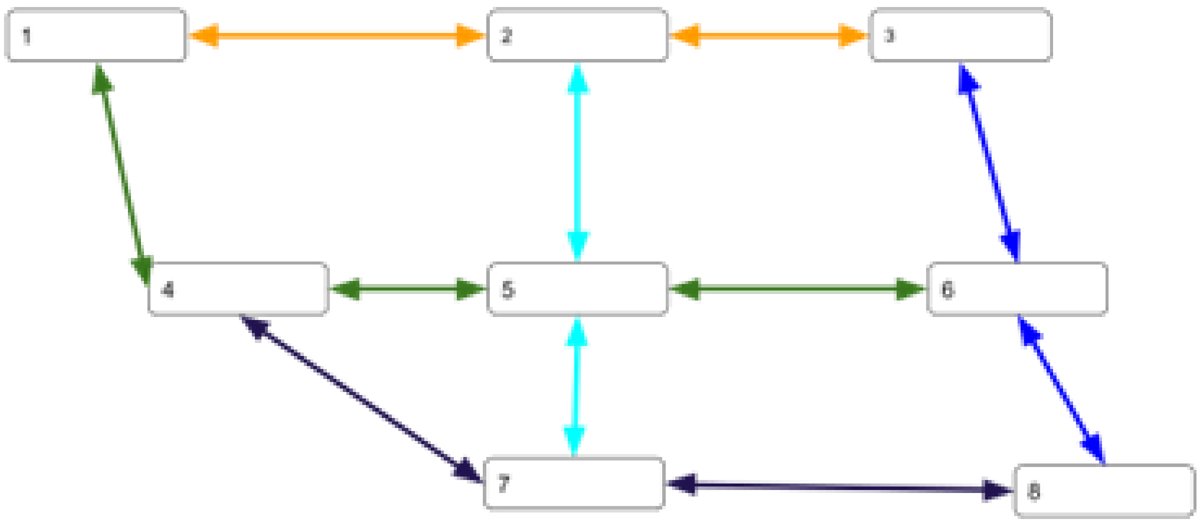
For example, in our toy model above we have the balance equation for UBC:

$$x_{\text{City Hall, UBC}} + x_{\text{Dunbar , UBC}} - x_{\text{UBC ,City Hall}} - x_{\text{UBC ,Dunbar}} = -q \tag{5}$$

We can encode the information in equation 5 for all balance equations of our model into a matrix vector product as follows:

$$Ax = -b \tag{6}$$

Where row k of $Ax = -b$ is in the form of equation 4. Since there are 11 bidirectional edges in our model, there are $2 \times 11 = 22$ flows, so that $\vec{x} \in Z^(22 \times 1)$ and x_{ij} represents flow from node i to j. Furthermore, there are 8 supply and demand constraints (like equation 4) such that $A \in Z^(8 \times 22)$ and hence $b \in Z^(8 \times 1)$.



We use the notation introduced in the graph above to avoid the repetition of the long names. The matrix vector product becomes: $Ax = -b$

$$\begin{bmatrix} -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{23} \\ x_{14} \\ x_{25} \\ x_{36} \\ x_{45} \\ x_{56} \\ x_{47} \\ x_{57} \\ x_{68} \\ x_{78} \\ x_{21} \\ x_{32} \\ x_{41} \\ x_{52} \\ x_{63} \\ x_{54} \\ x_{65} \\ x_{74} \\ x_{76} \\ x_{86} \\ x_{87} \end{bmatrix} = \begin{bmatrix} -q \\ -d \\ -f \\ -g \\ -h \\ -m \\ -n \\ -p \end{bmatrix}$$

Now the objective function is:

$$z = \vec{c} \cdot \vec{x} \tag{7}$$

Where $\vec{c} \in Z^(22 \times 1)$ and c_{ij} represents cost of one passenger traveling from node i to j. Since we want to minimize the cost, we get the final optimization problem:

$$\min_{Ax=-b, \quad x \geq 0} \vec{c} \cdot \vec{x} \tag{8}$$

For which we can then convert to standard inequality form to solve with the simplex method.

The spanning tree solution will indicate which subset of busses/trains need to run at a particular time to transport the students from UBC to their homes in the most cost-effective way possible.

3 Final Words and Further Problems

We were able to formulate public (student) transportation as a network flow problem. The reason we can't do multiple starting destinations and multiple end destinations at the same time is that the solution will find a way to send passengers to locations that satisfy that in total the right amount of people got to the right address, but it does not consider where they came from. Hence, we restrict the starting destination to one location (ex students wanting to get home from UBC). For example, if we said x number of students are leaving UBC and y are leaving SFU, and p students are going to Richmond, and q students are going to Burnaby, the solution won't take into account a possible restriction that only 50% of the UBC students are going to Richmond and the rest are going to Burnaby. It might find that the optimal solution is to send all the UBC students to Richmond and all the SFU students to Burnaby, which would not satisfy our problem. Finding a solution to this problem that takes into account these constraints is an interesting generalization of the problem I presented above, to be explored.