CE 272
Traffic Network Equilibrium

Lecture 23

Algorithm B - Part I

Previously on Traffic Network Equilibrium...

In the gradient projection method,

$$-\frac{g(0)}{g'(0)} = \frac{\tau_p - \tau_{p^*}}{\sum_{(i,j)\in\hat{A}} t'_{ij}(x_{ij})}$$

However, this flow shift may result in negative y_p. Hence, perform a projection step by setting

$$\Delta y = \min\left\{y_p, \frac{\tau_p - \tau_{p^*}}{\sum_{(i,j)\in\hat{A}} t'_{ij}(x_{ij})}\right\}$$

Previously on Traffic Network Equilibrium...

We saw that the equilibrium solutions satisfy two interesting properties.

1. The equilibrium OD flow cannot be present on both sides of a two-way street.
2. The ratio of flows on any two routes between an OD pair is independent of the OD demand. Further, for a given PAS, the ratio of flows on the two segments is same across all OD pairs.

Previously on Traffic Network Equilibrium...

To topologically order nodes, define the indegree of a node as the number of arcs coming into it. Consider a node i with zero indegree. Set $\vartheta_i = 1$.

Delete the arcs emanating from i. Pick a new node with indegree zero and set it's topological order to 2.

Repeat this procedure until there are no nodes with zero indegree. If there are leftover nodes, then the network is acyclic.

Previously on Traffic Network Equilibrium...

In topological ordering, we essentially delete arcs and update the indegrees of nodes. Hence, the complexity of topological ordering is $O(m)$.

To find the shortest path in a DAG, we simply scan nodes in increasing topological order and update the labels of the downstream arcs.

In this version, we do not find a node with minimum distance label nor do we keep track of a SEL. Each arc is scanned at most once. Hence, the complexity of this method is $O(m)$.

Lecture Outline

1. Introduction
2. Equilibrate Bushes
3. Optimize Bushes

Lecture Outline

Introduction

Bush-based or origin-based methods are ones in which we keep track of the set of "used arcs" for each origin.

These arcs can be identified by superposing all used paths, which results in an acyclic connected graph, i.e., a bush.

Bush-based methods essentially solve the following version of the TAP. The decision variables are x_{ij}^r, the link flows segregated by origins.

$$\min \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(\omega)\, d\omega$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} x_{ij}^r - \sum_{h:(h,i)\in A} x_{hi}^r = \begin{cases} \sum_{s\in Z} d_{is} & \text{if } i = r \\ -d_{ri} & \text{if } i = s \\ 0 & \text{otherwise} \end{cases} \quad \forall\, r \in Z$$

$$x_{ij} = \sum_{r\in Z} x_{ij}^r \,\forall\,(i,j) \in A$$

$$x_{ij}^r \geq 0 \,\forall\,(i,j) \in A, r \in Z$$

We will discuss two main components of the algorithm today and put these pieces together in the next class.

For now, we will look at the bushes for a single origin, while keeping the flows from other origins to other destinations fixed.

In other words, the flows from other OD pairs are treated as "background flows". They will be modified in different iterations of Algorithm B.

Lecture Outline

Equilibrate Bushes

Equilibrate Bushes

Introduction

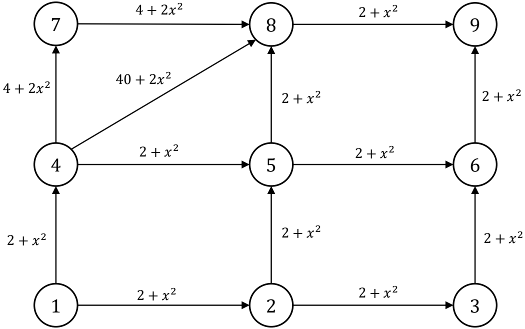
In the 'Equilibrate Bush' step, we try to satisfy the Wardrop principle for a **given origin and all destinations** by routing flows along different paths **only using the Bush arcs**.

The flow shifts are similar to gradient projection where we move travelers from longest to shortest PASs using Newton's method.

Equilibrate Bushes

Example

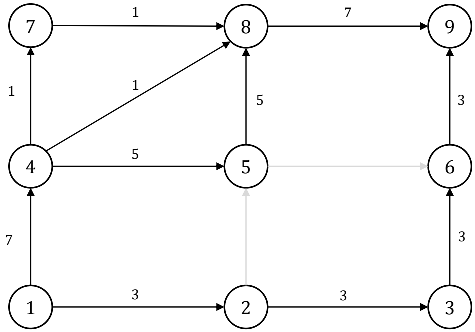
Copy the following example, we will use this to illustrate how bushes are equilibrated. Imagine there are 10 traveler from 1 to 9.



Equilibrate Bushes

Example

Suppose at an intermediate iteration, we get a bush with origin-based link flows as shown in the following figure.



Equilibrate Bushes

Example

Carry out the following steps on the 9 node network.

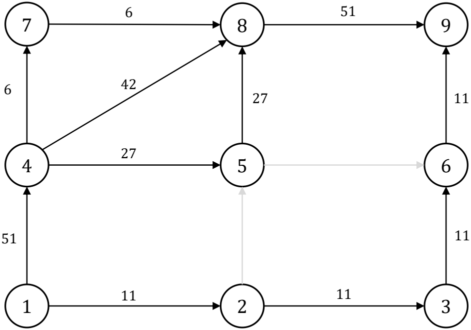
1. Compute link travel times
2. Compute link travel time derivatives
3. Find a topological ordering of the bush
4. Solve the shortest path problem
5. Solve the longest path problem

For the last two steps, find the distance and predecessor labels.

Equilibrate Bushes

Example

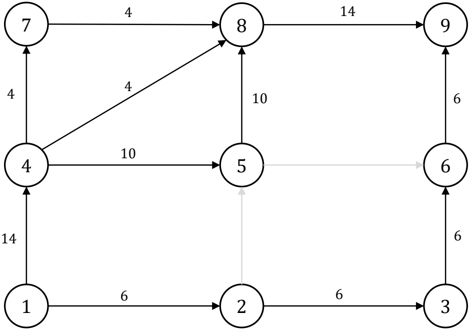
1. **Link travel times:**



Equilibrate Bushes

Example

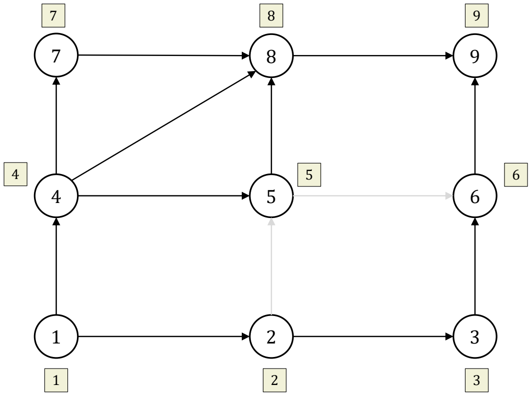
2. Link derivatives:



Equilibrate Bushes

Example

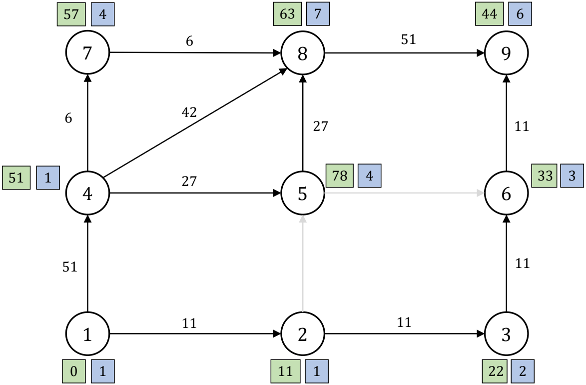
3. Topological Ordering:



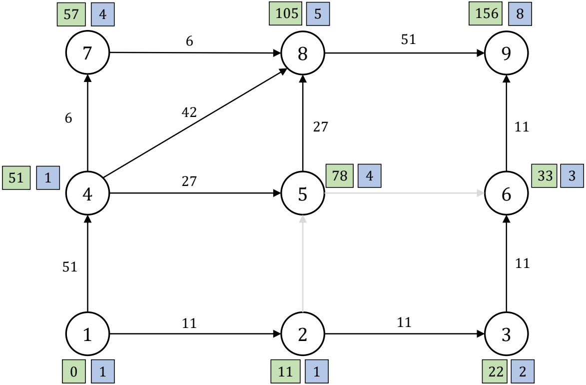
Note: There could be multiple ways to topologically order nodes (it dosen't matter which one you pick).

Equilibrate Bushes
Example

4.Shortest Path Problem:



5. Longest Path Problem:



Equilibrate Bushes

Pseudocode

After finding the shortest and longest path labels and predecessors, the following steps are carried out.

1. Pick a node j with the highest topological order.
2. Use the shortest and longest paths to j to identify the last common node or divergence node i.
3. Determine the PASs using the shortest and longest sub-paths $\underline{P}_{ij}$ and $\overline{P}_{ij}$.
4. Calculate flow shift using Newton's method.
5. Update origin-based flows for the bush.
6. Set j to the previous node topologically and repeat from Step 2 until the origin r is reached.

Equilibrate Bushes

Newton's Method

Let j represent the last common node.

$$\Delta y = \frac{(\nu_j^r - \nu_i^r) - (\mu_j^r - \mu_i^r)}{\sum_{(k,l) \in \underline{P}_{ij} \cup \overline{P}_{ij}} t'_{kl}}$$

However, we also need to ensure that the flow shift does not result in negative link flows. Hence, we impose

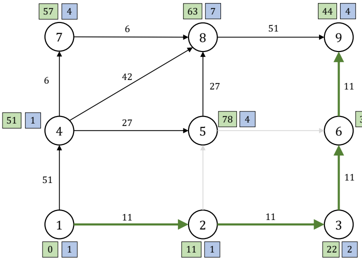
$$\Delta y \leq \min_{(k,l) \in \overline{P}_{ij}} x_{kl}^r$$

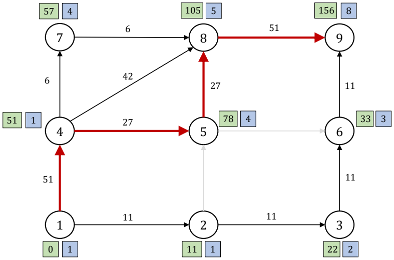
One can do multiple iterations of Newton's method (without resolving for the shortest and longest paths) to make the travel times as close as possible.

Equilibrate Bushes

Example

Iteration 1: $j = 9$





The shortest and longest paths to 9 are 1-2-3-6-9 and 1-4-5-8-9. Hence, the last common node is $i = 1$.

Equilibrate Bushes
Example

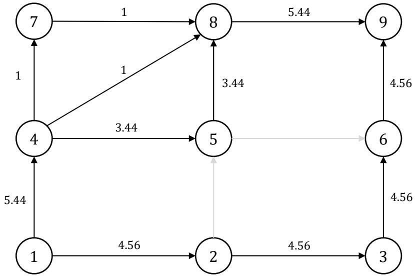
Iteration 1: $j = 9$

The PASs between i and j are

$$\underline{P}_{ij} = \text{1-2-3-6-9}$$
$$\overline{P}_{ij} = \text{1-4-5-8-9}$$

and the amount of flow to be shifted between them is

$$\min\left\{5, \frac{156 - 44}{72}\right\} = 1.56$$

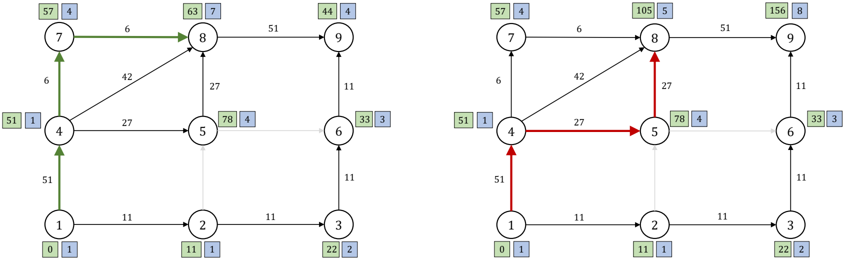


We could update travel times, derivatives, and the shortest and longest paths, but it is not needed for convergence. To save time, we will do that only after we examine all nodes j in reverse topological order.

Equilibrate Bushes

Example

Iteration 2: $j = 8$



The shortest and longest paths to 8 are 1-4-7-8 and 1-4-5-8. Hence, the last common node is $i = 4$.

Equilibrate Bushes

Example

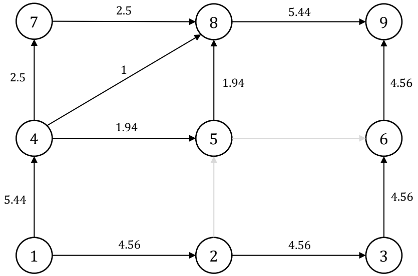
Iteration 2: $j = 8$

The PASs between i and j are

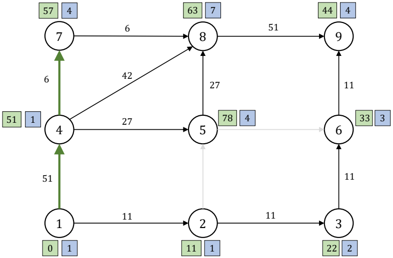
$$\underline{P}_{ij} = 4\text{-}7\text{-}8$$
$$\overline{P}_{ij} = 4\text{-}5\text{-}8$$

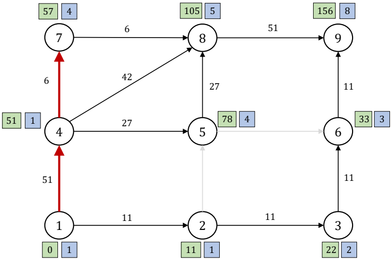
and the amount of flow to be shifted between them is

$$\min\left\{3.44, \frac{54-12}{28}\right\} = 1.5$$



Iteration 3: $j = 7$





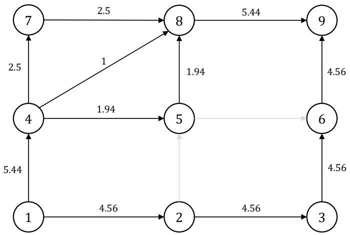
The shortest and longest paths to 7 are 1-4-7 and 1-4-7. Hence, the last common node is $i = 7$. Hence, no flow shift is necessary in this case.

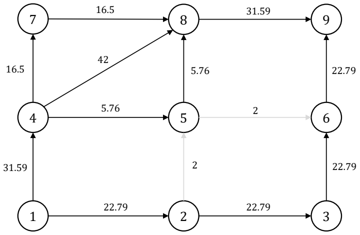
Equilibrate Bushes

Example

Proceeding similarly, we notice that the shortest and longest paths to all other nodes with lower topological order are the same. Therefore, the link flows won't change.

The new flows (left) and travel times (right) at the end of one 'Equilibrate Bush' step are





Lecture Outline

Optimize Bushes

Optimize Bushes

Introduction

After shifting flows in 'Equilibrate Bush', we get an ϵ-equilibrium bush. We can then update the link travel times for the new flows as seen earlier.

In the 'Optimize Bush' step, we try to update the bush by including shorter paths that are not a part of the current bush.

This step is carried out by

- Removing unused arcs
- Adding new shortcuts

Optimize Bushes

Removing Unused Arcs

Flow shifts in ‘Equilibrate Bush’ can result in zero flows on certain bush links.

We can delete these links unless they are needed to ensure connectivity.

Optimize Bushes

Adding New Shortcuts

To add links that are on new shortest paths, we have two options.

We could resolve the shortest path problem on the current bush B^r, and check for non-bush links $(i, j) \notin B^r$ that satisfy $\mu_j > \mu_i + t_{ij}$.

The only issue though is that this method may create cycles if B^r is not fully equilibrated!

Optimize Bushes

Adding New Shortcuts

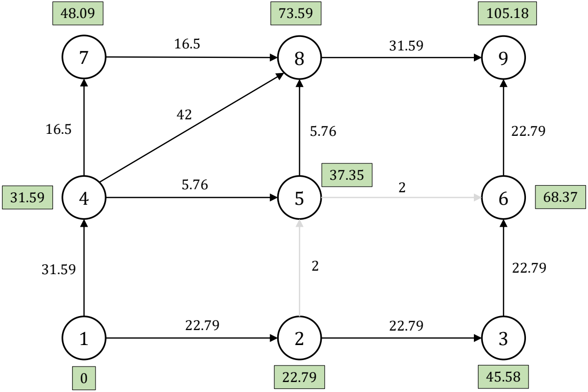
The second option is to use the longest path labels instead of the shortest paths.

A non-bush link $(i, j) \notin B^r$ is added to the bush only if $\nu_j > \nu_i + t_{ij}$.

It is easy to show that this approach does not induce cycles even if the bush B^r isn't fully equilibrated.

Optimize Bushes
Example

Find the new bush arcs in the following network using the longest path labels.



Your Moment of Zen