Bush-based algorithms for solving traffic assignment

CE 392C

OUTLINE

Recall that *link-based methods* require very little memory, but are also slow to converge.

By contrast, *path-based methods* are much faster, but can potentially require a large amount of memory.

The Chicago regional network has roughly 90 million equilibrium paths; each path contains several dozen links on average.

Bush-based methods involve substantially less memory than path-based methods, and don't seem to be any slower. Some evidence even suggests that they are faster.

Caveat: It is relatively difficult to compare the most advanced algorithms; much of their performance depends on subtle implementation details.

Remember that a bush is a acyclic subnetwork in which every node can be reached from the root.

At the equilibrium solution, *the set of all paths used by a given origin forms a bush.* (Why?)

In a bush-based algorithm, we maintain a set of bushes, one for each origin. We have two goals:

1. Find the *right bush* for each origin (containing all of the paths used at equilibrium).
2. Find the *link flows* on the bush links which satisfy the equilibrium principle.

Origins are only allowed to place flow on links in the bush.

All bush-based algorithms perform the following steps:

1. Create an initial bush for each origin (easy way: start with shortest paths using free-flow times)
2. Shift flows within each bush to move it closer to equilibrium.
3. Update the bushes to remove unused links, and to add links which provide shorter paths.

Step 2 is where most bush-based algorithms differ.

Some bush-based algorithms:

- Origin-based assignment (Bar-Gera, 1999–2000; Nie, 2009)
- Algorithm B (Dial, 1999–2006)
- LUCE (Gentile, 2009)
- TAPAS (Bar-Gera, 2011)

ALGORITHM B

Algorithm B adopts the following simple rule for step 2:

For each destination, find the *longest used path* from the origin as well as the *shortest path* on the bush. Use Newton's method to shift flow between these paths to move towards equilibrium.

The Newton shift works in the same way as for gradient projection, but the bush structure makes it easier to find longest and shortest paths.

This relies on the acyclic property of bushes: longest and shortest paths can be found by simply going over the network in topological order.

By contrast, if there are cycles, shortest path is a bit harder, and finding longest paths is *much* harder.

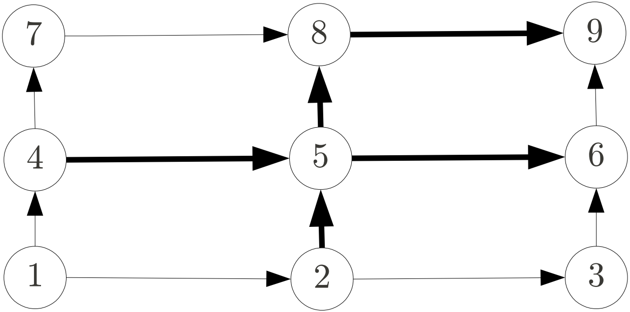
In step 3, bushes are updated using the following rules:

- If a bush link has zero flow, it is removed (unless doing so would disconnect a node from the root)
- If a non-bush link "provides a shortcut", it is added.

Specifically, if L_i is the travel time on the shortest path to node i *only using bush links*, a link (i, j) is a shortcut if $L_i + t_{ij} < L_j$.

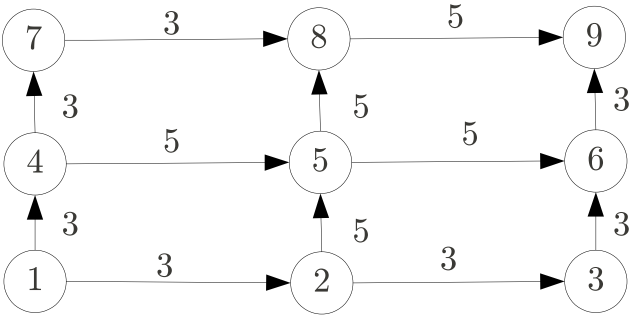
Will adding links according to this rule create a cycle?

COMPONENTS

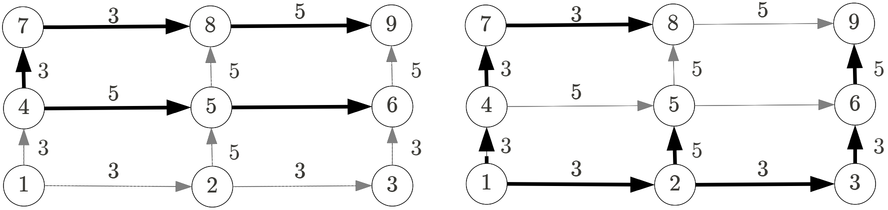


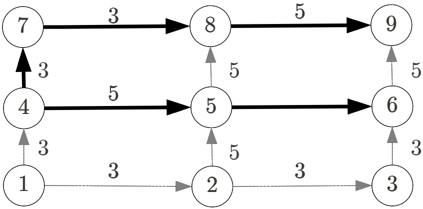
$t_{ij} = 3 + (x_{ij}/200)^2$ on thin links
$t_{ij} = 5 + (x_{ij}/100)^2$ on thin links
OD matrix: $d^{49} = 1000$, $d^{49} = 1000$

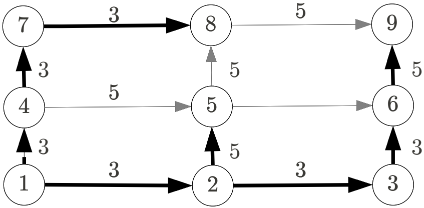
Free flow travel times



Initial bushes

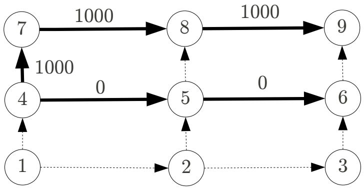


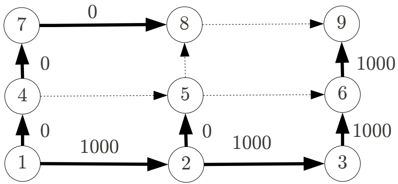




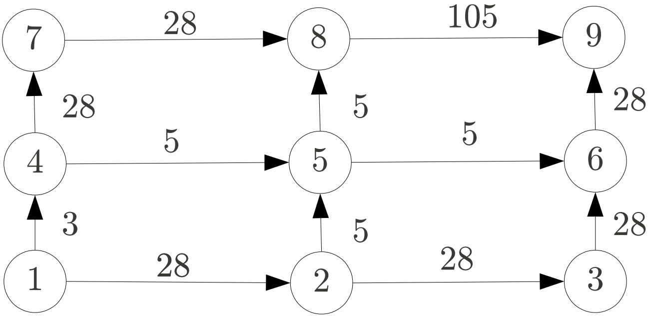
These bushes are at equilibrium because there is only one path from each origin to the destination.

Initial link flows

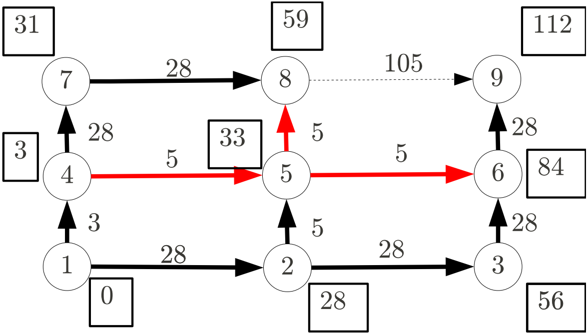




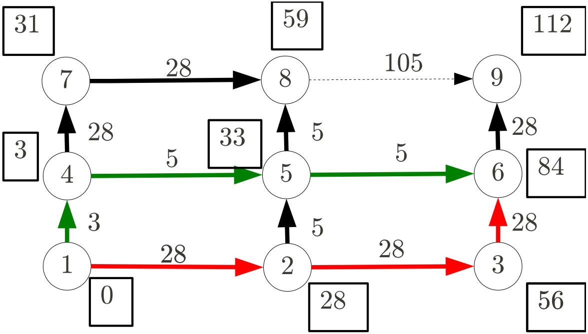
Travel times



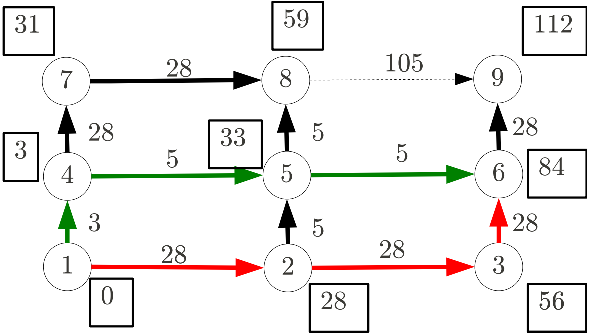
Add shortcuts to the bush for origin 1



Identify shortest and longest paths to destination 9



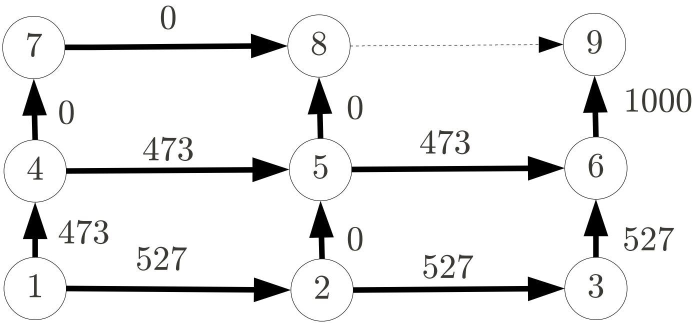
The pair of alternate segments is $\{[1, 4, 5, 6], [1, 2, 3, 6]\}$.



Flow on $[1, 4, 5, 6]$: 0
Flow on $[1, 2, 3, 4]$: 1000
Travel time on $[1, 4, 5, 6]$: 13
Travel time on $[1, 2, 3, 4]$: 84
Derivative on $[1, 4, 5, 6]$: 0
Derivative on $[1, 4, 5, 6]$: 0.15

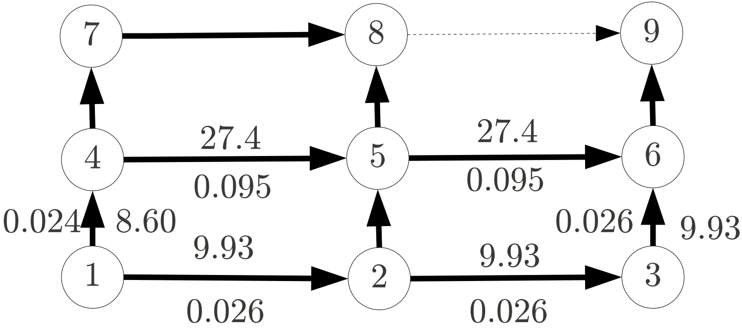
Newton shift is given by

$$\frac{84 - 13}{0 + 0.15} = 473$$



Newton shift is given by

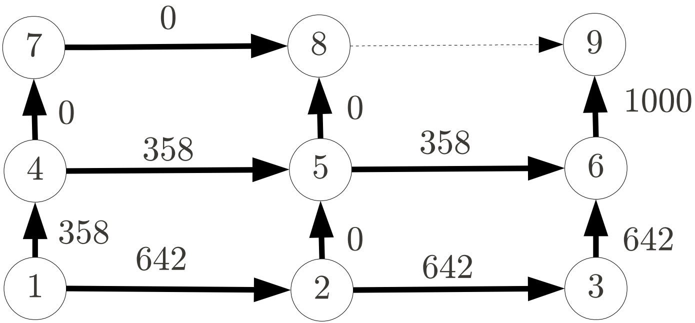
$$\frac{84-13}{0+0.15}=473$$



Flow on $[1, 4, 5, 6]$: 473
Flow on $[1, 2, 3, 4]$: 527
Travel time on $[1, 4, 5, 6]$: 63.4
Travel time on $[1, 2, 3, 4]$: 29.8
Derivative on $[1, 4, 5, 6]$: 0.213
Derivative on $[1, 4, 5, 6]$: 0.079

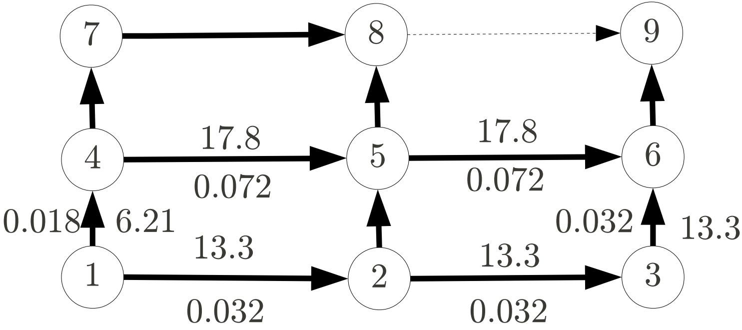
Newton shift is given by

$$\frac{63.4 - 29.8}{0.079 + 0.213} = 115$$



Newton shift is given by

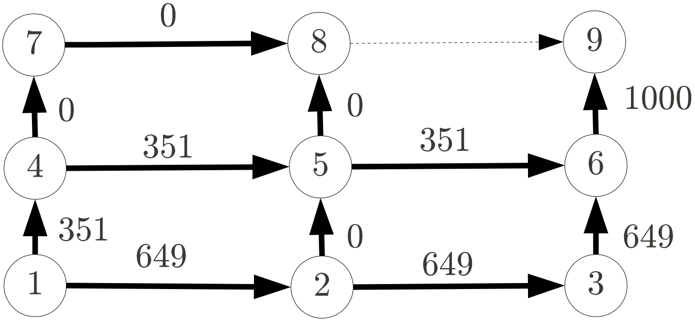
$$\frac{63.4 - 29.8}{0.079 + 0.213} = 115$$



Flow on $[1, 4, 5, 6]$: 358
Flow on $[1, 2, 3, 4]$: 642
Travel time on $[1, 4, 5, 6]$: 41.9
Travel time on $[1, 2, 3, 4]$: 39.9
Derivative on $[1, 4, 5, 6]$: 0.161
Derivative on $[1, 4, 5, 6]$: 0.097

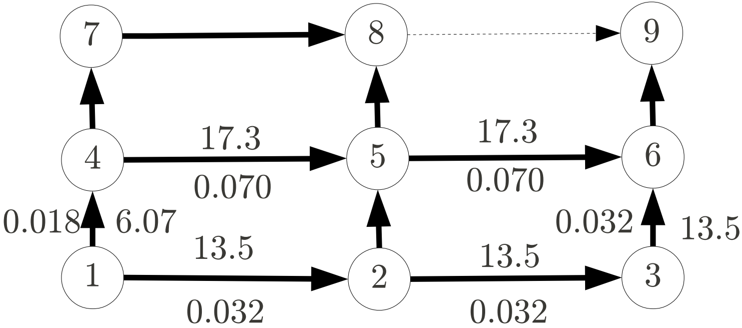
Newton shift is given by

$$\frac{41.9 - 39.9}{0.161 + 0.097} = 7.71$$



Newton shift is given by

$$\frac{41.9 - 39.9}{0.161 + 0.097} = 7.71$$



Flow on $[1, 4, 5, 6]$: 350
Flow on $[1, 2, 3, 4]$: 650
Travel time on $[1, 4, 5, 6]$: 40.64
Travel time on $[1, 2, 3, 4]$: 40.65
Derivative on $[1, 4, 5, 6]$: 0.158
Derivative on $[1, 4, 5, 6]$: 0.097

Newton shift is 0.034, close enough to equilibrium.

DEMONSTRATION

COMPLETE ALGORITHM SPECIFICATION

1. Create an initial bush for each origin (easy way: start with shortest paths using free-flow times)
2. Shift flows within each bush to move it closer to equilibrium: move flows from longest paths to shortest ones using Newton's method.
3. Update the bushes to remove unused links, and to add links which provide shorter paths.

The second and third steps are easier if we first calculated several labels for bush links and nodes.

BUSH LABELS

Bush labels

As with Dial's method for stochastic network loading, we use x_{ij}^r to denote the flow on each link, and L_i^r to denote the travel time on the shortest path from the origin r to node i.

A new set of labels are U_i^r, denoting the travel time on the *longest used path* from the origin r to node i.

We already know how to find shortest paths: how do we find longest paths?

Since a bush is acyclic, we can modify the shortest path algorithm to create a "longest path" algorithm:

1. Initialize by setting $U_i^r \leftarrow \infty$ and $\bar{q}_i^r \leftarrow -1 \; \forall i \in N$, and set $U_r^r \leftarrow 0$
2. Let i be the node topologically following r.
3. (*By this point, we have found the longest path from r to all nodes topologically before i*)
4. Find the longest used path from r to i by looking at each of the tail nodes one could arrive from:

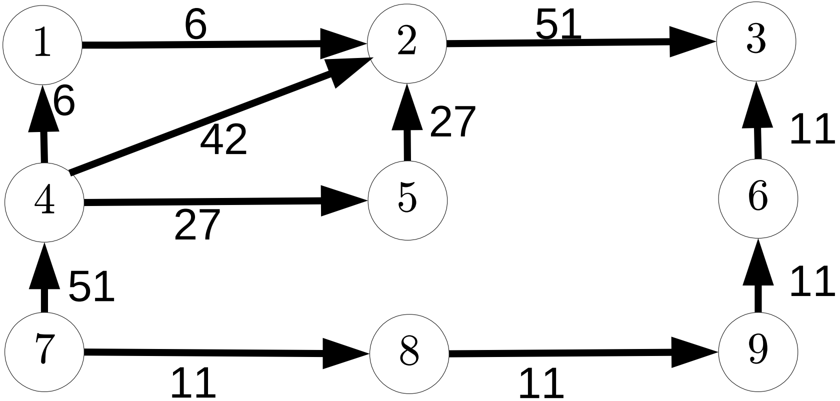
$$U_i^r = \max_{(h,i) \in A : x_{hi}^r > 0} \{U_h^r + t_{hi}\}$$

$$\bar{q}_i^r = \arg \max_{(h,i) \in A} \{U_h^r + t_{hi}\}$$

5. Is i the last node topologically? If so, stop. Otherwise, let i be the next node topologically and return to step 3.

Interestingly, the longest path problem is extremely difficult in networks with cycles.

Example



See Figures 7.11 and 7.12 in text.

SHIFTING FLOWS ON A BUSH

Once the L and U labels are calculated, we use these to shift flows from the longest paths to shortest paths.

Newton's method is used to determine how much flow to shift.

From each node i, trace the longest and shortest paths using the q and $\bar{q}$ labels until you find the *divergence node* a where they last split. (If $q_i = \bar{q}_i$, node i can be skipped.)

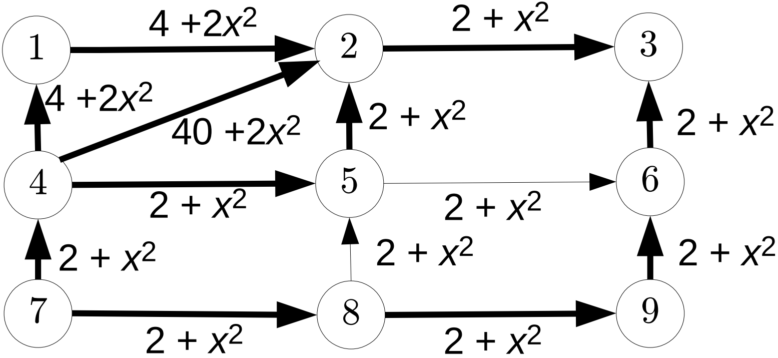
This gives a *pair of alternate segments*, one corresponding to the longest used path π_U, and one corresponding to the shortest path π_L.

Given this pair of alternate segments, Newton's method tells us to shift

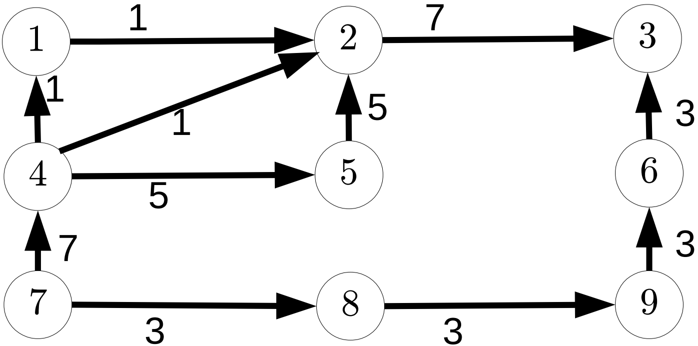
$$\Delta h = \frac{(U_i - U_d) - (L_i - L_d)}{\sum_{(g,h) \in \pi_L \cup \pi_U} t'_{gh}}$$

flow to approximately equalize travel times.

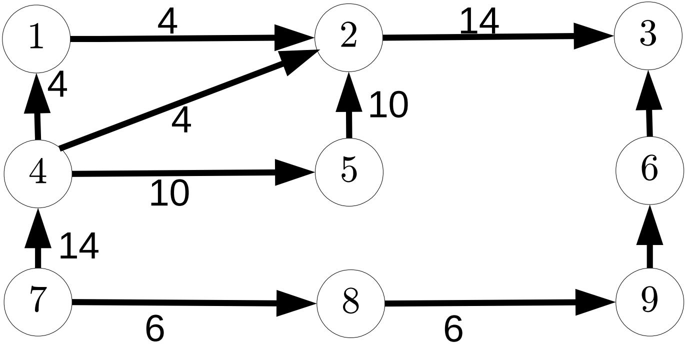
1. Calculate the L and U labels in forward topological order.
2. Let i be the topologically last node in the bush.
3. Scan i by performing the following steps:
    1. Use the L and U labels to determine the divergence node a and the pair of alternate segments π_L and π_U.
    2. Calculate Δh using Newton's method (capping Δh at $\min_{(i,j)\in\pi_U} x_{ij}$ if needed).
    3. Subtract Δh from the x label on each link in π_U, and add Δh to the x label on each link in π_L.
4. If $i = r$, go to the next step. Otherwise, let i be the previous node topologically and return to step 3.
5. Update all travel times t_{ij} and derivatives t'_{ij} using the new flows x (remembering to add flows from other bushes.)


1
2
3
4
5
6
7
8
9
4 +2x2
2 + x2
4 +2x2
40 +2x2
2 + x2
2 + x2
2 + x2
2 + x2
2 + x2
2 + x2
2 + x2
2 + x2
2 + x2

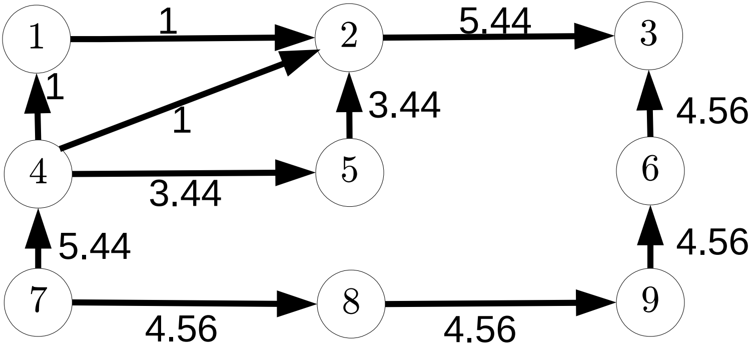
(Figure 8.10a)



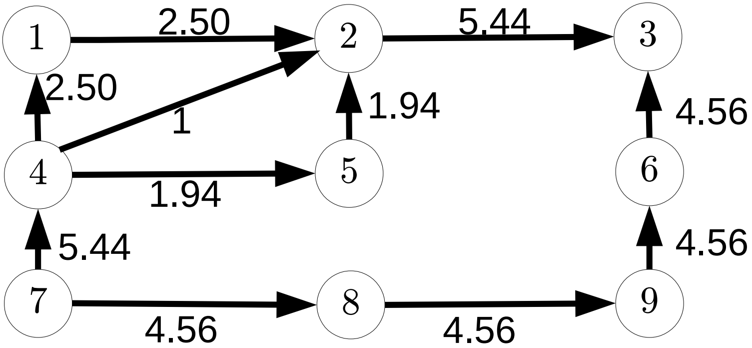
(Figure 8.10b)



(Figure 8.10d)



(Figure 8.12a)



(Figure 8.12b)

UPDATING A BUSH

After all the flow shifts are complete, the bush can be updated by recalculating the L and U labels with new travel times.

Any link with zero flow is removed from the bush (unless it is needed for connectivity).

Any link with $U_i + t_{ij} < U_j$ can be added to the bush.