Bush-based algorithms for solving UE

Pramesh Kumar

IIT Delhi

October 2, 2025

Introduction

- Link-based algorithms require less memory but show slow convergence especially when high accuracy is required.
- Path-based algorithms require high memory but show fast convergence especially when high accuracy is required.
- Bush-based algorithms try to address these limitations.
  - Instead of each origin-destination pair, they keep track of a bush associated to each origin.
  - A bush associated to an origin (or destination) is an acyclic subnetwork of original network rooted at that origin (or destination) which has a path connecting to each destination (origin).
  - You may think of a bush as a set of links associated to paths used by travelers for traveling between a given origin to each destination.
  - This means a bush can never include both directions of a two-way street since cycles are not allowed. This implies, in equilibrium, all travelers from the same origin will be traveling in the same direction.

Examples of bushes

Which of these subnetworks are bushes?

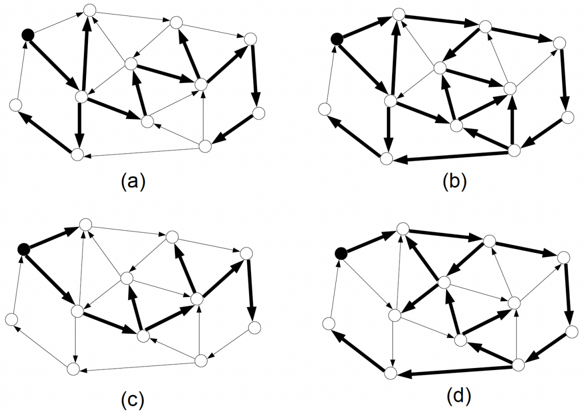

Figure: Figure 6.9 BLU book

General structure of bush-based algorithms

- InitializeBushes: Form an initial bush for each origin. This can be done by building a shortest path tree rooted at the origin using free flow travel times and loading the demand to shortest paths.
- EquilibrateBushes: Shift flows within each bush to bring it closer to equilibrium.
- UpdateBushes: Include links which can further improve travel times and remove links which are no longer used.

Bush-based algorithms

Following are the well-known bush-based algorithms for traffic assignment:

- Algorithm B by Robert Dial
- Origin-based Assignment (OBA) by Hillel Bar-Gera
- Local User Cost Equilibrium (LUCE) by Guido Gentile

We'll study only Algorithm B in this course

Algorithm B

Robert Dial describes his algorithm in a report submitted to TRB[1]:

Start with any feasible flow in a single-origin acyclic network. Find any two arc-independent paths that share start and end nodes, with the costliest one being used. Shift just enough flow from the max path to the min path to equalize their cost (or until the max path is unused). Repeat until all used max-path's costs are within ϵ of their corresponding min-path's.

[1] Dial, R. "Algorithm B: Accurate traffic equilibrium (and how to bobtail Frank-Wolfe." Volpe National Transportation Systems Center, Cambridge, MA (1999).

Algorithm B

```
procedure B($G, \mathbf{t}, \mathbf{d}, \mathsf{tol}$)
    $k = 1$, $x_{ij}^k = 0$, $t_{ij}^k = t_{ij}(0), \forall (i,j)$, and $\mathsf{gap} = \infty$
    for $r \in Z$ do
        InitializeBush($r$)
    end for
    while $\mathsf{gap} > \mathsf{tol}$ do
        for $r \in Z$ do
            UpdateBush($r$)
        end for
        for $r \in Z$ do
            EquilibrateBush($r$)
        end for
        Update link flows, travel times, and travel time derivatives
        Evaluate gap
        $k \leftarrow k + 1$
    end while
end procedure
```

InitializeBush

1. To initialize the bush for any origin $r \in Z$, we first run one-to-all shortest path algorithm. Store the shortest path tree as the initial bush for origin $r \in Z$
2. Load the demand onto shortest paths.
3. Repeat previous two steps for each origin
4. Update travel time of all the links.

UpdateBush(r)

For each origin $r \in Z$

1. We first find the topological order of nodes of the bush associated to r.
2. Find the longest path labels lp of nodes.
3. For each non-bush link (i, j) (link which is not present in bush associated to r), check if $lp_j > lp_i + t_{ij}$. If it is true, then add link (i, j) to the bush.

EquilibrateBush(r)

For each origin $r \in Z$

1. Find the shortest path labels sp and longest path labels lp to each node in the bush associated to r
2. For each node n in the bush associated to r, find the last common node i in both shortest path and longest path to n. Let us denote the sub-shortest path and sub-longest path from i to n using $\underline{\pi}$ and $\overline{\pi}$ (also known as pair of alternative segments (PAS)).
3. If such PAS exists, then move flow from $\overline{\pi}$ to $\underline{\pi}$ using the following formula:
   $\Delta x = \min \left\{ \frac{(lp_n - lp_i) - (sp_n - sp_i)}{\sum_{(i,j) \in \underline{\pi} \cup \overline{\pi}} t'_{ij}}, \min_{(i,j) \in \overline{\pi}} \{x^r_{ij}\} \right\}$ where, $\min_{(i,j) \in \overline{\pi}} \{x^r_{ij}\}$ make sure that we do not make any of the link flows negative on $\overline{\pi}$.
4. Remove links from the bush associated to r such that they carry zero flow and are not required for connectivity of the bush.

Suggested reading

- BLU book Section 6.4
- Dial, Robert B. "A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration." Transportation Research Part B: Methodological 40.10 (2006): 917-936.

Thank you!