A Study in Traffic Assignment Problems

Heren Wei, Applied Mathematics

Mentor: Dr. Ryan Szypowski

Kellogg Honors College Capstone Project

Abstract:

The cities we live in all have very complex traffic networks. An understanding of the traffic network is essential to a network's efficiency. The backbone of traffic network analysis is traffic assignment problems (TAPs), where we study the selection of paths between sets of origins and destinations, with given capacity on each link. There are traditionally three types of algorithms to solve a TAP: path-based, link-based, and origin-based. Path-based algorithms, such as gradient projection and disaggregate simplicial decomposition, are generally highly accurate, but at the expense of requiring large memories. Link-based algorithms are not as costly in terms of memory, but do achieve lesser accurate results with a slower convergence rate. Some examples of link-based algorithms are Frank-Wolfe and restricted simplicial decomposition. The origin-based algorithms are newer algorithms designed to balance accuracy and memory requirements.

In this research project, multiple algorithms and codes are studied and comparisons are made. In particular, a path-based gradient projection algorithm is studied in detail since it produces highly accurate results, and MATLAB code is written to model simple networks. The code will be able to generate the most favorable route from an origin to a destination, given the network capacities and flows.

Background:

Traffic Assignment Problems (TAPs) are essentially an optimization problem. There are two different objective functions that one can choose: user equilibrium and system optimal. User equilibrium indicates that the driver will change his or her route depending on which one is most efficient for the individual, whereas in a system optimal situation, we are minimizing the total system travel time, meaning that somehow a driver knows which route will result in the efficiency of the entire system. In real life, people will take a specific route so that they may get to their destination most efficiently, thus we will program using a user equilibrium objective function:

$$\mathrm{Z_{UE}} = \sum_{\mathrm{a}\in \mathrm{A}} \int_0^{\mathrm{x_a}} \mathrm{t_a(w)\,dw}$$

where x_a is the traffic flow on link $\mathrm{a} \in \mathrm{A}$

$$\mathrm{t_a(x_a) = t_0(1 + 0.15(\tfrac{x_a}{c_a})^4)}$$

$\mathrm{t_0}$ being the free flow travel time

$\mathrm{c_a}$ being the link capacity

The algorithm starts by finding the shortest path between each origin-destination, or O-D pairs. In *Figure 1*, a test network is illustrated. This sample network contains 9 nodes, 7 of which are origins, and 7 of which are destinations. To find the shortest path, we commonly use Dijkstra's Algorithm.

Once the shortest paths are found, we then allocate demand to each of those links. As flow gets placed on each link, it becomes more costly to add more flow, so some demand might get added to another link in order to optimize the network flow. We then update the flows many times before reaching an optimal flow pattern.

Results:

The MATLAB code generates a optimal flow pattern of the test network. However, since the network is simple, it is difficult to analyze the algorithm's efficiency using my program.
Linesch's study on the Anaheim network, which is much more complex with 1,406 O-D pairs, suggests that using an UE objective function is very useful when we need to increase capacity on nodes, resulting in faster convergence.

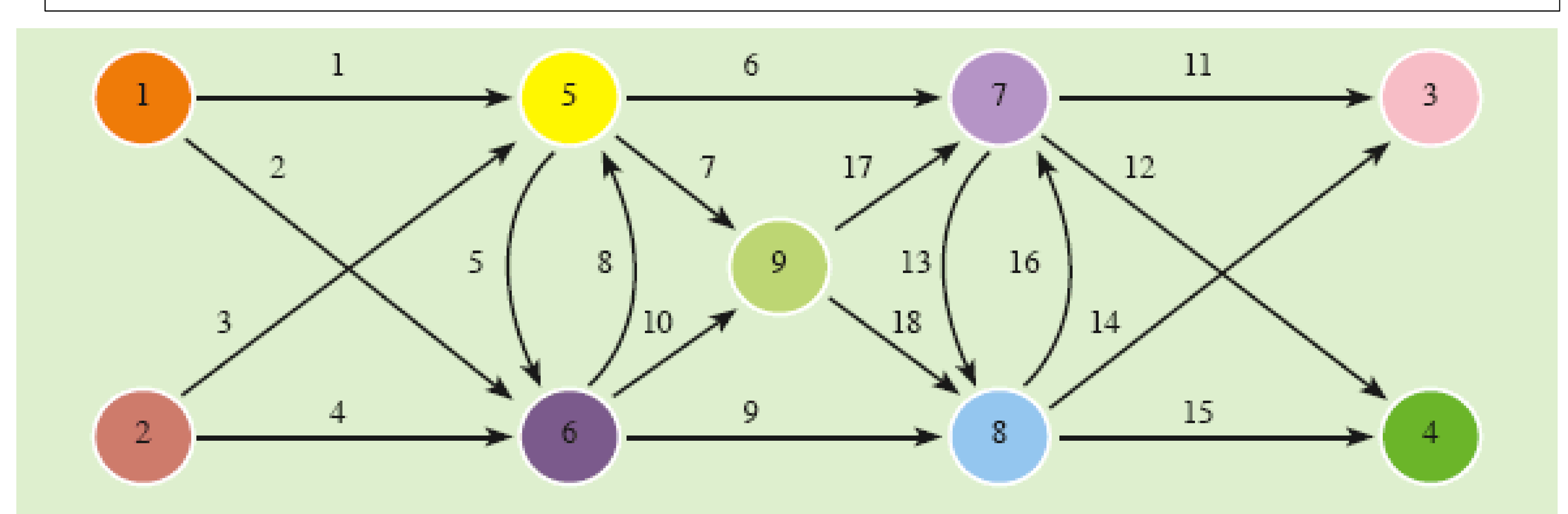

Figure 1: Test network.

Further Modifications of the Algorithm:

1) My MATLAB code uses an "all-or-nothing" assignment update method, where the total link-flow pattern is adjusted only once after all the path flows have been assigned to the network. The alternative is the "one-at-a-time" update method, where the link-flow is adjusted one O-D pair at a time in a cyclic manner. The latter actually converges faster, as observed by Chen and Jayakrishnan.
2) Instead of using the user equilibrium objective function, we could use the system optimal objective function: $\mathrm{Z_{SO}} = \sum_{\mathrm{a}\in\mathrm{A}} \mathrm{t_a(x_a)x_a}$ where we simply minimize the total system travel time. Generally, we study the UE objective function as it is closer to the real-world scenario, but the SO objective function is also an interesting concept to analyze.

Literature Cited:

Chen, A., R. Jayakrishnan. A Path-Based Gradient Projection Algorithm: Effects of Equilibration with a Restricted Path Set under Two Flow Update Policies. Institute of Transportation Studies, University of California, Irvine. January 1998.

Lee, D., Y. Nie, A. Chen, and Y. Leow. Link- and Path-Based Traffic Assignment Algorithms: Computational and Statistical Study. Transportation Research Record 1783. Paper No. 02-2900.

Linesch, N. A Traffic Assignment Assessment: The Anaheim Traffic Network. University of California, Davis. 2006