Public Transit as Networks and the Effect of Topology on Trip-Chaining Passengers

Alyssa Travitz, CMPLXSYS 530 Final Project, Due 30-APR-2020

Introduction

The layout of a city's public transit system governs the ease at which people can access low-cost travel. I'm interested in modeling subway system layouts and how they affect different types of travelers, specifically travelers that trip-chain. Trip-chaining is when a person incorporates at least one intermediate stop between their origin and destination[1]. For example, a parent might their child off at school on their trip between home and work. Trip-chaining is a trait influenced by gender roles. Care-givers for either children or the elderly are disproportionately women, and are caregivers are very likely to trip-chain[2]. Although trip-chaining is common, most public transportation systems are structured in a way that is believed to penalize trip-chaining, in that most public transit systems have a spoke-and-wheel layout, rather than a grid-like layout[3]. This structure assumes that people only need to travel inbound for work and outbound to their homes, discounting the caregivers that travel to several outbound locations. Travelers might have a time penalty from having to take circuitous routes to stop at their locations, or they have a monetary penalty by having to pay multiple fares (some cities do not have fare-transfers between stops or transportation modes). I would like to create a model that captures how the topology of a transportation network affects the time cost of trip-chaining travelers.

In my brief literature review, it seems that many researchers modeling public transit use software packages such as MATSim, a powerful javascript-based software[4,5].For the purpose of applying my knowledge from this class and in the interest of creating a model from scratch, I will use network for my model. I also found that there is active research in the area of congestion and crowding, and that the stochastic element of modeling transit networks may be necessary to accurately capture impacts on travel behavior[6]. Another area of interest to this topic is that of scheduling decisions. Much of trip-chaining comes from people optimizing their travel to minimize their overall travel times. I will not address this in my project, but it is a complex factor that I feel I should be transparent about neglecting[4].

The overall goal of my project is to develop a method to reduce a city's public transit system to a networkx graph to perform a simplified agent-based simulation. I will show that this simplified model is sufficient to demonstrate crowding effects and the effects of network topology on the overall efficacy of a public transit system for trip-chaining and non-trip-chaining agents.

Methods

Model Description

Properties: The agents in this model represent passengers using the CTA train system. The agents interact only with the environment and not with each other. Each agent has the following properties:

- **transit time**: the time spent on travel, in time units
- **home**: the node representing the agent's "home" station, randomly chosen.
- **itinerary**: the list of destination nodes the agent will visit, in the order they will visit them. Agents with more intermediate stops (more trip-chaining), will have more destination nodes in their itineraries. All itineraries begin and end with the **home** node. All desintation nodes are randomly chosen
- **itinerary index:** an index for tracking the agent's progression in their itinerary
- **current node:** the node the agent is currently occupying
- **destination:** the node the agent is currently en route to (as determined by the itinerary)
- **completed:** a simple tracker to record if the passenger has visited all the nodes on their itinerary

Actions: At each time step, all agents follow the below protocol, iterated over in a random order each time:

1. If the agent has not completed its trip and is at a destination, compute the next destination
2. Compute the shortest path to the agent's destination node
3. Move to the next node in the shortest path
4. Accumulate time penalties for transferring nodes or moving to a node that is over capacity (both 3x time penalty)
5. Add +1 to the agent's transit time

Rules: I should note that I initially set up time penalization for nodes being at/over capacity by not advancing an agent until its next node was under capacity. However, this often led to a permanent gridlock in the central loop of the network, so I implemented the time penalty without actually delaying the agents. I would appreciate any feedback on how to explicitly model crowding without causing a gridlock. I realize that this is a result of me treating subway cars as implicit in this model.

Time: Time is implemented discretely, in that each agent can move one node at each time step. Time penalties are counted as multiples of a time step, typically adding 3 time step increments to a passengers **transit time**.

Environment:

I used the Chicago Transit Authority (CTA) train service as a reference for my model. The CTA provides route data in the General Transit Feed Specification (GTFS) format, which provides stations, stops, transport routes, and other necessary data. I converted the CTA's GTFS data to a 2-D network (GTFS_to_netx.py) that keeps the spatial coordinates of the stations, as shown in **Fig. 1.**

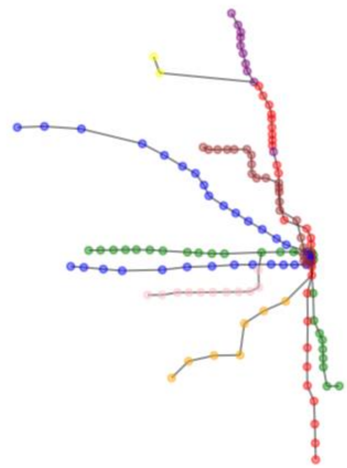

Figure 1 A networkx representation of the CTA train service

The nodes retain their information about which color line they are assigned to, but the connectivity of the graph is represented as single connections between stations, rather than accounting for multiple lines between stations. My goal was to represent the topology of the system in the most simplified format possible.

All of the 140 nodes on the graph are initialized with the same constant **capacity**, or the number of agents that can occupy that node at a given time step. At each time step a population balance is performed to update the nodes with the correct number of agents.

Model Analysis

All simulations were run with 10,000 passengers and a node capacity of 50 (except where stated otherwise). Simulations were run until all agents had completed their itineraries (with a maximum run time for 1,000 time steps in the case of grid locks). Any quantitative result is averaged over three simulations to account for random variance.

Results & Discussion

Station Capacity and Crowding

I first wanted to validate that I would see crowding effects on my network if I sufficiently reduced the capacity of the nodes. It would seem that for a system with 10,000 agents and 140 nodes that there would be a noticeable increase in transit times at a node capacity of 10,000/140, or 71 passengers per node. The limit of this would be when the node capacity is equal to the number of agents, which would represent a transit system where there never wait times. **Figure 2A** shows a violin plot of transit time distributions for each node capacity. From the distributions it seems that the peak of the distributions skews slightly higher than the mean, and that it broadens as the node capacity decreases. To more clearly show the trend of transit time as a function of node capacity, **Figure 2B** shows the mean values with error bars representing the 68% confidence interval (smaller than the marker sizes). As expected, we see a decrease in transit time as the node capacity increases.

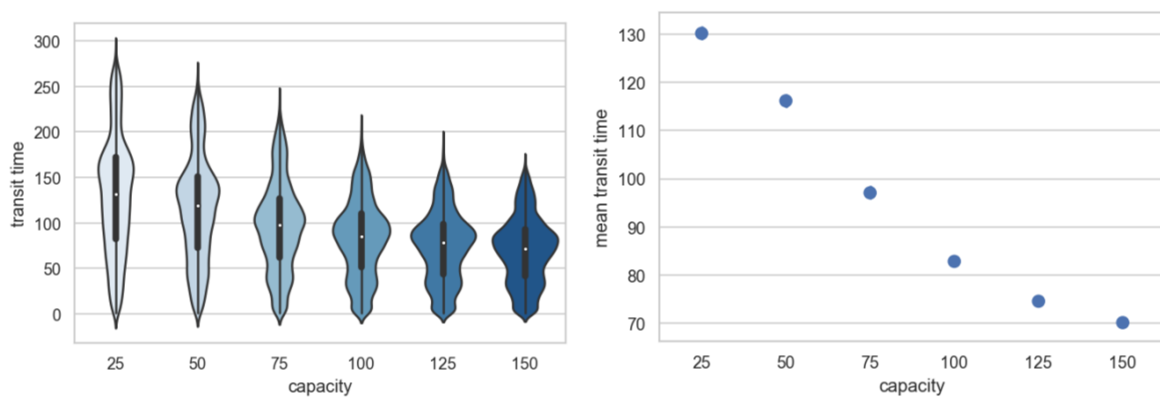

Figure 2: A) *A violin plot showing the distribution of transit times for varying node capacities.* ***B)*** *The same data as A), but depicting only mean values and 68% confidence intervals (error bars are smaller than the data markers)*

Trip Chaining Effects

The main motivation for this project is to see how trip-chaining affects travel times and if changes to the topology of the subway network can favor trip-chaining passengers. I included trip-chaining with the agent property **itinerary**. For each additional stop, I added another random node to the agent's itinerary, always beginning and ending with their **home** node. In **Figure 3** you can see that the mean transit time increases linearly with each additional stop. This is unsurprising but serves as a baseline measurement for the existing CTA network.

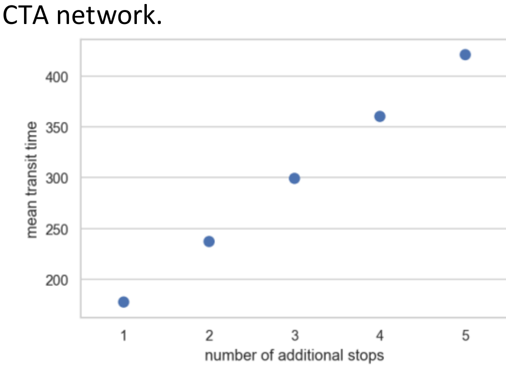

Figure 3 Mean transit times for varying numbers of trip-chained stops

Adding an "Outskirts" Subway Line

To look at how the topology of a network favors or disfavors trip-chaining agents, I added a new line to the CTA, referred to as "CTA_1," both of which are shown in **Figure 4.** This is also unsurprising, but it demonstrates that adding a less central line significantly lowers the mean transit time in this model.

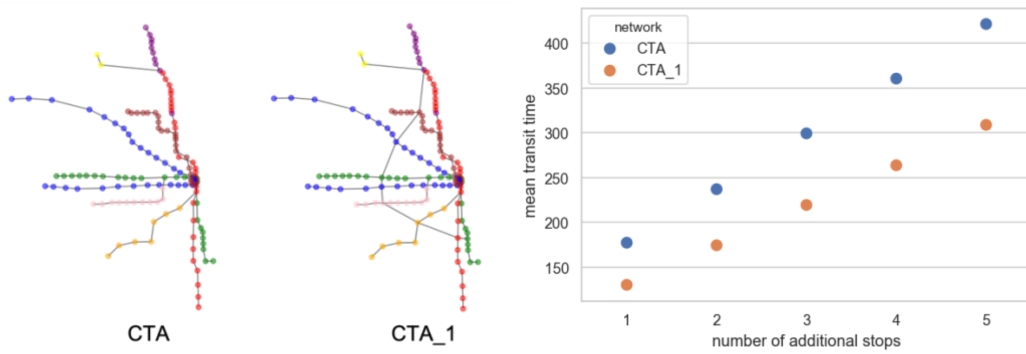

Figure 4: A) *CTA depicts the network graph for the current Chicago Transit Authority train system. CTA_1 depicts my added subway line connecting the outer "spokes" of the CTA.* ***B)*** *Transit times for the current CTA and my augmented network, CTA_1.*

Topology Comparisons

Finally, I wanted to do a little bit of analysis on the networks themselves. **Figure 5** shows the betweenness centrality for each of the nodes. It's clear from the betweenness centralities that adding this additional line is providing many of the shortest paths between nodes.

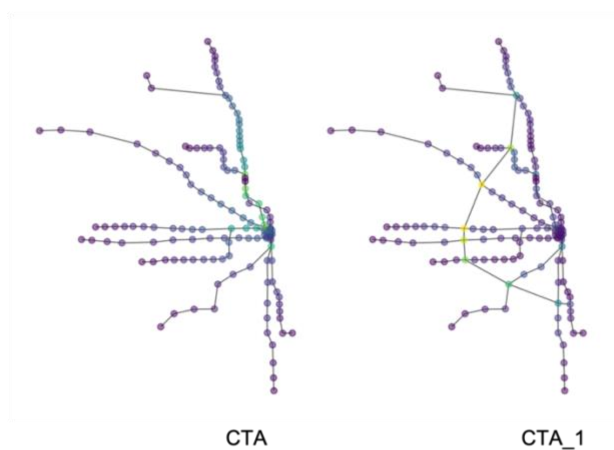

Figure 5*: CTA and my augmented graph, CTA_1. Nodes are color-coded by betweenness centrality using networkx. Lighter colors indicate higher betweenness centralities.*

Future Work

Expanding to More Cities

There is a lot in this project that I wish I could have added complexity-wise and explored analysis-wise. First of all, I would love to replicate some of this analysis for different cities' public transit systems to compare their topologies and bias for or against trip-chaining agents. Converting the publicly available data was a little bit messy and cumbersome, so I chose to stick with a single model, but I think there is a lot of value in comparing the distributions of connectivity and centrality for different cities.

Adding More Transit Modes

For this model I only included subway trains as a simplification, but a significant issue with public transit accessibility is that housing prices may be correlated to their proximity to public transit. I would like to improve my model by randomly initializing agents on a grid that is overlaid on the transit network (since all nodes have geographic coordinates). This way, I can account for walking times to/from subway stations, as well as include the option for agents to choose walking if it is comparable to subway transit (this would require some sort of cost function). I could also add in buses or other modes of transit, and weight their edges according to their travel speed.

Improving the Accuracy of Agent Distributions

A major drawback of my simplified model is assuming that agents' destinations are uniformly spread across the network. In reality, some nodes have higher housing population densities, some have higher workplace densities, etc. One way I can think to account for this is to collect public housing density data, map it to a 3-D probability density function, and then sample randomly using that probability density as a weight along with a random number generator. Similarly, I could modify the station node capacities based on the number of trains that typically pass through each station.

There are infinitely more ways to add accuracy and complexity to this model, and at a certain point I believe it would be more practical to use a more specific modeling software such as MATSim. However, I think that reducing something as complex as public transit systems to a simple network graph allows for looking at how topology fundamentally affects a system's efficacy.

Final Note: Feel free to check out my code repository (https://github.com/atravitz/transit_models), I made an interactive graph and some simulations gifs that I think are neat but couldn't include in my writeup. I'll hopefully clean it up more over the next few days, too.

References

1. Valiquette, F. & Morency, C. Trip Chaining and its Impact on Travel Behavior. in *12th WCTR* (2010).
2. McGuckin, N. & Murakami, E. *Examining Trip-Chaining Behavior, A Comparison of Travel by Men and Women*.
3. Criado Perez, C. *Invisible Women: Data Bias in a World Designed for Men*. (2019).
4. Kaddoura, I., Kickhöfer, B., Neumann, A. & Tirachini, A. Agent-based optimisation of public transport supply and pricing: impacts of activity scheduling decisions and simulation randomness. *Transportation (Amst).* **42**, 1039–1061 (2015).
5. Yang, B. *et al.* Phase transition in taxi dynamics and impact of ridesharing. *Transp. Sci.* **54**, 250–273 (2020).
6. Cats, O., West, J. & Eliasson, J. A dynamic stochastic model for evaluating congestion and crowding effects in transit systems. *Transp. Res. Part B Methodol.* (2016). doi:10.1016/j.trb.2016.04.001