CSC111 Project Proposal:
TTC Route Planner for Toronto, Ontario

Anna Cho, Charles Wong, Grace Tian, Raymond Li

Tuesday, March 16, 2021

Problem Description and Project Goal

In an expansive metropolitan area such as Toronto, it can be difficult for transit users to navigate the bewildering labyrinth of bus, streetcar, and subway paths. Travelling from one destination to another in the least amount of time poses complications for many new and existing users, and subway maps are often insufficient when attempting to discern the most efficient route.

There are 75 TTC subway stations that are currently divided into four lines (Yonge-University, Bloor-Danforth, Scarborough, and Sheppard) ("TTC Future System Map", n.d.), as well as a streetcar network composed of ten routes ("Streetcars", n.d.). Additionally, the TTC had 159 bus routes in 2018, with an average daily ridership of 1.58 million on weekdays ("Operating Statistics", n.d.). In terms of ridership, the TTC is the third most heavily-used transit system in North America ("List of North American rapid transit systems by ridership", n.d.), which is why we decided to create an efficient transit planner for millions of potential users.

We have chosen this as our project because it can be useful for many people taking the TTC—particularly commuters or students like ourselves—and the transit system is especially suited for a graph data structure. Moreover, we could seamlessly integrate relevant computations in order to search for the most efficient path between two sets of coordinates (both of which will be inputted by the user using a point-and-click map).

Although the incorporation of transit schedules (hours of operation for each type of vehicle), temporary construction dates, and future transit lines would likely be outside the scope of this project, our project will still be functional for regular hours of use. The transit destination planner that we plan to build will make use of multiple routes (for all types of vehicles within the TTC) as well as arrival times, in order to generate the shortest path between two points.

Project Goal: How do we create a system that uses TTC routes and arrival times in order to generate the quickest path between two points?

Computational Plan

Data transformation, filtering, aggregation, and structure:

We plan on using the TTC Routes and Schedules dataset from the City of Toronto's Open Data Portal ("TTC Routes and Schedules", n.d.), which provides data in a relational database format, following the General Transit Feed Specification ("GTFS Static Overview", n.d.).

The dataset consists of eight separate text files: agency, calendar, calendar_dates, routes, shapes, stop_times, stops, and trips. For this project, the most relevant files are as follows: stops, routes, and stop_times. Sample datasets of each of these files are included below.

We will represent the TTC routes as a graph, with the stops (encompassing bus, subway, and streetcar stops) as the vertices of the graph. The edges of the graph will represent routes connecting one stop to another.

Computations and Algorithms:

Using the A* pathfinding algorithm, we will be attempting to return the most optimal route between two nodes, which correspond to real-world locations. These paths will be optimized for average schedule times in specific time blocks, for example morning rush hour, noon, evening rush hour, and midnight.

The A* pathfinding algorithm factors in the edge weights of the graph as well as an implementer-specified heuristic in order to determine which nodes to pursue as a possible path. Our preliminary choice for a heuristic is simply the distance from a given node to the final destination. We plan on developing more sophisticated heuristics in order to optimize our pathfinding during development.

Reporting Visual Results:

The program will display a map with waypoints where the user can select their start and end destination, and then display the optimal path for the user on the map. The map will be interactive: the user can select waypoints, scroll, and zoom in/out. To report the results of our computations, the output path will be displayed using a map format as well.

Technical Requirement:

We will use the tkinter library to create a graphical user interface (GUI) for displaying our interactive map, with Pillow (PIL) to help display images. Tkinter has a Canvas widget that allows us to piece images together to create a map, and includes scroll methods to navigate the map as a user.

References

GTFS Static Overview. (n.d.). Google Developers. https://developers.google.com/transit/gtfs

List of North American rapid transit systems by ridership. (2008, March 26). Wikipedia. Retrieved March 13, 2021, from https://en.wikipedia.org/wiki/List_of_North_American_rapid_transit_systems_by_ridership

Operating Statistics. (n.d.). Toronto Transit Commission. https://www.ttc.ca/About_the_TTC/Operating_Statistics/index.jsp

Reference. (n.d.). Google Developers. https://developers.google.com/transit/gtfs/reference

TTC future system map. (n.d.). Toronto Transit Commission. https://www.ttc.ca/Spadina/About_the_Project/TTC_System_Map.jsp

TTC Routes and Schedules. (n.d.). City of Toronto. https://open.toronto.ca/dataset/ttc-routes-and-schedules/

TTC streetcars. (n.d.). Toronto Transit Commission. https://www.ttc.ca/Routes/Streetcars.jsp