Travel path finding algorithm minimizing COVID-19 infection risk

FORUM8

9th CPWC

Team : TJTX

Overview

Based on the need to fight ongoing global pandemic of coronavirus disease 2019 (COVID-19) and the fact that there's no existing application to predict the infection risk of people's travel in a city, we propose to develop an algorithm for travel path finding to minimize infection risk.

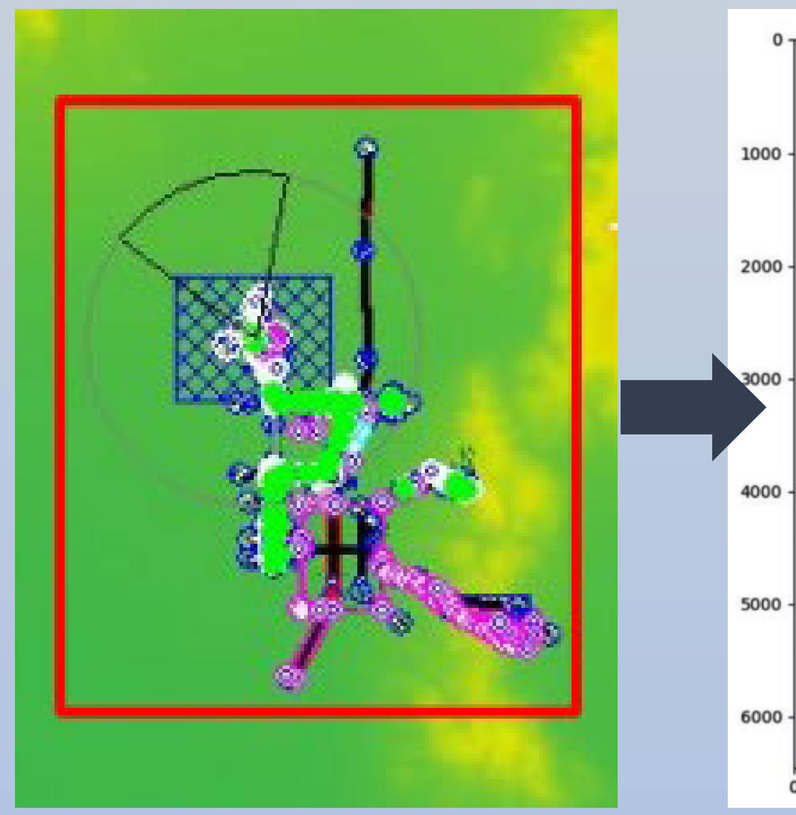
Method

1. Simulation of the traffic
2. Path finding with Dijkstra algorithm.

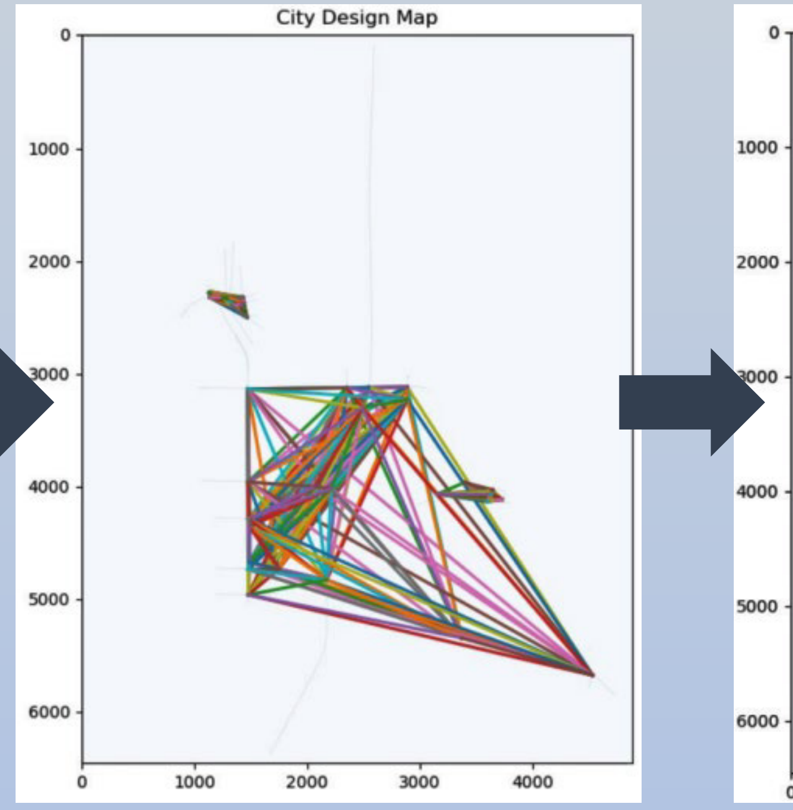
   Path finding algorithm to minimize infection risk value of each travel.

- ➢ After carrying out simulation and path finding algorithm, we did visualization for verification.

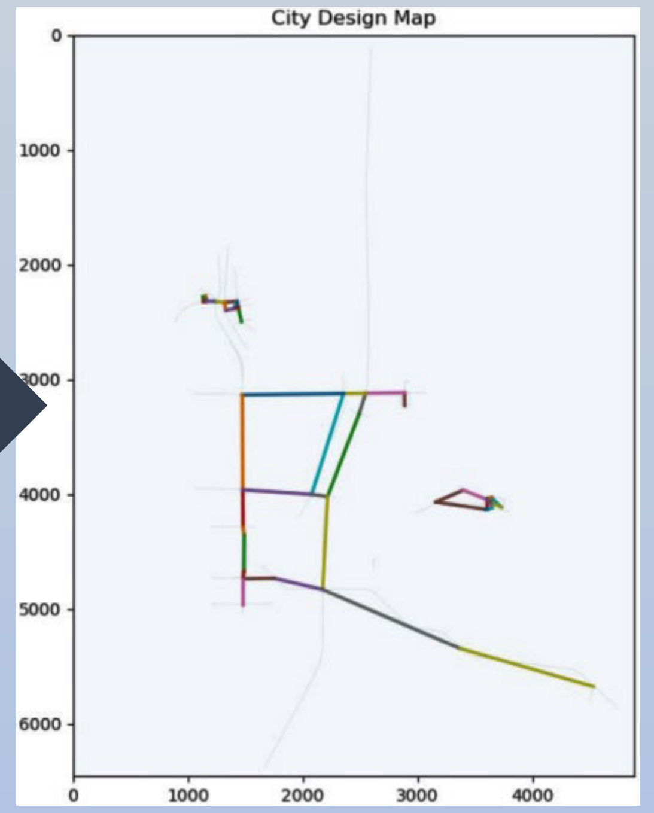
Data structure & Simulation



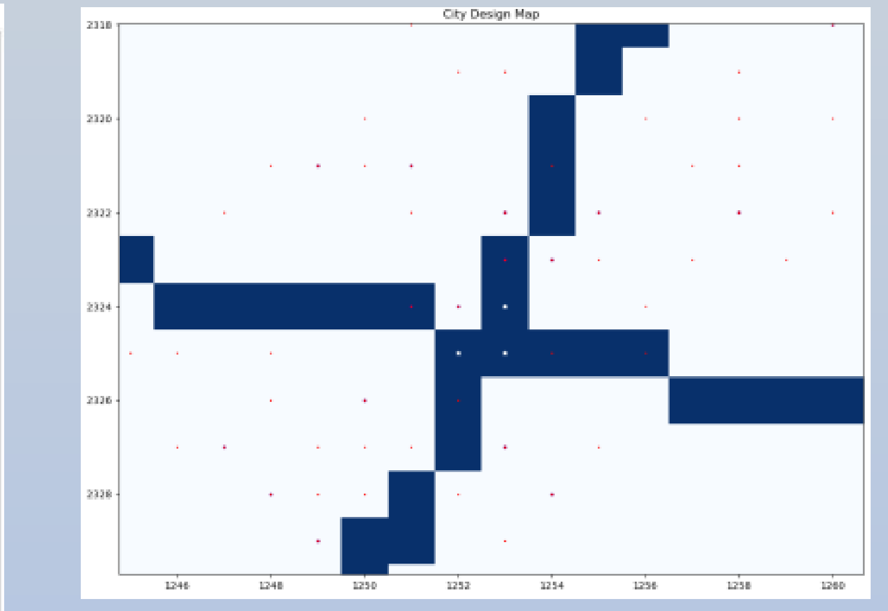
Use the part in the red rectangle to build



- Connected components for between each station
- Time complexity for building it
  m: number of points in the map
  n: number of stations
  BFS for each station: O(n*m)
  Union-Find: O(m)

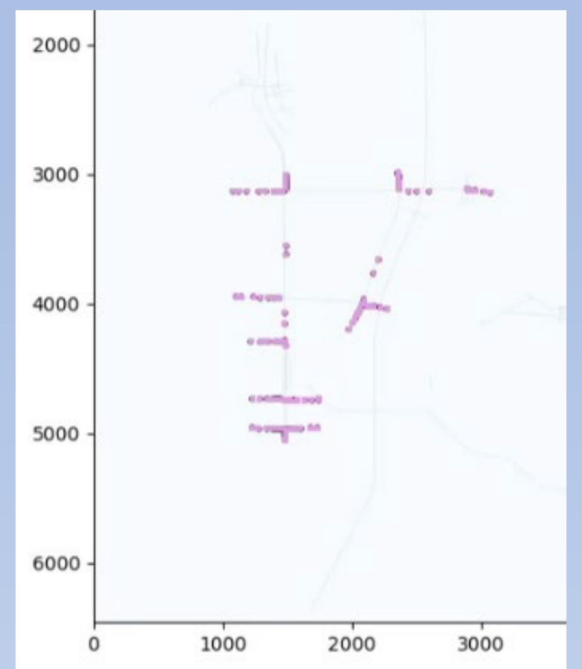


- Maintain a Connectivity Matrix for between each station
- Build it by traversing each road
- Will be used in Dijkstra algorithm

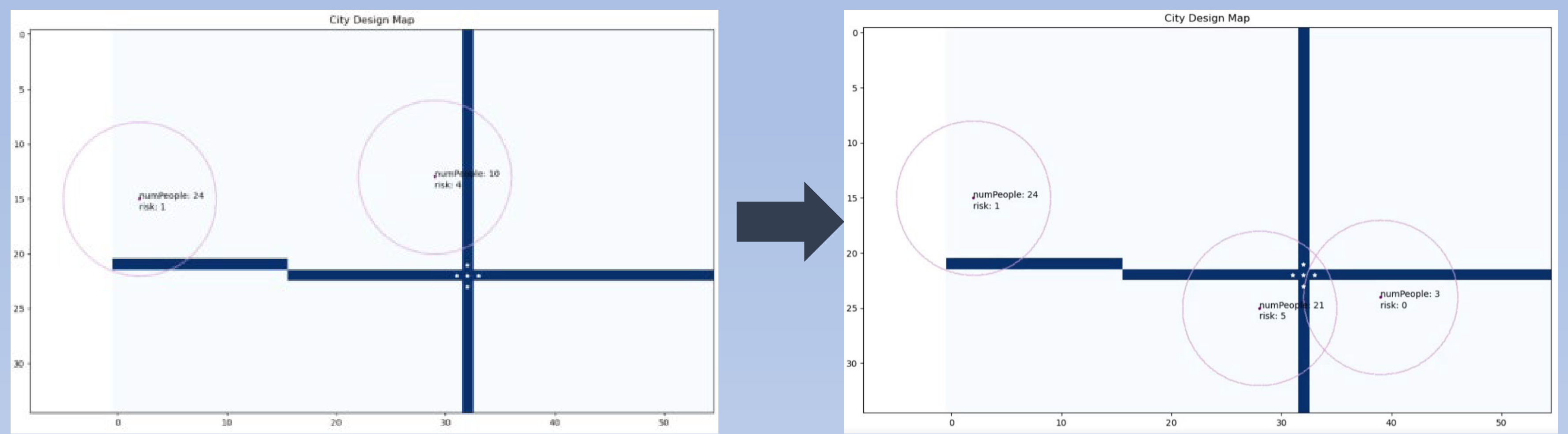


Representations
- Small red point: person moving in the city.
- Larger purple point: bus
- Purple circle around a bus: when the station is inside the circle we know the bus is about to arrive at the station
- White star: means that area is an intersection. For now we set bus stations at intersections

Simulation



Bus placement at an interval of 5 seconds



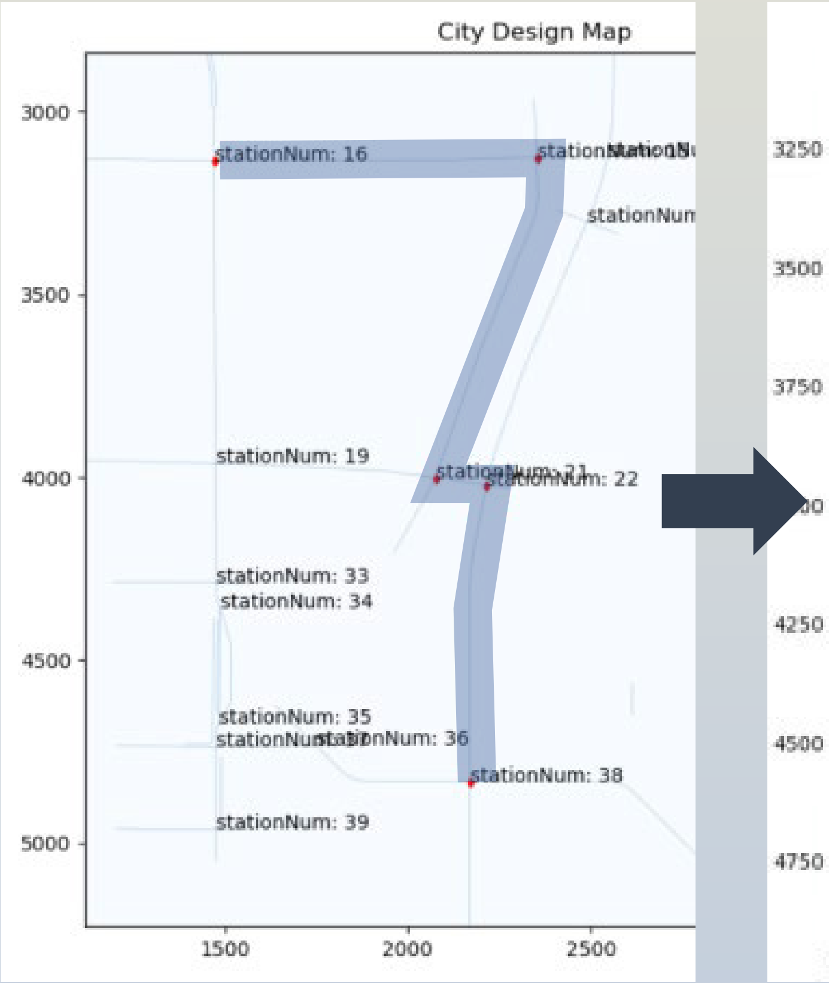
The DLL program places buses regularly on each possible roads, and changes the number of people on each bus thus the exposure risk when the traffic flow is on.

Simulation

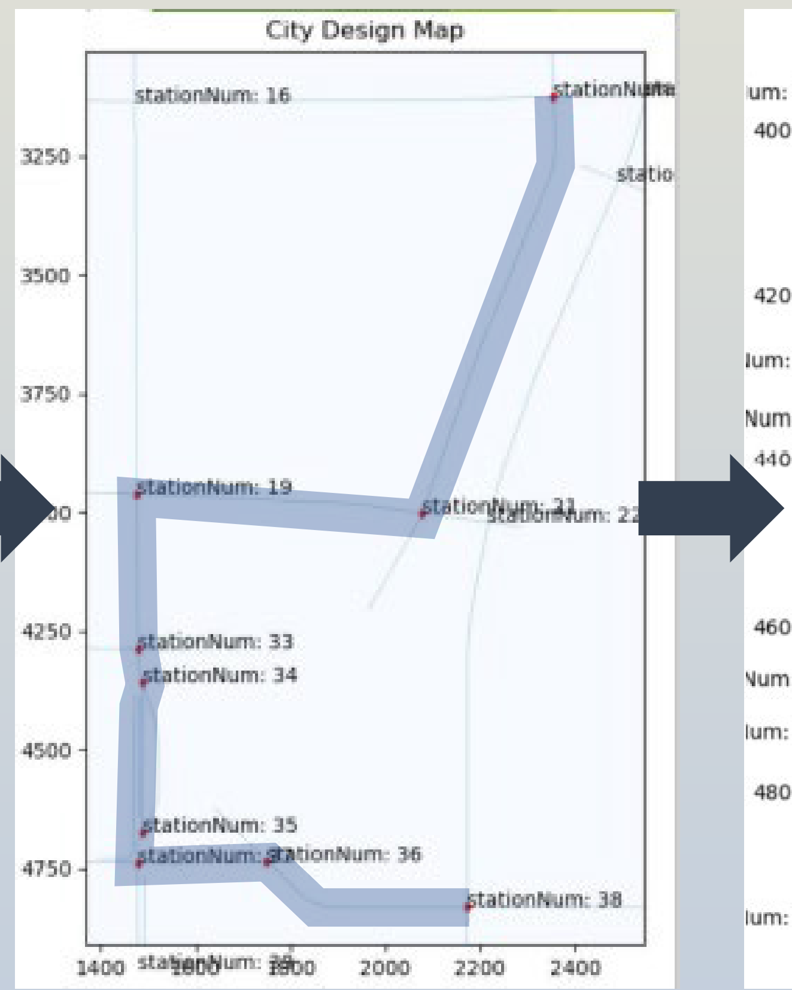
■ Path finding with Dijkstra algorithm

For now we transfer at every station. At each transfer, we calculate exposure risk and choose the path which minimize it.

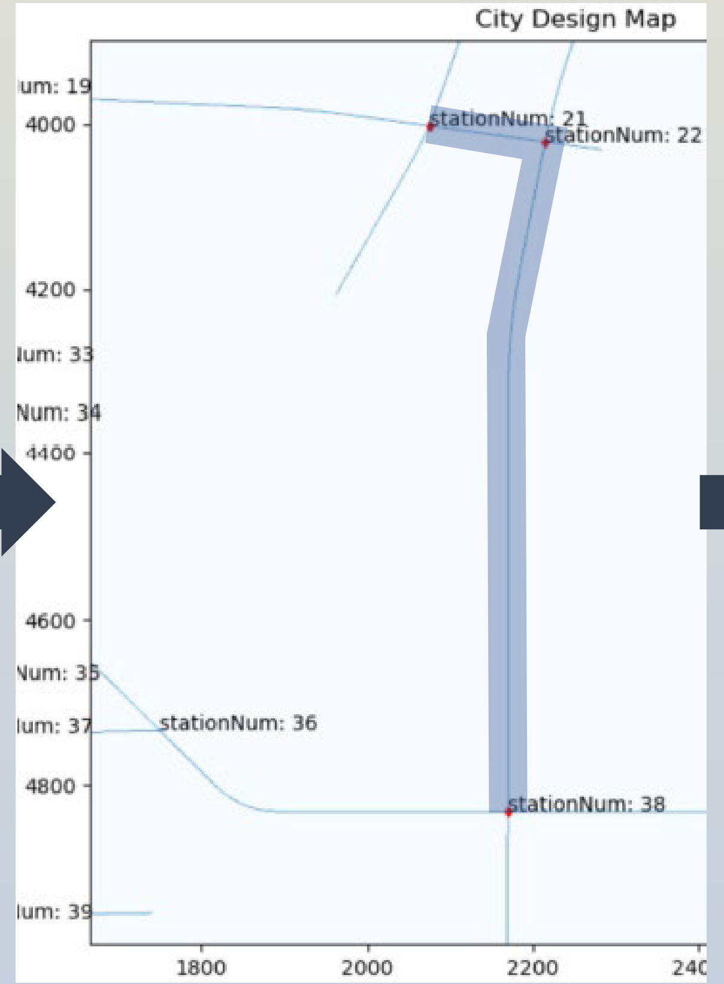
Starting point : station 16, Destination : station 38



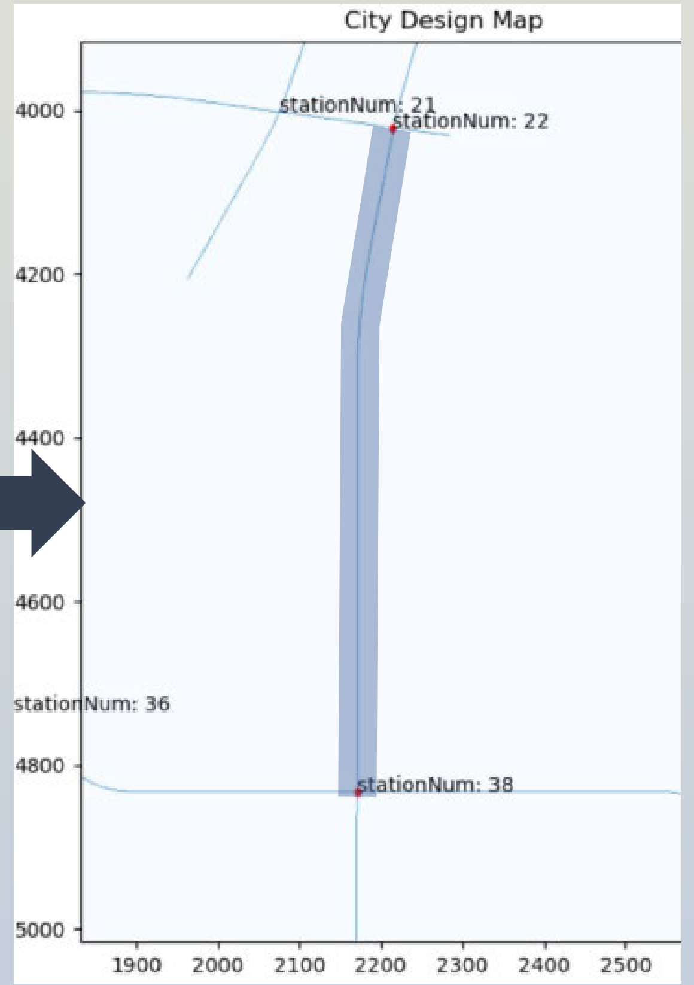
Step 1, the path is 16->15->21->22->38



Step 2, after reaching 15 and recalculating risk, the path is 15->21->19->33->34->37->36->38



Step 3, after reaching 21 and recalculating risk, the path is 21->22->38



Step 4, after reaching 22 and recalculating risk, the path is 22->38

Step 5, reach destination.

Future tasks

- ➢ As we can see, by Dijkstra algorithm with distance set as exposure risk of transportation, physical distance of the path might be much longer, hence a list of possible paths could be listed for user to choose.
- ➢ Algorithm can take into account both these two factors to give its own recommendation, thus is like a heuristic algorithm, e.g. A* algorithm.
- ➢ For now, the exposure risk is set random. Simulation of the quality of medium, i.e. temperature, air renewal rate, solar radiation, etc. could result in a more realistic risk value.
- ➢ Associate buses, roads and stations, and make public transportation schedule.
- ➢ Incorporate pedestrian simulator like REGION so that the human mobility trace and contact can be analyzed, including when people are in a bus or subway train.