Factors guaranteeing QoS in shared bus scheduling

Lee Jae Won
Sungkyunkwan University

지도 교수: 정재훈 교수님

연구실명: IoT Lab

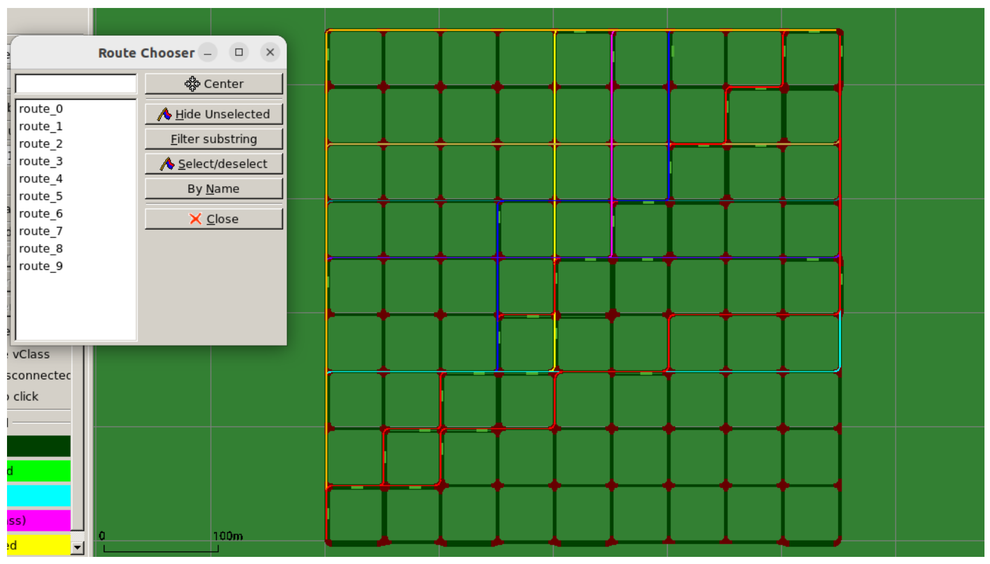
개 요

A shared bus can serve groups of passengers from different pick-up locations to drop-off locations. Previous research papers tried to provide algorithms for the most efficient and fastest path from the perspective of public transportation with minimum travel time, travel distance and energy cost. Greedy algorithm is one of the known algorithms to solve the Traveling Salesman Problem (TSP) in polynomial time. Meanwhile, the COVID-19 pandemic has changed the travel behavior, for instance, avoiding crowded buses at peak hours. This project will experiment three Quality of Service (QoS) factors that can affect the travelling time of passengers, particularly the number of buses running, the number of passengers taking the bus at once and the number of waiting passengers in bus stops. In this project, the Simulation of Urban Mobility (SUMO) program is used to test the relationship between each QoS factor and the average travelling time of passengers taking buses based on the greedy algorithm.

가설

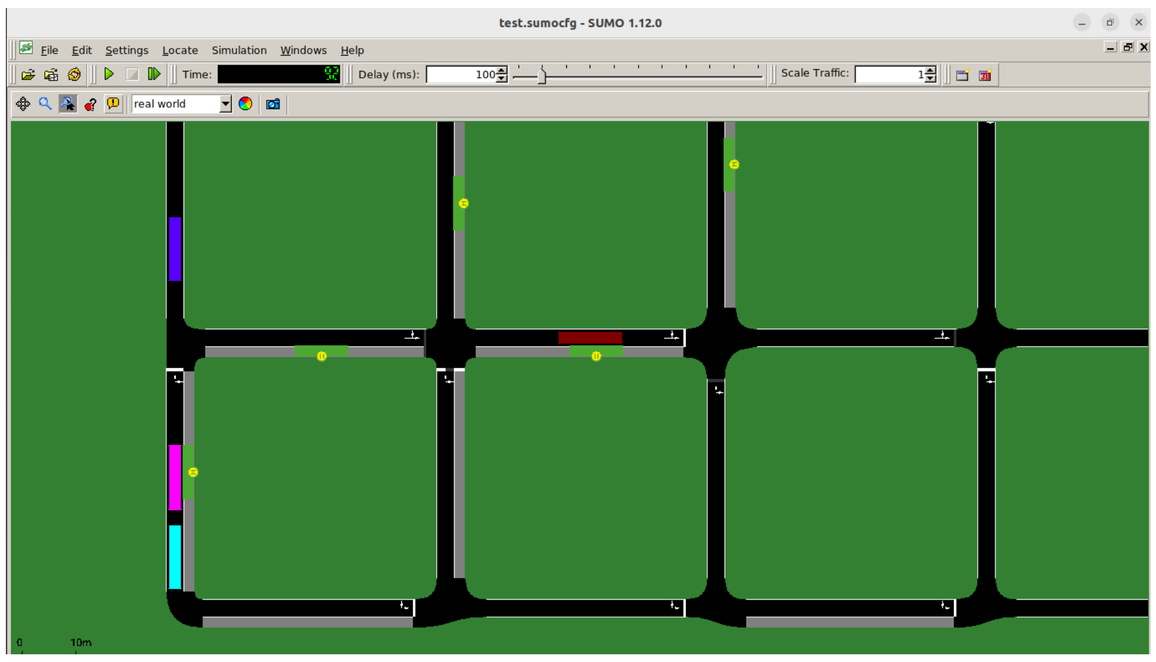
1. If more buses are running on the same route at a time, the passenger's average travelling time will decrease because the passengers will ride on the bus after short waiting time. Thus, it supports the increase in passenger's QoS level.
2. If maximum passenger capacity in each bus increases, then more passengers will get to ride the bus at once with lesser waiting time at the bus stops. Therefore. it supports the increase in passenger's QoS level, too.
3. If more and more passengers arrive at the bus stop and wait for the bus, then both waiting time and average travelling time will e ventually increase because of limited passenger capacity allowe d to take the bus. For this reason, passenger's QoS level will de crease.

시스템 구성



System	**Linux Ubuntu v22.04.3 LTS on Oracle VM VirtualBox v7.0**
Simulation Soft ware	**SUMO v1.8.0**
TraCI File	**python runner.py**
Configuration Files	**Net-file: 9x9grid.net.xml** **Route-file: bus.rou.xml** **Additional-file: test.add.xml** **Sumo Configuration File: test.sumocfg**

결론 및 소감



The TraCI python programming is quite challenging, especially starting from assigning destination bus stops to each person. SUMO TraCI programming in Python required many references and tutorials including SUMO Pydocs documentation. Although this project failed to achieve the goal of moving passengers and calculating the average travelling time in Traci code, it was very meaningful to study and apply the knowledge of simulation step by step. Thanks to a master degree student, Kwon Jun Hee, in IoT lab, weekly appointments were helpful sessions to understand netedit, SUMO and TraCI. Future research can be focused on ach ieving the goal mentioned earlier and also giving recommendation on faster travelling bus route for passengers who are dealing with similar routes that contain the destination bus stop. It can also be a possible factor affecting the QoS of passengers.