ROYAL INSTITUTE
OF TECHNOLOGY

AH2174 Traffic Simulation Modeling and Applications

Lab 2: Microscopic Traffic Simulation Modeling using SUMO - Part I

Xiaoliang Ma *

Nov 9, 2018

Objectives

The objects of this lab are to (i) introduce an open-source microscopic traffic simulation model SUMO (ii) practice for building a simple traffic network and preparing inputs, such as sampling of traffic demand, for microscopic traffic simulation (iii) understand detailed car-following models and their impacts through simulation experiments (iv) analyze the simulation outputs in applications.

1 SUMO installation

Before you install SUMO, you can first read through "SUMO at a glance" (http://sumo.dlr.de/userdoc/Sumo_at_a_Glance.html). Depending on your computer OS, you may first install SUMO from installation files in Windows or Linux. The detailed instruction can be found online (http://sumo.dlr.de/wiki/Installing).

*KTH Royal Institute of Technology, School of Architecture and the Built Environment, Department of Civil and Architectural Engineering, 10044 Stockholm, Sweden

For advanced users, you can also compile SUMO from source codes with the latest version, which is recommended, but you need to install some basic libraries in advance. For detailed intruction, you can refer to the official SUMO website (http://sumo.dlr.de/wiki/Installing/Linux_Build).

For MacOS, we suggest you installing a virtualbox and install a Linux virual machine (we recommend ubuntu). Then, you can install SUMO on the virtual machine according to the online instruction.

2 Simulation of traffic flows

2.1 Create simulation networks

You will first learn how to construct a simple road network and then simulate road traffic. There are several ways to build a road network in SUMO. For example, you can manually draw network using `netedit`, a SUMO tool for network editing. Or, you can directly import in the model information from open-street map. In this lab, we will however simulate only a simple highway road section and therefore you can easily construct the network model manually. The network modeling includes the following steps:

1. Create nodes in the file `sumo_test.nod.xml;`
2. Define edges (links) between nodes in the file `sumo_test.edge.xml;`
3. Define the edge types in the file `sumo_test.type.xml;`
4. Apply `netconvert` command to formulate a network file `sumo_test.net.xml` by the following command:

```
netconvert --node-files sumo_test.nod.xml --edge-files
sumo_test.edge.xml -t sumo_test.type.xml -o
sumo_test.net.xml
```

When you download and extract the zip file for this lab, you will find those files that have been initialized with one defined example. You can run the command and generate the network file and use `sumo_gui` to visualize the model and run simulation. Please don't edit the network file by yourself.

2.2 Prepare vehicle inputs for simulation

The vehicle demand in SUMO simulation can be defined in several ways or by applications. There is exhaustive documentation about detailed features of SUMO in the web (Vehicle Generation). In this lab, we simply add vehicle type, route and simulated vehicles by editing the file `sumo_test.rou.xml`. In the downloaded file for the lab, you will find that I have added some example lines to define route, vehicle type and generated vehicles for you to try some simulation runs. In order to run simulation, you need to futher create the SUMO configuration file `sumo_test.sumo.cfg` by defining the names of input files and simulation time as well as other necessary parameters for you to customize the simulation model. Again, I provide one example file.

Now you can open SUMO using `sumo-gui` and load simulation network and configuration by opening the configuration file. You can start simulation runs. Please check if the network is defined as you wanted and all vehicles are generated correctly.

2.3 Run simulations through TraCI API

In practice, we often need to customize different aspects of our simulatio model or even develop advanced application. SUMO provides a TraCI API that users can interactively access the simulated objects and program advanced applications. For example, you can run many simulations repeatly using different random seeds and model parameters using TraCI API. TracCI can be programmed using the Python script. In the extracted files, there is one file `sumo_traci_test.py`. By running the python code, you can try to load the network you created in the last section. You can run the simulation through TraCI. The sample code provides you an example to get the vehicles in the network every simulation time step.

3 Mini-project assignment part I

This mini project has two parts. In this part, you have the following tasks:

3.1 Model building and generate random vehicle input (2p)

The first task is to build a highway section model similar to the provided example. The highway section is composed of five edges (edge 1, edge 2, edge 3, edge 4 and edge 5) of one kilometer each (e.g. from (0,0) to (0, 1000), ..., to (0, 5000)). They are all three lanes with speed limit 70 km/h. Then you can load the network using `sumo-gui` and inspect the network from GUI. For detailed properties of edge, please look into the SUMO online document.

When the network is ready, you have to define the vehicle like what the example shows in `sumo_test.rou.xml`. Please code to generate random vehicle arrivals as a possion process (3,000 vehicle per hour) and write the vehicle generation in the file.

3.2 Simulation output (2p)

The example codes already show that you can access vehicle state per simulation step trough TraCI API. Your task here is to calculate and plot the average vehicle speed in the highway section every minute during the simulation. Please define the simulation time of 60 minutes with simulation timestep 0.1 sec (the default is one section). The first 15 minutes are considered simulation warmup time, not considered for the speed calculation. Please repeat 5 simulation runs with other random seeds and compare the results. Compare with the aggregate results of 5 simulation runs. What if we run 20 simulation runs?

3.3 Alternative scenario (2p)

An alternative scenario is that the last edge, edge 5, is changed to a two-lane segment (the rightmost lane closed) due to some construction event and the speed limit is set to 40 km/h for that edge. Please repeat the experiments with 20 simulation runs and compare the simulation results with the original scenario.

Examination

This project assignment will have in total 10 points and a minimum of 6 points is required to pass. Write a report and submit it in CANVAS before November 22, 2018. For all tasks, motivate with plots what you have implemented, including a description of what the plot shows. Also send your source codes (.zip) along with the report.