Project description:

This project simulates traffic in any location on earth. Cars are set to have random starting points and destinations. The goal of this project is to study the effect on traffic by calculating weight on edges using different capacity/time step ratio. The program also provide visualization option (may be laggy).

Pre-Request:

-Have Python 3.x

-Install osmnx package:

‘conda install -c conda-forge osmnx’

OR ‘pip install osmnx’

\*packages for master branch is provided in the Python-Code directory

\*activation command ‘source venv/bin/activate’

\*deactivation command ‘deactivate’

-git clone <https://github.com/T-Wick/Python-Patrol>

-(Parallel branch only) pip install joblib

In trying to create the best model we ended up having two major branches that simulate the same phenomenon. Therefore we included a how to for the master branch version and the parallel version.

**If using a git clone version:**

Just to be sure:

‘git checkout master’

Then to run the project

‘python driver.py’

User-adjustable variables:

User adjustable variables are located in driver.py

CENTER\_LATITUDE = 47.608013

Adjust the latitude of the center point from which the map is made

CENTER\_LONGITUDE = -122.335167

Adjust the longitude of the center point from which the map is made

DISTANCE\_FROM\_CENTER = 500

Adjust the radius of the map, thus increasing or decreasing number of

Intersections and roads. (meter)

NUM\_CARS = 1000

Adjust the number of cars to be initialized on the map

RANDOM\_START\_DESTINATION = True

True: Each car’s start and destination are randomly determined

False: Every car starts with the same randomly determined start and destination

TRAFFIC\_TOLERANCE = 0.75

Adjust the percentage of traffic a car will considered congested.

MODIFIED = False

True: Use modified capacity dijkstra’s

False: Use original unmodified dijkstra’s

**VISUALIZATION = True**

True: Animate a single simulation based on above values

**False: Do not display an animation instead display analysis graphs**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

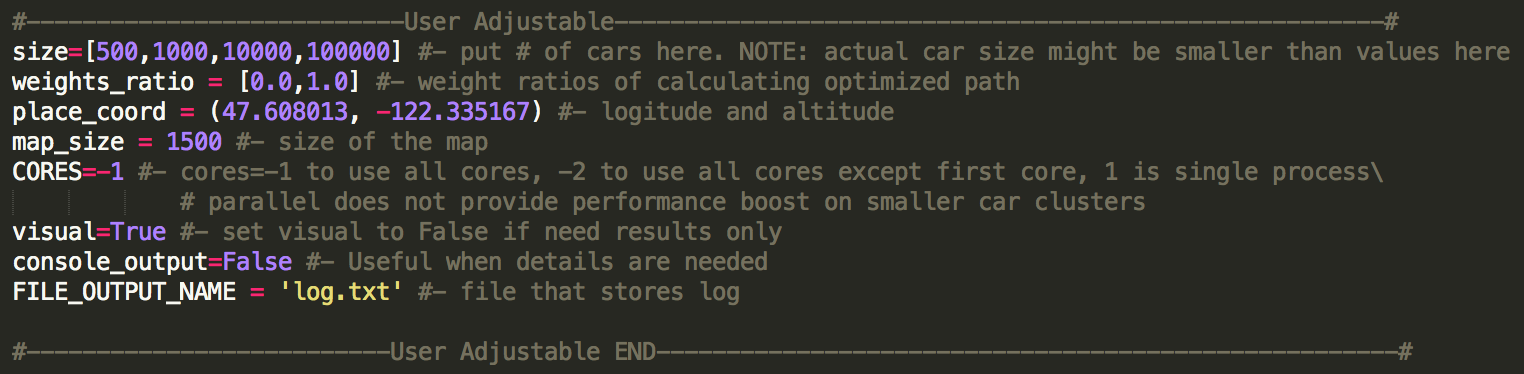
To run the parallel branch:

‘git checkout parallel’

Then to run the project

‘Python driver.py

User-adjustable variables:

User adjustable variables are located in driver.py

Size - list of car numbers desired to simulate. It can hold a list with single value if single car size size simulation is desired. The program will iterate through size and run simulations.

Weights\_ratio - list of ratio desired to simulate. The number should be in [0.0:1.0]. Any number outside of the range is not promised to run properly.

Place\_coord - the longitude and altitude of the map center. Default is Seattle Downtown.

Map\_size - the radius of the map, default is 15000

Visual - if visual is true, the program will use the first number in size and weights\_ratio and provide animation. (only one simulation runs if visual is true). Suggest turning visual off if doing analysis only

CORES - number of CPU CORE desired to compute. -1 represents all cores, -2 represents all cores except the first one. A number > actual core number will result in running on single core.

Console\_output - if true, the program will print information on the terminal.

FILE\_OUTPUT\_NAME - str, path to log file. Log file will record information during the simulation.