Traffic Intersections

-- CS166 Location-Based Project

Executive Summary

This project models a particular section of traffic in London to compare different traffic engineering strategies to optimize the traffic flow, satisfying the city's needs. The conclusion is that the new traffic light strategy that takes the current road density into consideration does not improves the average traffic flow compared to the deterministic traffic light strategy in place right now. The deterministic strategy is performing well enough for the current scenario.

Introduction

In this project, a model is implemented by Python to simulate the traffic system in a particular secion in London. The goal is to compare different traffic engineering strategies and find out how to optimize the system to satisfy various needs. The performance of the system is analyzed by average traffic flow, which is the average percentage of all of the roads being filled with cars.

Data Collection

The data is collected offline. The section chosen is in Bloomsbury, London. There are four intersections (labeled A, B, C and D in Figure 1) and four roads (with 17 lanes labeld in Figure 2). The four roads are Tavistock Pl. on the north, Russel Square on the south Bedford Way on the west and Woburn Pl on the west. The allowed direction to travel for each road and how the cars can turn at each intersection is presented in Figure 2.

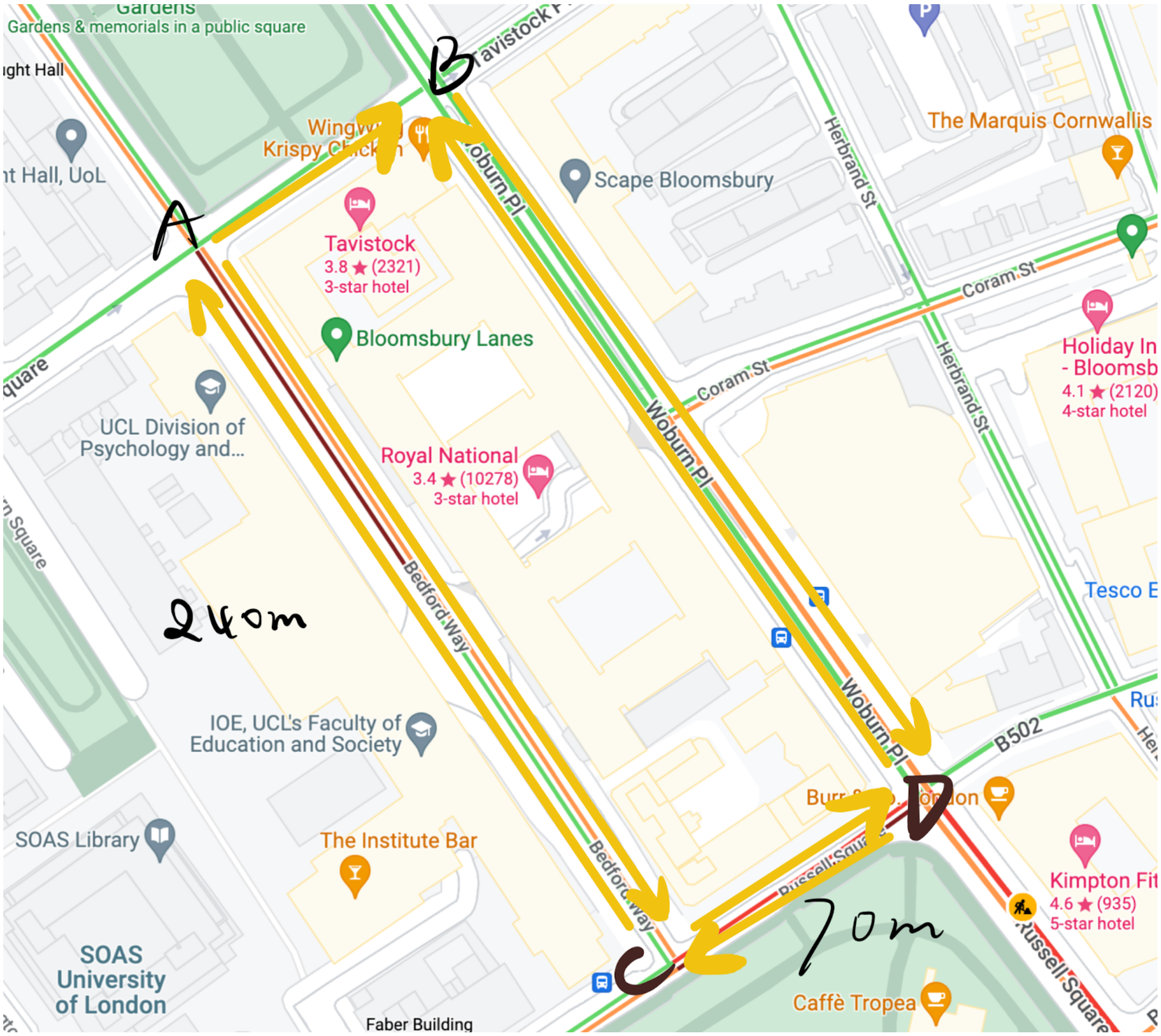

Figure 1. Screenshot of the Google map of the region chosen

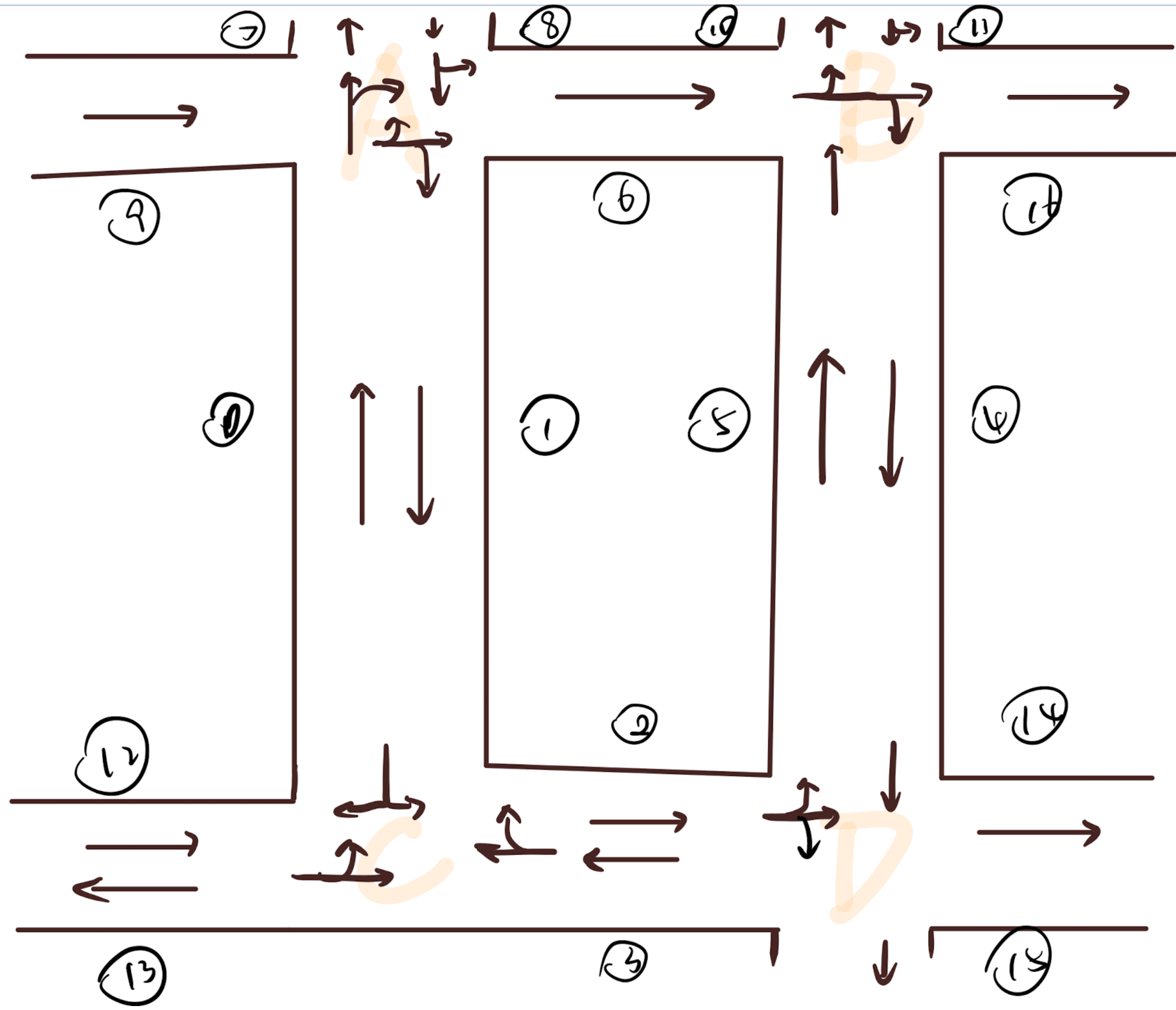

Figure 2. Simplified map of the chosen area with arrows representing the direction of the road and numbers indexing the roads

The traffic density in each street is estimated by the total number of cars that pass in each direction in a predetermined time interval. More specifically, it's the average number of cars that enter the street in 65 seconds. 65 seconds is chosen because the traffic light cycle for each interval is 65 seconds and we assume the car to go through a street within every cycle. So the count would indicate the number of cars currently in the street (which is taken by average over multiple cycles).

Road Index	0	1	2	3	4	5	6	7	8
Density	1	1	7	1	2	5	6	2	1
Road Index	9	10	11	12	13	14	15	16	

Density	5	9	6	6	1	1	9	7	

Table 1. The average density of each road based on the collected data

Road length is measured by the ruler tool on Google map. The north-south roads 0, 1, 5, 4, 7, 8, 10, 11, 15 are of length of 240 meters and the other ones (east-west roads) are of length 70 meters. If we assume that the average length that a car takes (itself and intervals between cars) is 6 meters, then each of the two kinds of the roads have 40 slots and 11 slots.

Traffic light strategy

Besides intersection C which doesn't need a traffic light, each intersection has a traffic light cycle of north-south, east-west, pedestrians which alternates after a predetermined number of seconds listed below. Every intersection have around the same time for each cycle which is around 65 seconds, enabling them to coordinate. The flow of traffic is also guaranteed by having the road in front empty first and then allow cars into the emptied road. According to [Tfl (Transport for London)](), there are also other strategies in place like real-time update of traffic light intervals to adjust for different amount of traffic.

Intersection index	North-south (second)	East-west (second)	Pedestrians (second)
0	10	40	15
1	25	25	15
2	0	0	0
3	40	25	15

Table 2. Traffic light engineering strategy of the chosen area

Model Description

The model simulates a simplified version of the traffic system. Assumptions and variables of the model are discussed in this section.

Cellular Automata model is chosen to represent the situation. In cellular automata models, a set of automata are arranged along grids whose states are simultaneously updated by a uniformly applied state-transition function that refers to the states of their neighbors.

This model is built upon the model described in Nagel & Schreckenberg (1992). Each lane is defined as a one-dimensional array of n slots and cars enter from left and exit from right. Each slot can either be occupied by one vehicle or empty. Each car has an integer velocity with values between 0 and the maximum speed the road allows. The car that hit the boundary will move on to the next lane.

The system is broken down into cars, traffic lights and lanes. The assumptions and justifications are listed in the section below.

Model Assumptions

1. To simplify the behavior of cars, cars are assumed to be unable to take over other cars that they can only follow the previous ones and accelerate and decelerate accordingly.
2. The average length of cars plus the gap distance between cras is assumed to be 6 meters. Therefore, the north-south road has 240/6 = 40 positions for cars and the east-west road has 70/5 = 14 positions for cars.
3. The time spent on passing the intersection is ignored for simplicity.
4. In case in which there's no preceding lanes (such as for 8, 9, 11, 12), the number of cars estimated to enter the road is assumed to follow a normal distribution of the observed number of cars nentering as a mean and some variation based on n.

$$\#\ of\ cars\ entering\ (min) \sim Normal\ (n,\ 0.01\sqrt{n}\,)$$

5. The direction of the cars are drawn from a multinomial distribution with the probability vector corresponding to the observed percentage of cars going in each direction.
6. If a car exited from a previous road but the road that it's moving forward to is full, it won't block cars going into the intersection from other directions.

Model Specification

Based on the assumptions, the chosen model, variables/parameters, and update rules are shown below:

Model Variables and Parameters

- For car
  - Number of cars
  - Current speed of the car
- For road
  - Number of cars
  - Maximum speed
  - Probability of random slow-down
  - Length of the road (number of cells)
  - Density of cars (average cars per cell)
  - Arrival distribution (for some roads)
- For traffic light
  - Time of green of the traffic light

Update rules

For a single road (adapted from the study guide)

- Each car has a current speed, vt. We first determine the next speed vt+1 for each car on the lane
- If vt < vmax, then accelerate to vt+1 = vt + 1, otherwise do not accelerate
- If the number of empty cells between this car and the next car is d < vt+1, reduce the speed to vt+1 - d , otherwise remains unchanged
- If vt+1 > 0, reduce the speed vt+1 = vt+1 - 1 with probability p (models when cars randomly slowing down or over-braking)
- Each car changes its position (moves forward) by vt+1 cells to the right. If the current position is over the length of the road, then it will be added to a list to be added to the road ahead.

For each timestep, check for each intersection which traffic light and corresponding second is at and update for corresponding roads

- First update the road that the traffic light is allowing to move according the rules listed above
- Then update the road that the current moving roads are connected to add cars from the exited list
  - If the road has empty slots at the beginning and the distance is more than vt+1, then the exited car will move forward by vt+1 cells from the previous road to the current road

- If the road has empty slots at the beginning but the distance is shorter than vt+1, then the exited car will be added to the last available spot
- If the road doesn't have empty slots at the beginning, then the exited car will be wait at the intersection and be added in the next cycle

Implementation

The model is implemented using an object-oriented approach, with classes for car, road, traffic light, traffic system. The attributes keep track of the properties of each object and the method handles the update rules. The design of classes is explained in the comments of the code.

Results and Discussion

Empirical analysis

The results are obtained from running the simulations 50 times for various initial density, probability of slowing down after a time unit of 500 seconds for the system to reach equilibrium (but the number should be higher if not because of the computational constraint. The metric used is average traffic flow which indicates how crammed the roads are. Ideally we would run the simulation for over 1000 times for a more certain result because of the randomness of Monte Carlo simulations.

Expected value and 95% confidence interval are chosen to analyze the results because the expected value would show the average value over simulations and the confidence interval would show the span of possible outcomes and how much confidence we have. All results contribute to answering the question 'What's a better traffic light engineering strategy for the chosen areae?'

With the current traffic light strategy:

Varying the car density of the road, it's showed in Figure 1 that average traffic flow grows steadily as car density increases from 0 to 1. There's a jump around car density of 1 resulting in average traffic flow jumping from 0.5 to 1. This might be because a high initial density would already cause traffic jam in the first place and cars existing from a road can't enter the next roads. The confidence intervals are very narrow.

Varying the probabilities of slowing down, the simulation has real-life car density based on the data collected and it's shown that average traffic flow is around 0.08 when car density is lower than 0.4, and it increases to around 0.16 as car density increases from 0.4 to 0.6. The confidence interval also increases because more cars would randomly slow down, causing traffic jam on one hand and more uncertainty on the other hand.

The simulation results shows that, even with high probabilities of slowing down and high initial car density, the average traffic flow is still lower than 0.5 which indicates that it’s very unlikely that congestion will happen. The system is doing well.

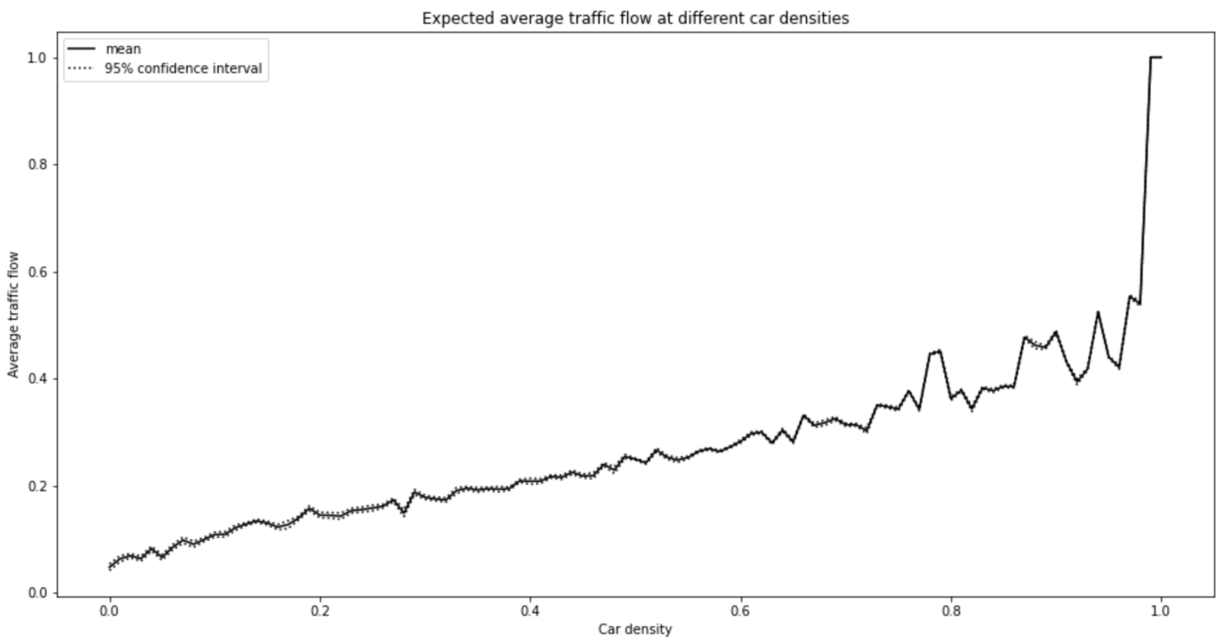

Figure 1. Mean and 95% confidence interval of average traffic flow given current traffic light strategy regarding different car densities

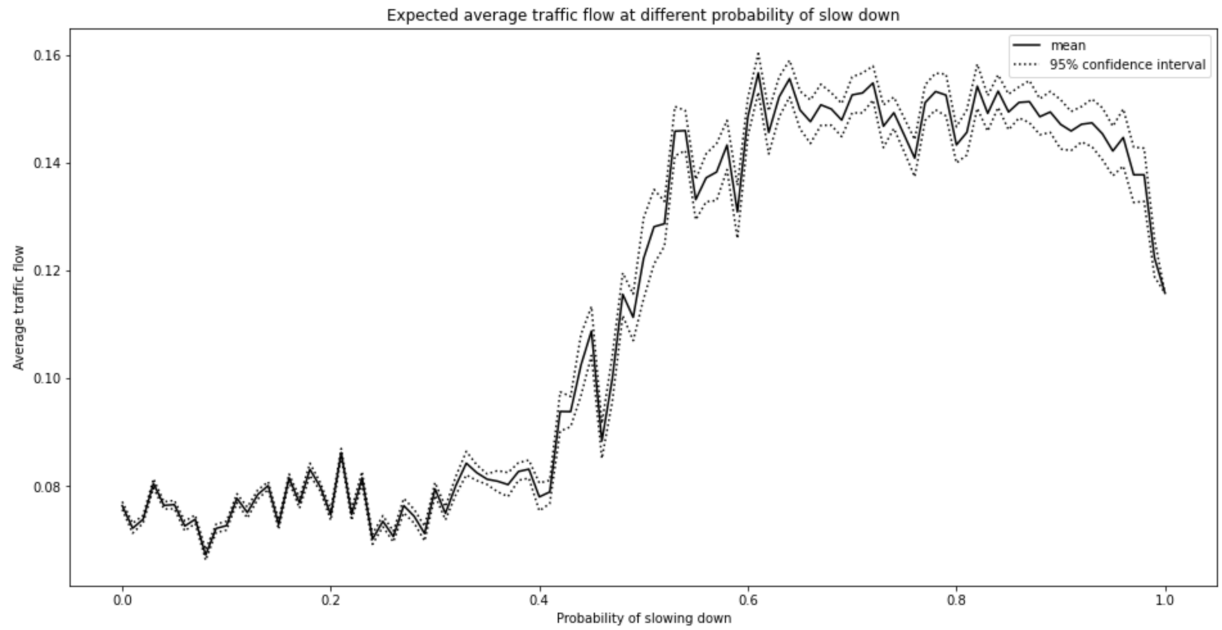

Figure 2. Mean and 95% confidence interval of average traffic flow given current traffic light strategy regarding different probabilities of slowing down

With the new traffic light strategy:

A new traffic light strategy is designed to make the system more efficient, allowing more time when the average traffic flow is high and shorter time when the average traffic flow is low. Specifically, when the average traffic flow for a certain lane is above 0.5, for every 0.1 increase, the corresponding traffic light is 10% second longer; when the average traffic flow for a certain lane is below 0.5, for every 0.1 increase, the corresponding traffic light is 10% second shorter. It is expected to make average traffic flow better distributed and cars to wait for shorter time period.

Varying the car density of the road, it's showed in Figure 3 that average traffic flow also grows steadily as car density increases from 0 to 1, similar to the behavior with the current strategy, but with a slightly wider confidence interval.

Varying the probabilities of slowing down, the average traffic flow is always between 0.14 and 0.16. Comparing with the current strategy, it has wider confidence intervals and higher average traffic flow when the probability of slowing down is low.

The simulation results shows that, the new strategy doesn’t improve the current system significantly, but managed to average the behavior of the system for varying parameter values. However, it’s expected that drivers would wait less with the dynamic strategy. Ideally, more metrics should be used to compare different strategies. For example, average driver waiting time can be used to capture system behavior when the car density is low.

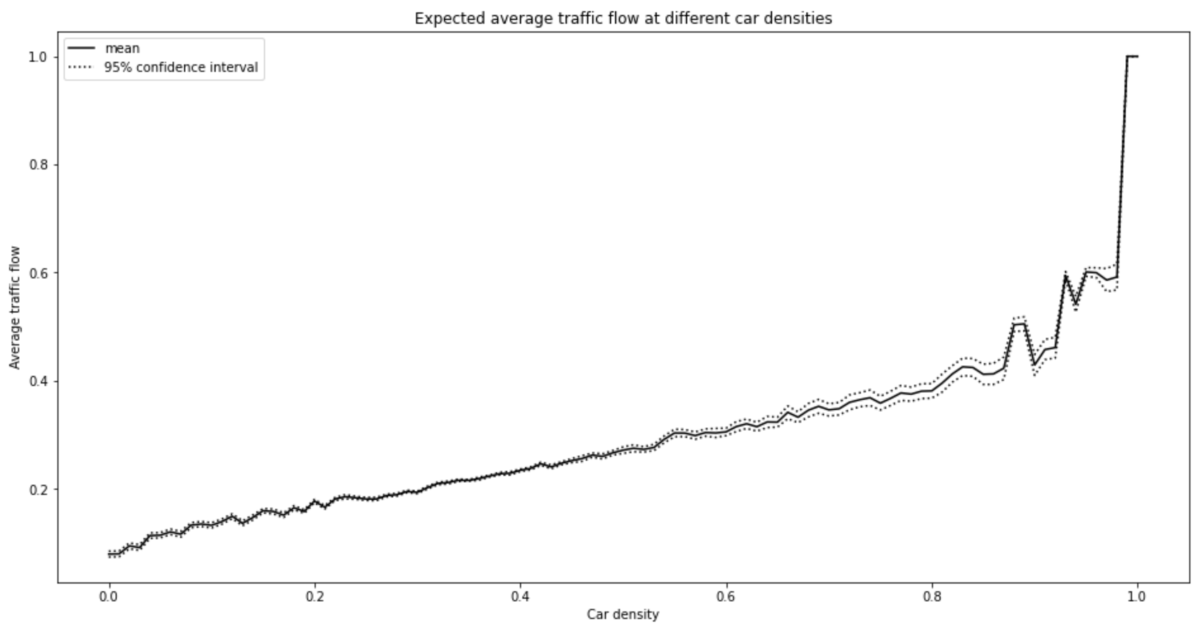

Figure 3. Mean and 95% confidence interval of average traffic flow given new traffic light strategy regarding different car densities

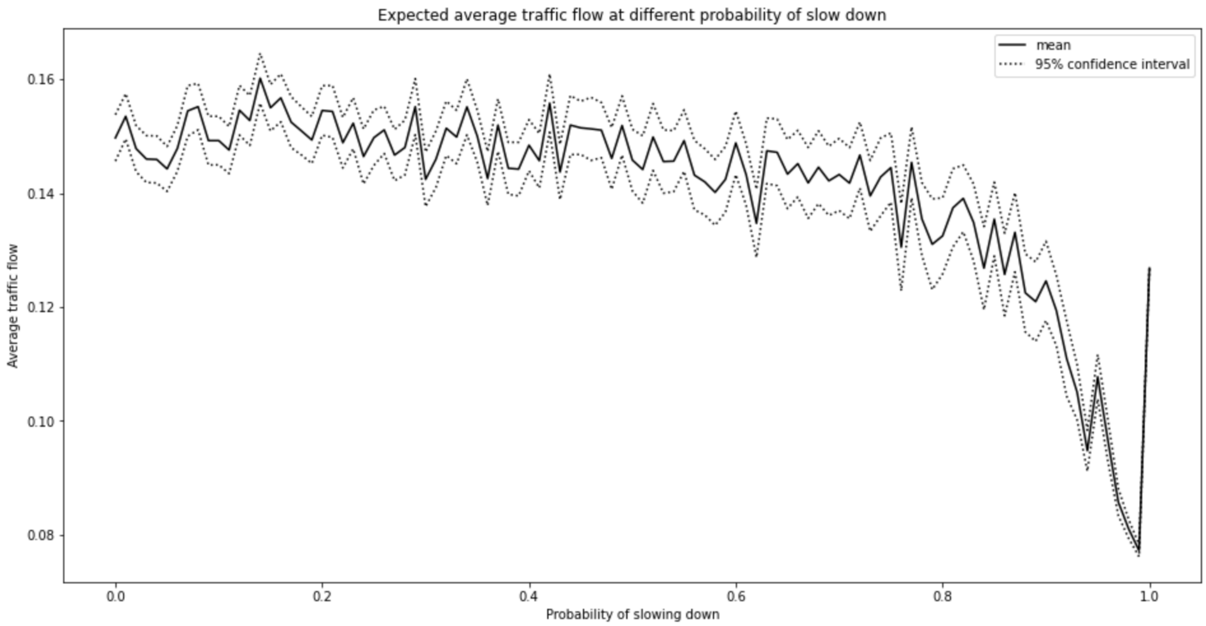

Figure 4. Mean and 95% confidence interval of average traffic flow given new traffic light strategy regarding different probabilities of slowing down

Critique of the model

Although the model correctly simulates the system behavior, we also acknowledge some drawbacks of the model that would decrease its representation of the reality. For example, the assumption of the probability of turning may not represent the true percentage of cars turning at the intersection. Also the time there's a chance that cars are stuck in the middle of the intersection causing traffic congestion. Therefore, an extension of the model would consider those complex situations and make the simulation more accurate.