# LBA

Hyderabad: Traffic Simulation

Minerva University
CS166: Modeling and Analysis of Complex Systems
Professor Ribeiro
March 20, 2023

**Table of Content**

**Introduction**

As I have learned in the past two months, traffic in Hyderabad, and India more generally, is no joke. Despite the enormous amount of vehicles on the streets, the traffic seems to rarely ever stop flowing. The motivation behind this paper is to analyze one of the main roads in Kondapur, the neighborhood where I live (Figure 1).

First the paper will describe how the relevant data was collected and then it will focus on modeling this block of the road to get insight on the role of the traffic lights and how the traffic flow could be optimized.

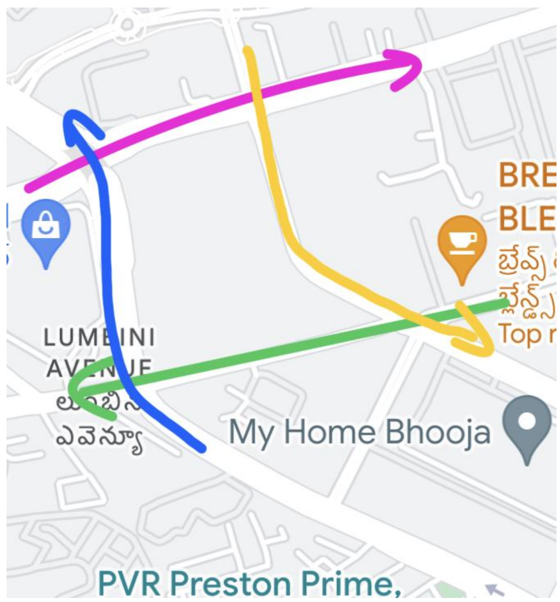



Figure 1. The intersection Hitech City Main Road that the paper will explore.

**Collecting Data**

I went to the intersections in order to collect data which I used to decide what are suitable values for the parameters of the model. It is important to note that this method of data collection is prone to human errors as I cannot count all relevant values efficiently and accurately.

Furthermore, the data collection was done on a Tuesday morning. I broke down the intervals into 5 intervals of 1 minute for each intersection to get several samples. However, it is likely that the traffic varies depending on the time and day of the week.

- The traffic lights were almost synchronized – the difference between the intersections was about 2 seconds which I assume is to give extra time for the cars to leave the intersections. For the purposes of abstracting the details, I will ignore this small difference. The traffic lights followed the main road (Hightech City Road) – when that one is green the roads that cross it have a red light so they were inverted.
- The green traffic light had a time span of 55 seconds.
- The area is not pedestrian heavy as these are main roads with no sidewalks – it looks more like a new construction area.

I collected separate data for each of the intersections.

Hightech City Main Road and Shipa Layout Road (blue and green intersection)

- There were 1 pedestrian in 5 minutes.
- Vehicles that turned left at each traffic light change: [5, 7, 7, 4, 3] – 26 in 5 minutes.
- Vehicles that turned right at each traffic light change: [9, 14, 8, 12, 5] – 48 in 5 minutes.

Hightech City Main Road and Inorbid Mall Road (blue and pink intersection)

- There were 5 pedestrians in 5 minutes.
- Vehicles that turned right at each traffic light change: [3, 0, 0, 1, 1] – 5 in 5 minutes.
- Vehicles that turned left at each traffic light change: [4, 8, 3, 6, 2] – 23 in 5 minutes.

Dallas Centre Road and Shipa Layout Road (yellow and green intersection)

- There were 0 pedestrians in 5 minutes.
- Vehicles that turned left at each traffic light change: [5, 4, 7, 2, 5] – 23 in 5 minutes.
- Vehicles that turned right at each traffic light change: [0, 2, 2, 0, 0] – 4 in 5 minutes.

Dallas Centre Road and Inorbid Mall Road (yellow and pink intersection)

- There was 1 pedestrian in 5 minutes.
- Vehicles that turned left: 7 in 5 minutes.
- Vehicles that turned right: 34 in 5 minutes.

After gathering this information, I was able to estimate what some of the parameter values should be. I decided that one time step in the simulation would represent one second of real time. The probability of a pedestrian showing up per time step at the first intersection is 0.003, at the second intersection 0.017, at the third intersection 0.000, at the fourth intersection 0.003. Furthermore, none of the intersections actually have crosswalks. Whenever you need to cross a street in Hyderabad, you have to wait for all the cars and then go. Since the probability of a pedestrian showing up is close to zero for all intersections and vehicles do not slow down because of them, I decided to leave the pedestrians out of this model. This can be a further improvement in my model which would allow for a more rigorous analysis and applicability to different contexts.

I also found the speed limit for those roads which is 40 km/h (Today, 2022). I used google maps to estimate the lengths of the relevant road segments that this paper is modeling, which is about 1 kilometer (1000 meters). To speed up the simulations I decided to divide this by ten so that I model a road length of 100. I also divided the speed limit by 10 leading to a maximum velocity of 4.

Furthermore, in India vehicles rarely slow down unless they have to stop because there is no more space in front of them. That is why the probability of random slow down is pretty low and I will set it to 10% to account for situations like random slow down, dogs, cows, pedestrians, or other animals crossing, as well as holes in the road. I will also keep the car density to about a third of the road since that is not a very busy block of road and there is plenty of empty space.

**Model and Simulation**

*Variables*

For the model we will be using a cellular automaton. It initializes a state full of grass with a side of size road_length (int).[1] After that it adds as many roads as our road_num (int) parameter. It rotates between a horizontal and vertical road (i.e. if there are four roads it will make two horizontal and two vertical roads at symmetrical locations in the grid). For each type of road it also rotates the direction (i.e. if there are two vertical/horizontal roads the first one will move up/to the right and the second one will move down/to the left).

The other parameters in the model include:

- car_density (float): the fraction of road cells that have a car on them – this parameter is used to initialize the state of the simulation at time step 0 – cars can only be positioned on road cells;
- max_speed (int): the maximum speed in car cells per update step;

[1] This is the formatting that Minerva uses for code in a Google Doc in other documents (i.e. the summer assessments for incoming students). That is why I chose this style to differentiate the variables from the rest of the text.

- prob_slow (float): the probability that a car will randomly slow down;
- green_light_time (int): the number of steps when the traffic lights will be green (i.e. they are green for the horizontal streets and red for the vertical streets or vice versa).

*Update Rules*

At each time step the state is updated according to the following rules:

- The cars can accelerate up to the max_speed. At each time step a car's speed can be increased by one if there are enough cells between the car and the first car in front of it.
- If there is a car in front that is moving slower, then the current car will have to slow down to the speed of the slower car.
- Cars may also randomly slow down by one with prob_slow.
- Cars move according to traffic lights. If the traffic light on their road is red, they can move up until that point (or up until other stopped cars) where they will stop and have a speed of 0. This means that even if the traffic light is red, cars will keep moving as long as they have free cells in front of them or until they reach a traffic light.
- Sometimes cars can turn left or right (depending on the direction of the road). This event occurs whenever a car is stuck at an intersection at the exact time when the traffic lights change lights from green to red.

*Assumptions*

The model makes several assumptions:

- The model assumes that the distribution of the car speeds are uniformly distributed in the range between 0 to the max_speed. We do not know the real distribution of how cars move. In reality most cars drive at a speed around the speed limit unless they have to slow down for some reason (i.e. cross walk, to turn, etc.). An improvement would be to model the speeds using a more informed distribution.
- All cars follow the rules on the road. This is especially not true for India where vehicles drive in any direction they want to from my experience. Vehicles also do not always drive in the designated lanes but anywhere where there is space. Drivers often do not wait at red lights but slow down and then keep driving if there is an opening for them.
- All vehicles are of the same size – each car takes up a single cell on the grid. However there could be many different types of vehicles on the road (i.e. motorbikes, tuk-tuks, cars, trucks). In India the most common vehicles are motorbikes and tuk-tuks which do not take up the whole width of a lane which is why there are often many more vehicles parallel to each other.
- The simulation assumes that drivers react immediately and can stop at any point. Usually drivers have slower reaction times. They are also not able to stop suddenly. This is why car accidents cannot occur in the model.
- If a car is stuck at an intersection when the traffic light for its road switches from green to red, it is assumed that it always turns left or right depending on what direction is possible. This simplifies the code but in reality this may not be the case. Furthermore, the model can be expanded to include a parameter for the probability for a vehicle to turn left or right.

Considering all of these assumptions, the model represents a limited version of reality but is powerful enough to provide us with insights about the overall traffic flow.

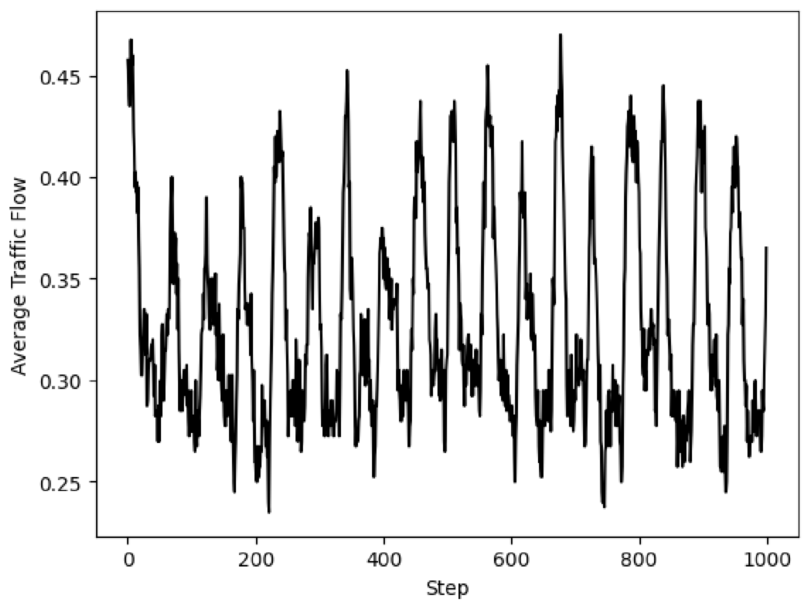
**Analysis**

This section will explore empirical results by running the simulation.

*Average Traffic Flow (with Traffic Lights)*

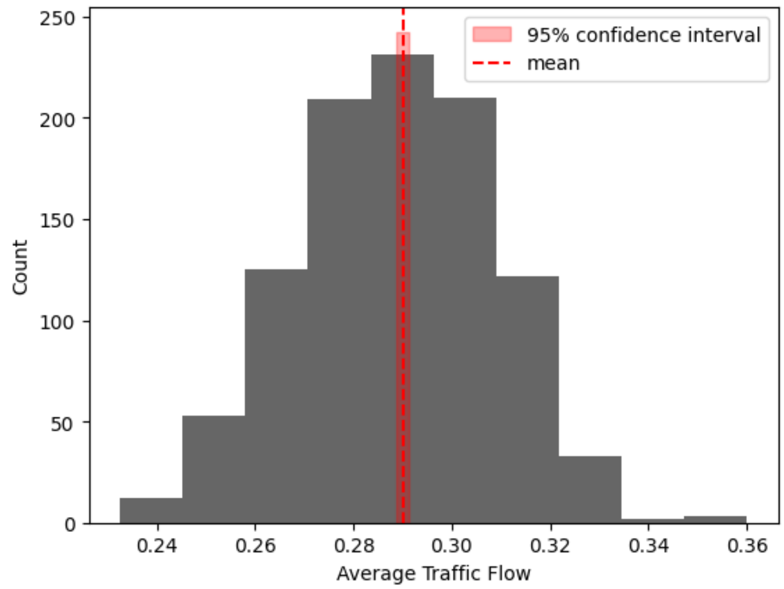
Using all of the gathered data, first I explored how the average traffic flow changes with traffic lights in Figures 2 and 3 below.

**Figure 2.** Average traffic flow as the simulation converges.




*Note:* The line plot shows how the average traffic flow changes over time as there is a small to medium fluctuation that makes the state oscillate.

**Figure 3.** Distribution of the average traffic flow from 1000 simulations.




*Note:* The distribution over 1000 simulations after 100 updates is shaped similar to a normal distribution (CLT).

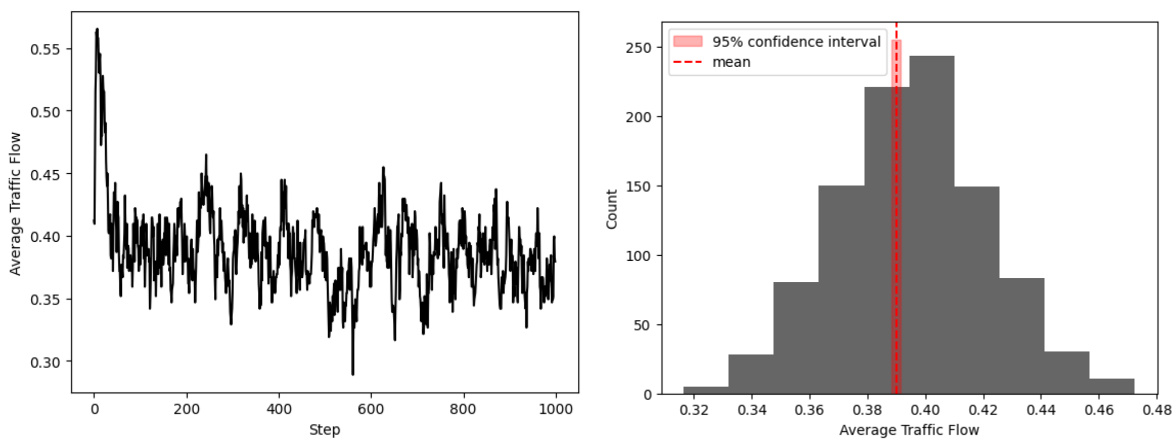
From Figure 1 we can see that the traffic flow oscillates and there are about 4 peaks approximately every 50 time steps. The fluctuation of the traffic flow is not too big – about 0.15. The fluctuations are not even which is expected since this is a stochastic model. These fluctuations are partly caused by the traffic lights: right after a switch from red to a green light cars have the biggest chance to be moving causing the peaks. As the time step gets closer to a time to switch the traffic lights, most cars are waiting at them or behind other stopped cars.

The traffic flow increases until about the mean average value and then goes down creating the oscillation pattern. From Figure 2 we can see that the sample mean is 0.29 with a 95% confidence interval of $(0.289, 0.291)$. The interval is narrow because the sample is big (1000 data points) that are simulations which have mostly converged (after 100 updated time steps). We can see that the distribution resembles the normal distribution. This is because we are observing the effect of the central limit theorem since we are plotting the average traffic flow of 1000 identical and independent trials.

*Average Traffic Flow (no Traffic Lights)*

I also wanted to see if the traffic lights actually help – you can see Figure 4 that is the same as Figures 2 and 3 above but without the traffic lights

**Figure 4.** Average traffic flow as the simulation converges (left) and distribution of the average traffic flow from 1000 simulations (right) with no traffic lights.



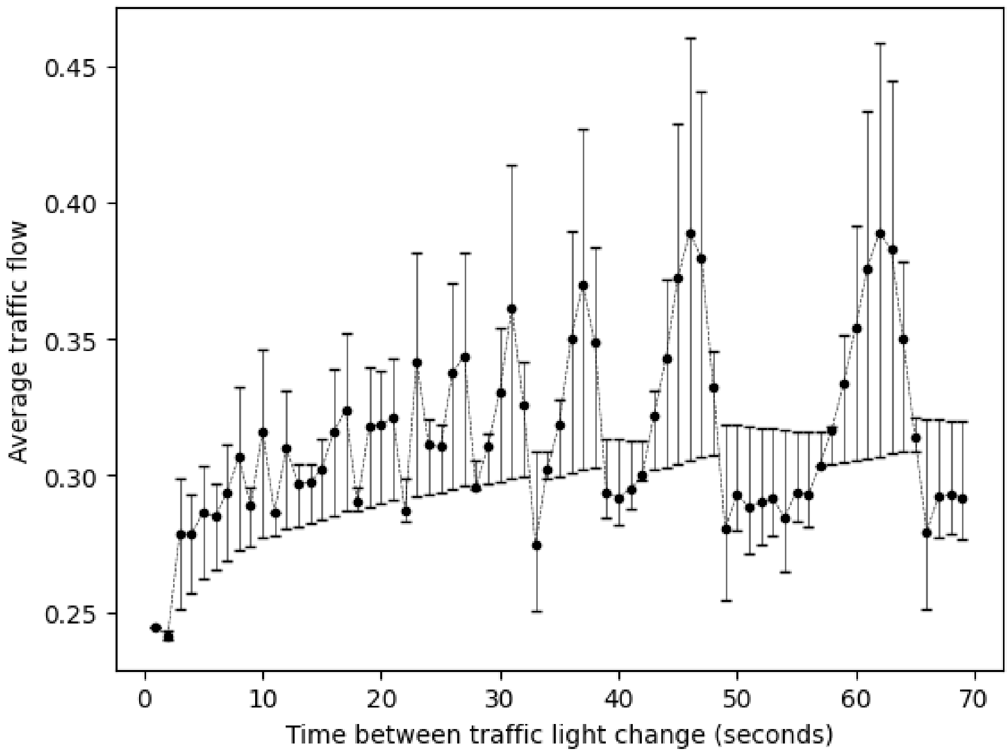

*Note:* The oscillating pattern on the left is similar to what we observed in Figure 2 and the shape of the distribution is also similar to Figure 3.

If we compare the results from Figure 4 with those in Figures 2 and 3 we can see that they are similar. The traffic lights seem to make the oscillating pattern less frequent (a longer period of time between the peaks). Again the line plot on the left of Figure 4 oscillates about the mean of the average flow. From the histogram on the right of Figure 4 we know that the sample mean is 0.39 with a 95% confidence interval of $(0.388,\ 0.392)$. From these results it seems like the traffic lights actually get in the way of the system and the average flow would be higher if we simply left the cars without the guidance of the traffic lights. We cannot fully trust these empirical results. This is because the model is biased towards having intersections without traffic lights. In real life when a driver approaches an intersection they slow down (usually there is a stop sign) where they have to look for coming cars and only then begin accelerating again to continue moving. In the model there are no such conditions. Cars keep driving as long as there is space in front of them which leads to a higher average traffic flow.

*Optimal Traffic Light Timing*

After this I was interested in seeing how the average traffic flow changes as the time of the green light of the traffic light increases.

**Figure 5.** Error bar plot of the time between traffic light change vs the average traffic flow.



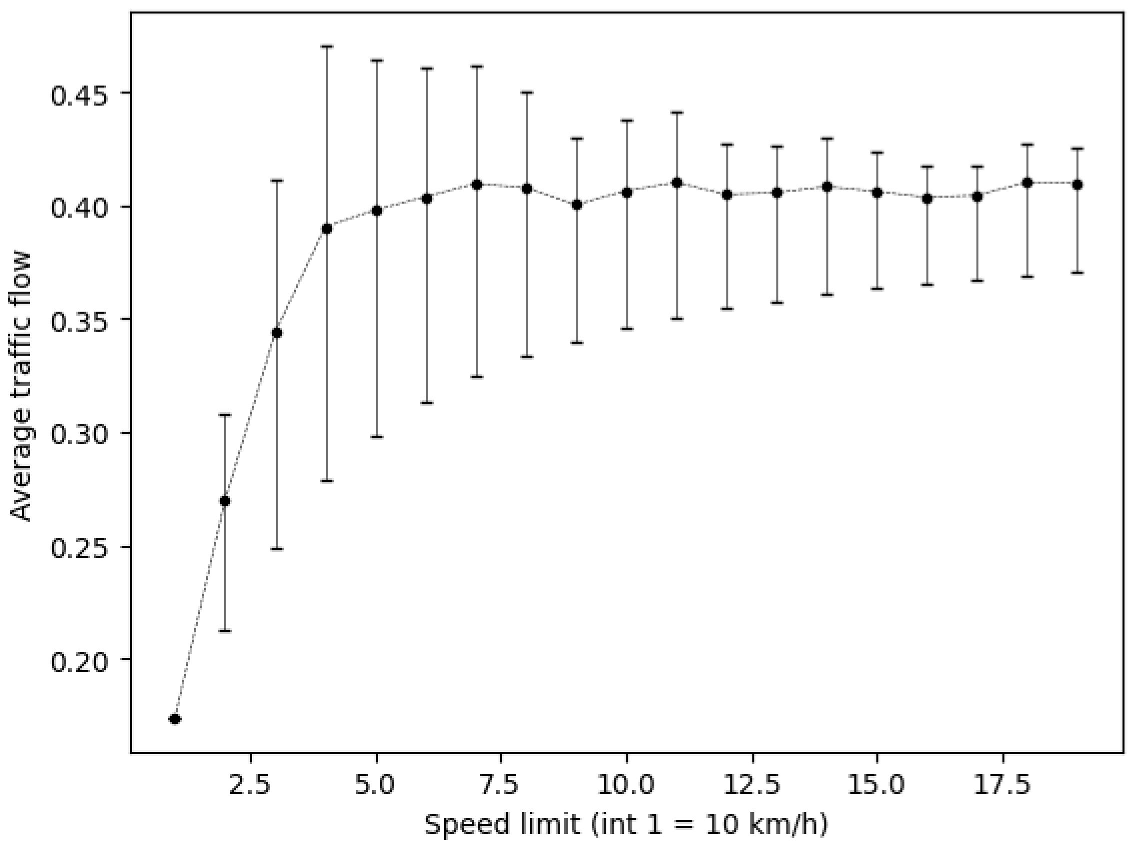

*Note:* The plot was produced by taking the average value of 100 trials that were updated for 200 steps for each point .

In Figure 5 above we can see how the average traffic flow changes as the time between the traffic light switches increases. The error bars are wide but that is because the plot is produced by making only 100 trials (compared to at least 1000 trials in the previous figures). We can see that there is not a strict linear relationship between the independent and dependent variable. I thought that there would be an optimal value for this indicated by an increase leading to peak traffic flow and then descending back down, which does not seem to be the case. Instead, we are again observing the oscillating pattern as previously seen in Figures 2 and 4. We can see that the two peaks are when the traffic light duration is approximately 46 seconds or 62 seconds. The current value of the parameter is 55 seconds which is what I gathered from my data collection. These results indicate that we can either lower or increase the time of the green lights by about 7 seconds to improve the flow through the roads.

*Optimal Speed Limit*

Something else I was interested in exploring was what the optimal speed limit would be. I created a similar error bar plot as Figure 5.

**Figure 6.** Error bar plot of the speed limit vs the average traffic flow.

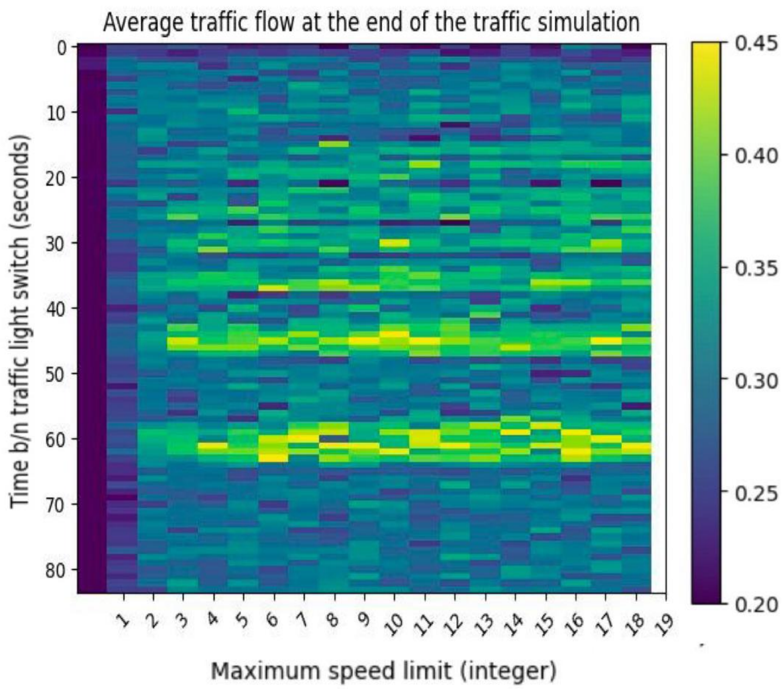


*Note:* The plot was produced by taking the average value of 100 trials that were updated for 200 steps for each point .

In Figure 6 we can see that there is a logarithmic relationship between the speed limit and the average traffic flow. As we increase the speed limit, the traffic flow increases to about 0.4 after which it plateaus and remains around that value. For an optimal speed we could pick a value around 6 = 60 km/h. This is because in real life we also need to consider safety so we do not want drivers to be too fast which increases chances of accidents. A speed limit between 50-60 km/h is standard for many big cities especially on bigger streets so this value seems realistic.

Since in Figure 6 I had to keep the green light time parameter constant, I wanted to find the optimal parameter configuration so I also created a heatmap (see Figure 7 below).

**Figure 7.** Heatmap of average traffic flow at the end of the traffic simulation (speed limit vs time between traffic light switch).



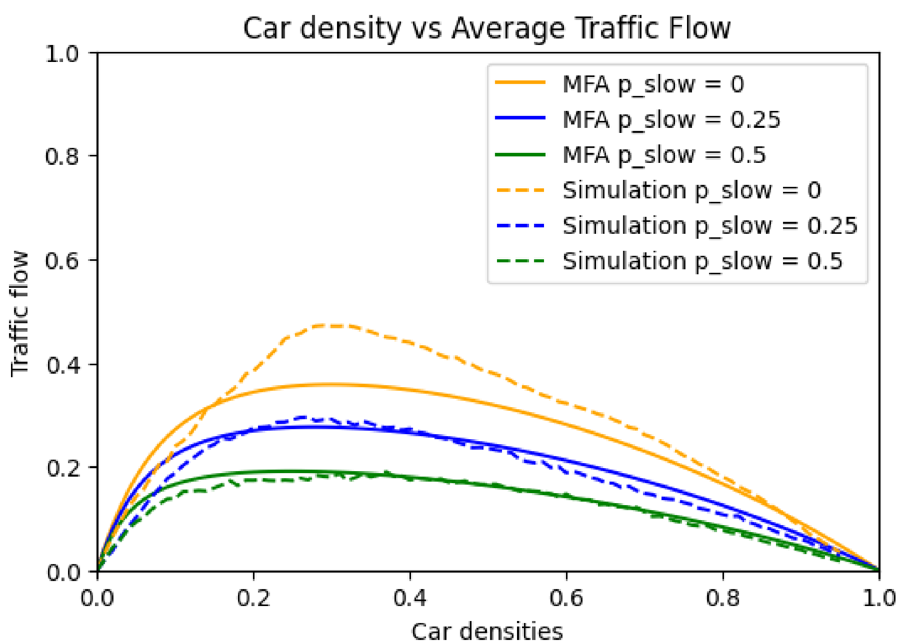

*Note:* The heatmap allows us to explore the optimal configuration. It's important to note that on the x-axis one step is equal to 10 km/h in real life.

If we explore the heat map of average traffic flow at the end of the traffic simulation represented as the speed limit vs time between traffic light switch, we can see that the pattern reminds us of the oscillating pattern of Figure 5 (the error bar plot of the time between traffic light change vs the average traffic flow). There are two horizontal strips that draw our attention: when the y-axis is about 45 and 60. This is where the average traffic flow is the highest. The maximum speed limit does not seem to have as much of an impact. As long as it is above a 4 the average values are similar. This is consistent with our results in Figure 6 where we saw that the average traffic flow reaches a peak when the speed limit is 5 and then remains pretty constant. From this plot we can also see that the time between the traffic light switches is actually more optimal at about 60 rather than 45. This is because even at lower speed limits the traffic flow is higher – therefore we also assure a higher level of safety.

*MFA: Theoretical Analysis*

Now let's use mean field approximation – a mathematical model underlying this scenario, to compare our empirical results with.

**Figure 8.** Car density vs Average Traffic Flow (MFA and Simulation results).




*Note:* The graph compares the theoretical results from the MFA and those from the simulation for different car densities and different probabilities of random slow down.

In Figure 8, we compared the results from the simulation and those from the MFS using several probabilities of slowing down and car densities. The general trend is that the results are similar.

The results from the simulation have somewhat of a sharper peak compared to the smooth MFA results but the difference is not that big. Something else that we notice is that as the probability of slowing down increases, meaning as we add randomness, the plots become more wiggly.

Something important to note here is that usually, as we have seen in class, the MFA underestimates the average traffic flow. However, here it is not really the case or the difference is much smaller. This is because in our simulation now there are traffic lights. Therefore cars have a higher probability of slowing down – it does not just depend on the random slow downs. The MFA does not take that into account. I tried to include this in the MFA but my attempt was unsuccessful. The probability of a car being at a traffic light is the average velocity over the length of the road (or the number of traffic lights divided by the length of the road). However the MFA does not take into account that road length which is why I left it out.

*Queuing Theory: Theoretical Analysis*

Another way to model and analyze the given scenario is by using queuing theory. Each intersection in a road can be perceived as a queue; each traffic light as a server that is deterministic, predetermined and constant (D). This serving time is the average time a car spends at the traffic light at a red light waiting (for the light to switch to green and 'to be served'). Furthermore, we currently do not know the distribution of speed of the cars which proceed to move stochastically when the light turns green (after they have been served). This is why the arrival rate is unknown (G). I assume an appropriate distribution could be chosen by studying how vehicles move and/or by analyzing real time empirical data to inform us about the best fit. Having said this, it means that each intersection can be modeled as a G/D/1 queue. This can be further expanded. If there are two parallel roads then we could model them as G/D/1 $\times$ 2. After modeling the system in such a way we can utilize different formulas to study the waiting time – how long vehicles have to wait at traffic lights, and (maximum) queue length – how many cars wait at each intersection. This can help us find the optimal configuration so that we minimize the waiting times as well as the average lengths of the queue meaning shorter traffic jams.

## Conclusion

Ultimately, this analysis helped us get a better understanding of one of the future city centers of Hyderabad – Hightech City Main Road. To improve the model we could extend it further to allow for multiple lanes, specific right and left turn probabilities, as well as pedestrians in order to generalize it and even potentially be able to optimize the whole traffic network.

**WORD COUNT:** 3300 words

# Appendix

***Appendix A:*** *LOs*

**#Modeling**

I applied this LO in the Introduction, Collecting Data and Model and Simulation sections. I included a thorough description of the model scenario, rules, parameters, and assumptions. I conducted research dn data collection to estimate the most appropriate values for the parameters. Furthermore I explain all of the variables and parameters as well as the update rules. I also critique some of the assumptions of the model.

Word count: 67

**#PythonImplementation**

I extended the code from class to create a working implementation of the model in Python. It follows all of the relevant update rules and assumptions. I also use appropriate data structures throughout the implementation. For example, I use a list to store the average traffic flow history. For the state I use a 2-dimensional numpy array that allows for the updates at each time step. The state of the traffic lights is represented as a boolean variable: True when green and False when red. What is more, I created an animation that makes it easy to visualize the state of the simulation and to follow through all of the updates. The code is broken down into different functions that are reused. I also included many different tests to show that the model is working as expected (i.e. testing the grid generation, horizontal roads, vertical roads, intersections, then both at the same time). The tests build on each other. To see some of the test cases please view the ‘My Code’ section in the Jupyter notebook.

Word count: 177

**#CodeReadability**

All of the code is well documented. There is an appropriate level of comments that makes it easy to follow the logic throughout the code. All of the methods have docstrings. I also formatted the Jupyter Notebook with titles and explanations. The variable names are also descriptive and make it easy to follow the code used to generate the graphs as well.

Word count: 62

**#Professionalism**

The report follows an appropriate format, with good grammar, and organization. There is a table of contents that allows for easy navigation together with the page numbers. There are also clear and concise titles and subtitles for all sections. All graphs have corresponding figure numbers, titles, labels, units, and descriptions.

Word count: 177

**#EmpiricalAnalysis**

I applied this LO in the Analysis section. After implementing the model in Python, I used it to explore the scenario and gain valuable insights. I created appropriate plots that allowed me to determine the optimal set of parameters given the context. I ran each of the experiments with a big number of trials. I also included relevant stats like confidence intervals (on the histogram and error bar plots) as well as the mean values. I explain how the central limit theorem was relevant.

Word count: 84

**#TheoreticalAnalysis**

I applied this LO in the MFA and Queuing Theory sections. I used MFA to model how the traffic flow changes as the car density increases and compared it with the empirical results. This was useful because the roads I am modeling get much busier at rush hours. I also explain how queuing theory is relevant and how we can utilize it to get more insights.

Word count: 66

***Appendix B:*** *HCs*[2]

*#sampling*

I applied this HC together with the #EmpiricalAnalysis LO. In order to get useful data to then interpret and gain insights, I had to choose an appropriate sampling method. The goal was to get samples that are representative of the whole population. To do this I chose random sampling by creating a large number of trials. Each trial was updated 200 times in order to make sure that the result is not influenced by the initial state and that all such fluctuations have been washed out by then – this means that the simulation has converged to a stable state. Ultimately, this method of sampling allowed me to get accurate results and explore different patterns.

Word count: 114

*#confidenceintervals*

I applied this HC together with the #EmpiricalAnalysis LO. I accurately computed the confidence interval of the average traffic flow for several different samples. I also effectively include it when relevant (i.e. in the histogram and in the error bar plots). I also interpret those values and compare them. For example, in sections Average Traffic Flow with and without traffic lights I explain why the upper and lower bounds of the confidence interval are higher than when we include the traffic lights. I also justify why the intervals are narrow since we have a large number of trials.

Word count: 114

[2] I tagged some HCs for transfer scores. Please only score HCs if they are at 3 or above.

## References

Today, T. (2022, August 26). *Here are the speed limits for different roads in Hyderabad*. Telangana Today. https://telanganatoday.com/here-are-the-speed-limits-for-different-roads-in-hyderabad