Optimizing Urban Traffic Flow

Ben Katzman

Complex Systems 270

December 4, 2023

The intricacies of traffic flow present a difficult challenge in urban environments, impacting both commuter experience and city planning. Traffic brings about a multitude of problems, such as wasted time, increased pollution, and economic inefficiencies. However, there is no quick fix to this problem, as there are a countless number of factors that influence traffic flow. To address this issue, a model has been developed in order to simulate and analyze traffic dynamics. This model incorporates a myriad of factors that may affect how efficiently cars arrive at their destination and the number of traffic jams. These factors include the number of cars on the road, the number of roads, the number of different destinations, whether or not the stoplights are synchronized, and the probability of the lights changing. Using these components, I aim to optimize traffic management strategies by finding combinations of these variables that get all cars to where they need to be as quickly as possible while minimizing traffic jams.

The study of traffic flow dynamics has been a subject of extensive research in previous years and a number of different methods have been deployed. The simplest of which is intersections were simply watched, and the number of cars that passed through were manually counted. This method is incredibly time consuming and does not give a holistic view of the entire ecosystem of cars. However, this technique has been improved over time using cameras or bluetooth trackers in order to get a sense of all the cars in an area and where they are. This is very effective for collecting real world data, but is not quickly replicable. Furthermore, using this method does not allow for the testing of new ideas that may improve traffic flow.

Due to the limitations of real world observations, simulations have often been used to model traffic flow. According to Storani et al, there are three typical kinds of traffic models: macroscopic, mesoscopic and microscopic (Storani et al). Macroscopic models view traffic flow at an aggregate level, focusing on overall traffic characteristics without considering individual vehicle behavior. Microscopic models focus on individual vehicles, simulating the behavior of each vehicle in the traffic flow. Mesoscopic models bridge the gap between macroscopic and microscopic models by considering traffic flow at an intermediate level of detail. For the purposes of my simulation, I will be using a microscopic model that focuses on the decisions of individual vehicles.

The model I created is agent based and uses a grid format. The first step is to initialize the board. This first means creating the roads, which span across the board and are 7 units wide. Each road contains two lanes, each 3 units wide, so that a car can remain in the middle of its lane. These roads loop around, meaning that when a car reaches the edge of the board it can move forward and begin again at the opposite side. The next step is to create stoplights, which all begin synchronized, that appear every time the roads intersect. I then randomly added destinations, each of which is input at a random location at the side of a road, but not next to an intersection. The last step in initializing the board is adding cars. Each car is randomly assigned a road and a side of the road. Given the side of the road, each car is given an orientation, as all cars on certain sides of the road must move in the same direction. Lastly, all cars are randomly assigned a spot on the road, each at least one spot away from another car and never initially in an intersection.

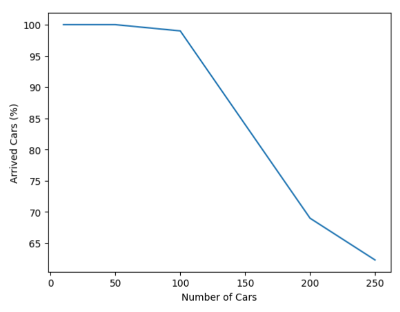
The next step is to begin the simulation, iterating over a given number of timesteps. For each timestep we update the traffic lights and then the cars. When updating the traffic lights, there are two options: synchronized or random. If they are synchronized, all traffic lights will change in unison. If random, the traffic lights will change independently of each other. For each timestep, we will randomly check whether to change the lights according to the given probability. If the light is changed, the green will turn to yellow for 5 timesteps, instructing the cars that are not in the intersection to stop and allowing the cars already in the intersection to clear out. Once this concludes, the red lights will turn to green and the yellow lights will turn to red.

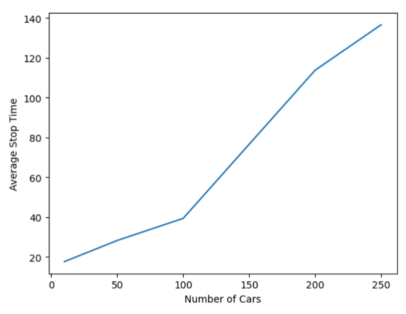
Next we must update the cars, which is done in a random order for each time step. The main objective of the car is to get to its destination using the fastest route possible. This is done in four phases, which I call driving, intersection, final road, and arriving. In the driving phase, the car simply moves forward, unless there is another car ahead. If at the end of the movement it sees a traffic light, it transitions to the intersection phase. In the intersection phase, the car first looks at the light. If it is red or yellow, the car stays put, and if the light is green, the car moves forward. Once in the intersection, the car measures which direction is the smallest distance to get to the destination, and turns in that direction. If the road the car turns contains its destination, it enters the final road phase. Otherwise, it will return to the driving phase. If the car is in the final road phase, the procedure is similar to that of driving, moving forward and stopping for red lights and cars in front of it. However, when the car becomes parallel to its destination, it will stop and enter the arriving phase. For the arriving phase, if the destination is on the right it simply turns right and moves towards it. If the destination is on the left, the car must first check that the opposite lane is clear before moving towards it. Once the car reaches the destination, it is removed from the board.

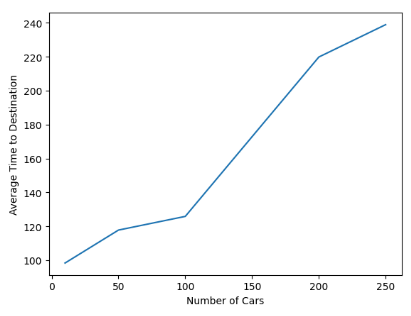
When judging how successful a simulation is, I will be using three metrics for measurement. The first metric is the percentage of cars that arrived at their destination. This metric is simple and shows how effective the format is at getting cars where they need to be. The next metric is the average stop time across cars. This tells us if many cars are either waiting at a lot of stop lights or stuck in a traffic jam. The last metric I will be using is the average time to arrive at the destination. For cars that did not arrive at their destination, their time is listed as the number of timesteps run plus a penalty. This penalty is the Manhattan distance (the distance between two points measured along axes at right angles (Educative)), which symbolizes the quickest possible amount of time the car could have arrived at the destination, assuming no stoppages. This lets us know not only the rate at which cars are arriving at their destination, but the efficiency at which they are doing so.

When testing the model, I ran a multitude of tests, attempting to isolate each variable to decipher the best setting for each. For simplicity, I set the number of timesteps to 300 to allow most cars to reach their destination, the grid size as 100 x 100, and the number of roads equal to 6 (3 across the x axis and 3 across the y axis) across all trials. When testing a variable, I set it equal to a couple different values, testing each 10 times and taking the average over all runs.

The first variable tested was the number of cars on the road. For this variable I tested the values of 10, 50, 100, 200, and 250 cars. Below are tables showing the exact values:

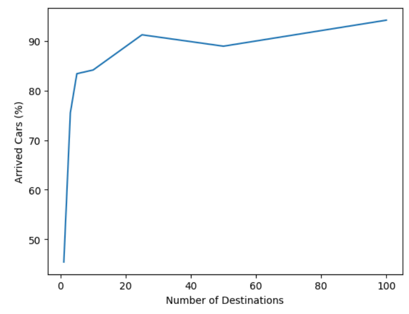


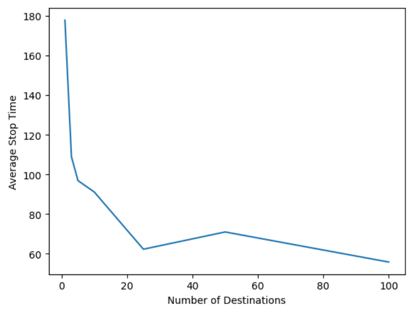


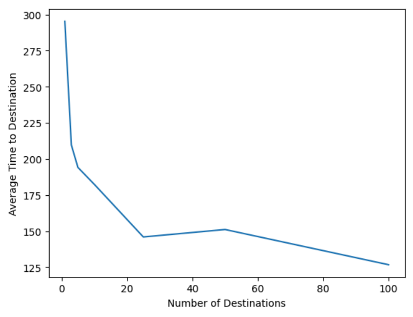


Predictably, all metrics were significantly worse as the number of cars increased. However, in real life we often will not be able to control the number of cars on the road. Thus we will proceed with 200 cars, in order to see emergent behavior out of other variables when traffic density is high.

The next variable we will be testing is the number of possible destinations a car can go to. For this variable I will be assigning values of 1, 3, 5, 10, 25, 50, and 100. Below are tables showing the exact values:







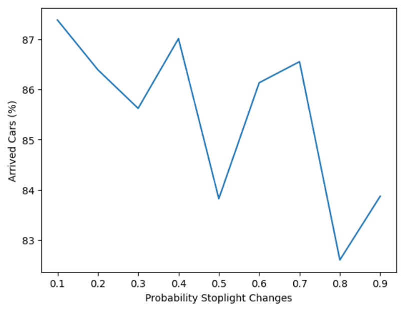
All metrics seem to get better as the number of destinations increases, although they all seem to level out around 10 destinations. That being said, once again the number of different places people are going is not something that can be controlled in real life. Thus, we will be proceeding with 10 destinations for the remainder of the trials.

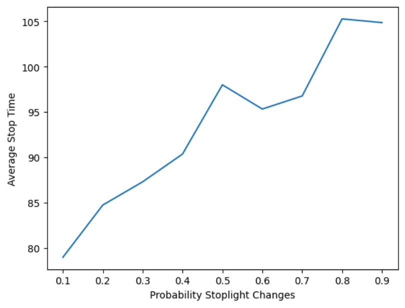
The next factor I will be looking at is whether the stop lights are synchronized or not. Because there are only two options, I will be increasing the number of runs to 50 for each option. Below is a table showing the result (all values rounded to 2 decimal points):

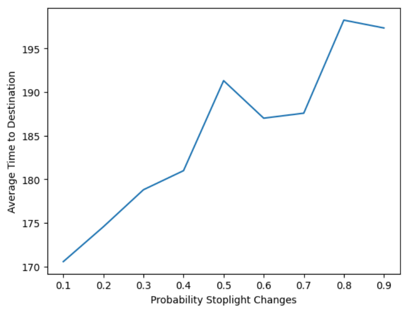
	Arrived Cars	Average Stop Time	Average Time to Destination
Lights Synchronized	90.52%	163.57	75.52
Lights Not Synchronized	87.38%	172.18	82.50

From this, we can tell that the synchronized lights provide a slight but definitive upgrade in efficiency over the lights not being synchronized. This is most likely because light synchronization prevents unnecessary stoppages, as a car can continue going straight without interruption once they hit a green light. Due to this, we will be moving forward with synchronized lights.

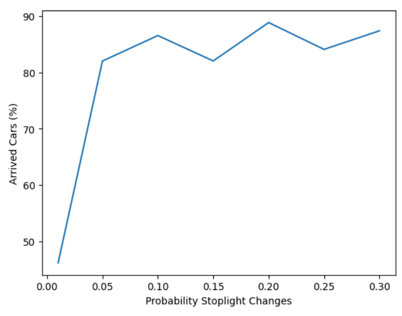
The last metric we will be testing is the probability of the lights changing in a given timestep. For this probability I will be assigning values of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. When running this I got extremely variable behavior, so I increased the number of runs to 50 in order to get the best results possible. Below are tables showing the results:

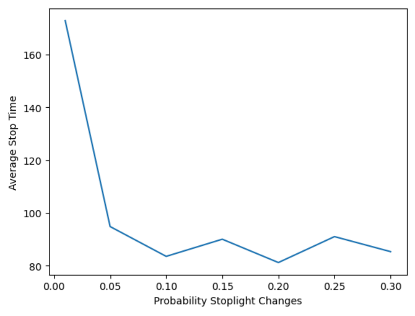


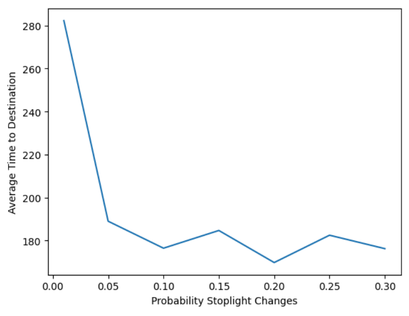




These results show an overall decrease in efficiency as the probability of the stoplight changing increases. However, since the result has such a high variability, especially in the percentage of cars that arrive at their destination, I decided to do a more targeted trial. I assigned new values to the probability of a light change, this time using values 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3. Below are the results from these trials:







From these results, we can conclude that the optimal probability for the stoplights changing is somewhere between 0.1 and 0.2. This means that to optimize traffic efficiency, we would like for the lights to change less often. This is likely because infrequent changing of the lights help to prevent the unnecessary stoppages.

In summary, this study looked into the different factors that affect the flow of traffic, searching to find a method that would increase efficiency. Through meticulous exploration and analysis, the model revealed critical insights into factors that best optimize traffic flow. When looking into the results, it is best to have less cars on the road with a greater number of destinations they wish to visit. However, both of these factors cannot be controlled in real life situations. The factors that can be controlled are the synchronization of traffic lights and the probability that they change. The results indicate that traffic flow improves when stoplights are synchronized and the probability of them changing is in the range of 0.1 to 0.2. The insights gained from this study offer promising avenues for traffic management strategies, offering potential solutions to mitigate congestion and reduce travel delays.

Works Cited

“Analysis and comparison of traffic flow models: a new hybrid traffic flow model vs benchmark models - European Transport Research Review.” *European Transport Research Review*, 24 November 2021, https://etrr.springeropen.com/articles/10.1186/s12544-021-00515-0#citeas. Accessed 3 December 2023.

“What is Manhattan Distance in machine learning?” *Educative.io*, https://www.educative.io/answers/what-is-manhattan-distance-in-machine-learning. Accessed 3 December 2023.