CS166

Abdul Qadir

Traffic Simulation

Part 1: Traffic jams on a circular road

In the first part of our assignment, we had to implement the Nagel Schreckenberg traffic model on a single lane with periodic boundary conditions (which is the equivalent of a circular road). Varying the density of the cars on a lane gives us different patterns on how traffic jams form (see Figure 1 below).

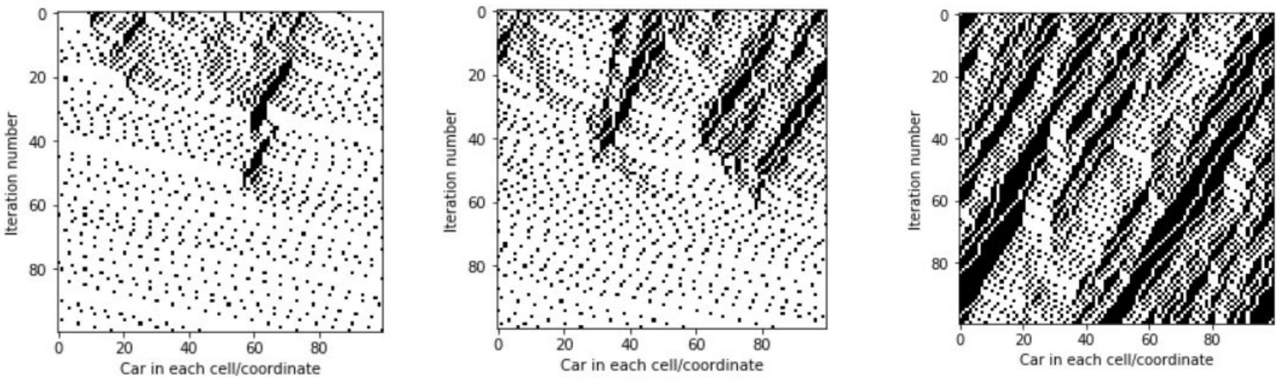

Fig 1. Shows how traffic flows over different densities. The leftmost graph has an initial density of 0.2, the middle graph has a starting density of 0.4, the graph on the right has a density of 0.5[1].

The simulations carried out in Figure 1 were done with a highway of 100 cells, varying densities, a maximum velocity of 5, and 0.3 as the probability of a car randomly slowing down. We can see from the results that higher traffic densities form jams that persist throughout time, while traffic with lower densities tends to get rid of traffic jams over time.

[1] #dataviz: Generated a data visualization to effectively portray how traffic flows through a system with different densities. The visualization makes it much easier to see the flow of traffic rather than printing large lists of different velocities.

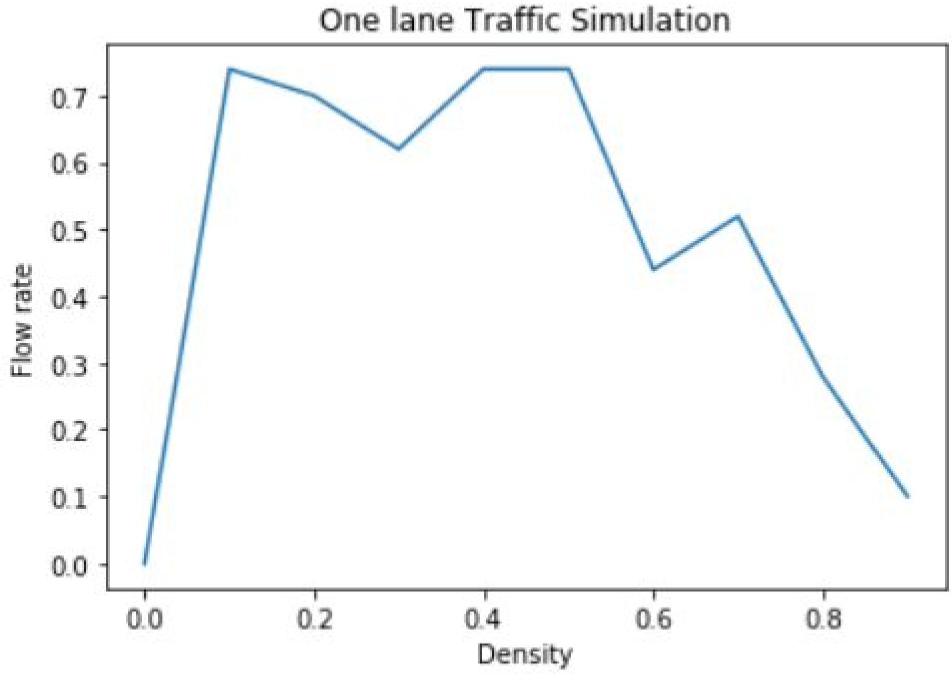

Fig 2. Showing how the flow rate varies over density.

Overall, the flow rate seems to increase linearly till 0.1, and then tip over, and average downwards till the density becomes 0.1. However, there is still a jump around 0.3 to 0.4 on the Y axis, which may just be due to randomness. The results are similar to the original paper where the flow rate starts decreasing at ~0.08 density.

Part 2: Multi-lane highways

In part two of our assignment, we had to extend our model to two lanes. In order to cater for lane switching, new rules were introduced based on the original paper. Specifically, the rules for each car are:

1- Switch Lane or Not

1. Check ahead of yourself if you can move (v+1 steps where v is your current velocity)
2. If you can't move, then check if there is space on the adjacent cell on the other lane
3. Check ahead on the other lane (v+1 steps) if you can move

4. Check behind you so you do not get in the way of a car. How far you should check behind is based on the maximum velocity parameter.
5. Switch lane with a probability p (a parameter you can define in the model) if condition 1 is false, and condition 2,3, and 4 are true.

2- Carry out movement updates

1. Accelerate your speed by 1
2. If there is a car in your way, slow down to the point behind the car e.g. if your speed after accelerating is 3, and there is a car 3 steps ahead of you, slow down your speed to 2.
3. Slow down by 1 based on the probability P (which is a parameter you can specify in the model). This makes the model stochastic and imitates real human behaviour since not all humans are perfect drivers.
4. After the speed is calculated, move the number of steps of your speed.

When initializing the model, one has to define a few parameters. Specifically, the length of each lane (number of cells on each lane), the density of cars on the road, the maximum velocity a car can have, the probability of slowing down, and the probability of whether you will switch lanes. In the model, we assume that the probability of switching lanes and the probability of slowing down mimic human like behaviour, since not everyone always switches lanes (e.g. they might be new drivers who are afraid), and people are not robotically efficient drivers (e.g. the reaction time to brake may be slow)[2].

[2] #modeling: Programmed a model for traffic simulation to generate predictions as to how traffic flows with different traffic densities, and also calculated the flow rates for different densities using the same model. Explained in the last section the limitations of the model, where it could be applicable, and how it could be improved.

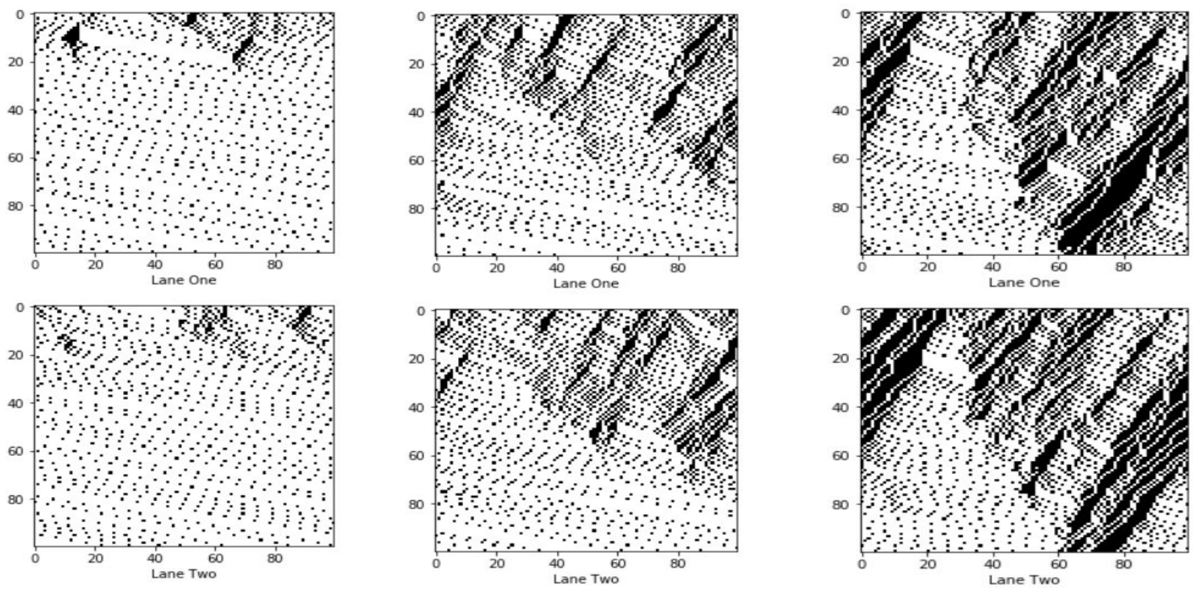

Fig 3. Shows how traffic flows in two lanes with various densities. The leftmost graph has an initial density of 0.2, the middle graph has a starting density of 0.4, the graph on the right has a density of 0.5.

The results from figure 3 are similar to the results from a one lane simulation, except when the density is 0.5 in the two lane simulation, traffic jams are not as many compared to the one lane simulation.

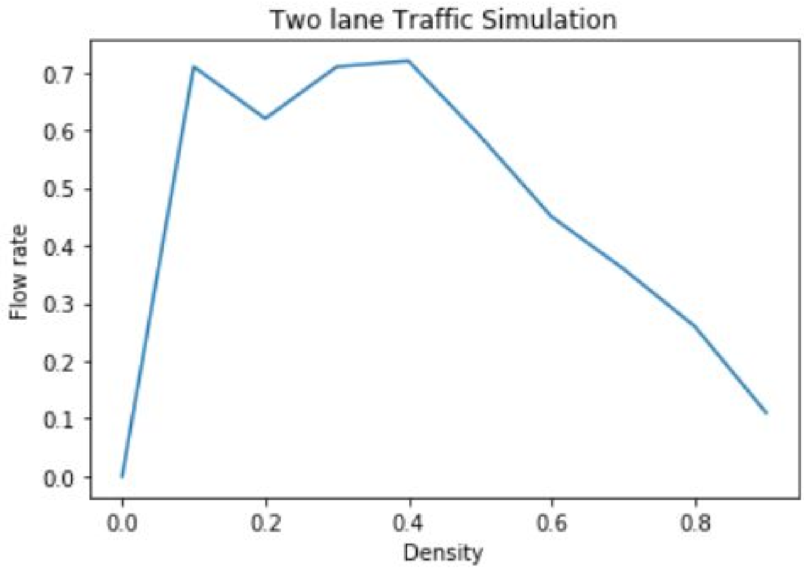

Fig 4. Flow rate vs Density for the two-lane traffic simulation.

The traffic flow rate in the two lane simulation is also similar to the traffic flow in a single lane. I believe this conclusion could be extended to more than two lanes. The reason being that the traffic density determines the number of cars in each lane, and the density for each lane does not change. For example, if the flow rate is 0.5 for a 100 meter (cell) highway with a density of 0.6 (60 cars), then adding another lane with a density of 0.6 simply adds 60 cars to the total system, without really affecting how traffic flows. On the other hand, if you just added another lane and kept the number of cars constant i.e. two lanes with 60 cars instead of two lanes with 120 cars; then the flow rate could be higher as its more likely for the highway to have empty spots after crossing the periodic boundary, and then cars are more likely to cross over (which would imply a higher flow rate).

Applicability to Berlin:

It is hard to generalize the model to traffic in a city because of a few assumptions. Firstly, traffic is unidirectional. The second and more major assumption is the lack of cyclists in our model. Berlin's roads are filled with people using cycles, and without having a cycling lane or cyclists in our model, I believe our current model will not be applicable to Berlin at all.

However, it could be applied to some specific areas where traffic is unidirectional, and cyclists can only cycle on the sidewalk.

Future Work:

This model is a good starting point for modeling traffic flow, but in order to truly model the flow of traffic at highway or city, numerous more features will have to be added. Firstly, we will need to implement bidirectional lanes, U-turns, intersections, and anything else in a city's infrastructure that is related to cars. Moreover, we need to have more robust parameter values. Right now, we are just using educated guesses for values such as the probability of slowing down or switching lanes, but perhaps it is possible to get an empirical value for those parameters by collecting data. All in all, we would just need to implement more rules, and a more dynamic road to cater those rules.