Simulation of Road Traffic

Joshua Fatimilehin-Tulip and Felipe Tcach
10457454 and 10826579

April 2023

Abstract

In this project, the cellular automaton model was used to simulate road traffic. The effects of varying parameters, such as initial car density, velocity, and the rate of influx of cars on the traffic flow were observed. Behaviours of different kinds of roads were compared, including: linear and closed-loop road models; single and multiple lanes; junctions with and without traffic lights; roundabout junctions. Simulations performed for this report indicate the most efficient time interval for traffic lights to switch was approximately 6 seconds in simulation time, likely corresponding to approximately 18 seconds in real-world time, using a scale factor which was derived from [3]. Simulating traffic flow both exhibited parallels with and provided insights into theoretical and experimental findings within fluid dynamics and other fields of research.

Contents

1 Introduction

The starting point of this simulation was to randomly generate indices in an array (called 'sites') and generate random numbers to be assigned to these indices. These would represent the cars, with the numbers in the array corresponding to the velocity of the car at that index and the random indices corresponding to the starting locations of the cars on the road. The location of each car was then saved to a different list, so that any stationary cars and empty sites could then be distinguished. These 'cars' were then moved along the array by their respective velocities after each iteration. Therefore, since the velocities of the cars are in metres per second, each site represents a metre in space and each iteration represents a second in time. Each car is affected by the car in front of it, causing traffic on a large scale.

This is an example of applying cellular automaton (CA) to model the physical behaviour of road traffic, as used in [6][4][5]. The CA model was initially developed by John von Neumann, widely regarded as the 'father' of cellular automata[8]. However, one of the most famous applications of the CA model was in John Horton Conway's Game of Life [1], where it can be seen that natural oscillatory motion is formed, based on a set of small scale rules which result in macroscopic traits. This is at the core of how CA models function. Therefore, they can be extremely useful in modelling complex macro-systems, only using laws that define the interactions of the individual components [8], hence, being useful for simulations in Physics and Biology.

2 Theory

The CA model is comprised of a system of cells, or atoms, where each cell has a finite value that defines the interaction it has with neighbouring cells. This interaction depends on a set of rules which dictate which value a cell should take given the values of the cells around it, or in this model, the cells in front of it [2][10][4][6][5].

2.1 Calculating rate of flow of cars

One of the tools used to measure the intensity of traffic on any road was the rate of flow of cars. This could be expressed as the derivative:

$$\frac{dN}{dt} = \frac{dx}{dt}\frac{dN}{dx} = v_{avg} * \rho, \tag{1}$$

where N is the number of cars that flow past an arbitrary point on the road, so $\frac{dN}{dt}$ is the rate of flow of traffic. Furthermore, by manipulating the derivative, as can be seen above in eq. (1); we can see that the rate of flow of traffic is equal to the product of the average velocity of the cars, v_{avg}, and the density of the cars, ρ. However, in [6], the method used to calculate the rate of flow uses the fundamental definition, that of counting how many cars pass a certain index in a time period:

$$\frac{dN}{dt} = \frac{1}{T}\sum_{t}^{t+T} n_{x,x+1}(t), \tag{2}$$

where T is the total time over which the rate of flow is averaged, and $n_{x,x+1}(t)$ is equal to 1 if a car is flowing in the region x to $x+1$, and 0 if no car is detected.

2.2 Simulation Models

Using this cellular automaton model as the basis of approach to traffic simulation, we modelled various kinds of roads. The purpose of doing so was both to compare and contrast the behaviours of different kinds of roads and junctions, with attention mostly given to the traffic efficiency, primarily defined by flow rate, i.e., how many cars flow through any site per iteration, under the fixed parameters described.

2.2.1 Linear Road

This model was the starting point for the rest of the simulation. It consisted of cars moving along a 1-dimensional road, eventually going off the end of the road and disappearing.

The first condition for a car being able to accelerate was that the speed of the car had to be less than the maximum speed of the road:

$$v < v_{max}. \tag{3}$$

In this simulation v_{max} was set to 5 metres per second. The second condition which had to be met was that the distance to the next car in the road had to be greater than the speed of the car if it were to accelerate by one unit of velocity. This was so that if the car did accelerate, it wouldn't 'collide' with the car in front of it in the next iteration. Since the road was modelled as an array of numbers, a collision would represent two cars occupying the same site. This gave rise to eq. (4):

$$d > v + 1, \tag{4}$$

where

$$d = x_{n+1} - x_n, \tag{5}$$

v is the speed of the car, x_n is the position of the n^{th} car, and d is the distance to the next car in the road.
The condition for a car being decelerated was its velocity being greater than or equal to the distance to the next car:

$$v \geq d. \tag{6}$$

If this condition was satisfied, the car would decelerate to a velocity equal to the distance to the next car minus 1, so that it would move into the site just behind the car in front of it in the following iteration, instead of occupying the same site as the car in front of it. Otherwise, two cars would occupy the same space in a single lane, contradicting physical reality. This gave eq. (7):

$$v_t = d - 1. \tag{7}$$

Furthermore, a random deceleration feature was added to take into account the random fluctuations in traffic caused by human error and hesitation. This would have a set probability of affecting each car and would decrease the velocity of the car by 1, but only if the speed of the car was greater than zero. Finally, the algorithm to move (or translate) cars relatively simple. The new location of each car was the current location of the car, plus the current speed of the car after all the acceleration algorithms had been applied:

$$x_t = x_{t-1} + v_{t-1}, \tag{8}$$

where x_t indicates the position of a car at time t and v_t is the speed of the car at time t. Then, for a linear road, if the new location was less than the length of the road, the road at the index of the new location would be modified to have the value of the speed of the car, and this new location would be added to a new list of locations. If the car in question was the last on the road, then it only had to obey eq. (3), since there would be no car in front of it.

2.2.2 Closed-loop Road

This model was created so that it could be used to replace the traffic lights at a junction, to compare the two models and see which one was more successful in increasing the flow of traffic. It functioned by placing the cars that go off the end of the road back onto the start, hence creating a circular system, so it is essentially a type of periodic boundary condition.
Therefore, for closed-loop roads, equation 8 differs slightly as any car which has a value of x_t greater than the length of the road would be placed back at the start of the road, hence simulating a circular system.

$$x_t = x_{t-1} + v_{t-1} - L \tag{9}$$

Where L is the length of the road, and the meanings of the other symbols have already been defined above.

2.2.3 Multiple Lane Case

Depending on the distance to the car ahead of it in the same lane and the car ahead of it in the adjacent lane, a car will undergo a lane switch in order to maximise the free space (the distance to the next car) ahead of it. This is so that it will not have to decelerate due to the car in front of it, or at least by a minimum amount. However, if there is a car in the same site in the adjacent lane, then the car will not switch. Additionally, it will not switch if the distances to the next cars in both lanes are equal.

2.2.4 Bottleneck Scenario

This is when there is an obstruction in one lane of a two lane system, hence, causing any cars that reach this obstruction to stop, meaning that the cars in the obstructed lane will have to switch into the adjacent lane. Cars in the unobstructed lane will not switch into the obstructed lane in the model simulated.

2.2.5 Junctions

The initial approach for this model was to simulate a side road intersecting a main road, rather like a T-junction, and to then expand this to two roads intersecting. Once this had been simulated, the next step was to replace the junction with a roundabout which had 2 entry points and 2 exit points, and roads attached to these locations. The condition for a car to be able to go onto the roundabout was derived from reality; if the car behind the entry point on the roundabout will pass the entry point in the next iteration, then any car that is about to go onto the roundabout will need to wait, specifically, at the end of the entry road before the roundabout, until the condition stated in eq. (10) is met:

$$v_{behind} < d_{behind}, \tag{10}$$

where v_{behind} is the speed of the car behind the entry point on the roundabout and d_{behind} is the distance from the car in the roundabout to the entry point.

However, in reality, some cars still cross over, even if the car behind is going fast enough to cross the entry point. In these cases, the car behind must slow down. Thus, in the model for T-junctions, we applied this thinking, and added a 'politeness' feature. This functioned as a distance, and the higher the 'politeness' factor, the more distance the car going onto the road must leave behind it before the next car if it is to go onto the road. This can be understood more concisely in eq. (11). Here, d_{behind} is still the distance from the car behind the entry point to the entry point itself and d_{polite} is the politeness factor:

$$d_{polite} > d_{behind}. \tag{11}$$

The primary focus was to compare the efficiency of a traffic light system against a roundabout system, which was done by calculating the average time taken for all of the cars in the intersection system to leave the roads. A shorter 'emptying' time would mean that the system is better at resolving and managing traffic. The traffic lights would be switched on for a given number of iterations. A primary aim of this simulation was to determine what the optimal amount of time was for the traffic lights to be switched on.

2.2.6 Buses

In this model, vehicles called buses containing passengers were added to the road. These buses would occupy two sites on the road, representing their larger size compared to cars. These buses stopped periodically at 'bus stops', which were located at regular intervals along the road. At each stop, there were a given number of passengers waiting. Once a bus arrived at that stop, a random proportion of both passengers 'on' the bus and passengers 'waiting' at the bus stop would get off and on the bus, respectively, while the remaining passengers remain at the stop.

The passengers getting on the bus must wait for all of the passengers getting off, to get off the first, before they getting on, in accordance to bus etiquette observed in Manchester, also another manifestation of 'politeness'. This meant that each bus would spend a certain amount of time at each stop, given by eq. (12):.

$$\tau_{stop} = n_{on} + n_{off}, \tag{12}$$

where τ_{stop} is the time that each bus would spend at any stop and n is the number of passengers getting on or off. This is based on the approximation that each passenger takes 1 second to get on or off the bus.

Naturally, the 'occupancy' of each bus would increase by the number of passengers getting on the bus, but the passengers that get off the bus do not increase the occupancy of the stop, as in real life. This is why it is necessary to have an influx of passengers. Additionally, any bus that does not have to drop off or pick up any passengers, as in a real life scenario, will not decelerate before a stop and will continue on its journey.

3 Methods

3.1 Road Simulation

The method for generating any type of road would follow the same general procedure. First, an empty road had to be generated, essentially just a list of zeros. Then, from the list of indices in the road, a random index would be picked, and a random integer between zero and the maximum speed would then be chosen randomly as well. This number would then be the initial speed of the car in the first iteration, so this speed would be assigned to the road at the random index generated. This random index would then be removed from the list of indices in the road, so that another car could not be generated at that same index, and appended to another list, which would contain all of the locations of the cars on the road. This process would then repeat for however many cars should be on the road.

3.2 Random Deceleration

As mentioned earlier and described in section 2 of this report, the basis for the acceleration and deceleration algorithm is from [6]. The random deceleration was decided for each car by generating a uniform distribution of random numbers between zero and one. A random number was then chosen from this distribution, and if this number was less than the probability of deceleration, the car would undergo a random deceleration.

3.3 Lane switching

The main idea behind switching lanes was that there were two distances which each car had to compare; the distance to the next car in its own lane, and the distance to the next car in the adjacent lane. Therefore, this was done by generating a tuple for each car on the road, which contained these two distances, and adding these tuples to a list. However, if there were 2 cars in equal positions in both lanes, then neither car could switch lanes, so the tuple for the car in the case of having a 'neighbour' was $(0, 0)$. Then, iterating over this list for each car, it could be determined which cars would switch lanes and which cars would not. The condition for switching was therefore:

$$d^0 < d^1 \tag{13}$$

Here, d^i represents the distance to the next car in each lane, with $i = 0$ being the current lane, and $i = 1$ being the adjacent lane.

3.4 Bottleneck scenario

This was coded by generating one lane shorter than the other, and altering the deceleration algorithm for this lane so that when the last car is approaching the end of the road it will slow down as if there were another car at the end of the road. In this model the lane switching algorithm worked exactly the same as in the normal lane switching between 2 lanes model, except for the minor difference that the two roads were different lengths so the calculation of the distance to the next car in either lane was different.

3.5 Junctions

Most of the algorithms used in junction models had the same basis as the rest of the models. That being said, a major difference was the introduction of oscillating traffic lights, i.e., for an intersection between 2 roads, when the traffic light in one road is on, the traffic light in the other is off, and vice versa. Furthermore, the two lights would switch after a certain amount of seconds, called the 'interval' of the traffic lights.

The traffic lights functioned by only slowing down the car right before the traffic light itself. That way, all the other cars behind the first car would also slow down. The deceleration of the first car was identical to the deceleration of a car if another at 0 velocity was placed at the location of the traffic light.

3.6 Buses

There were two different models: a linear road with buses as well as cars, and then a linear road with only buses and stops. In both cases, the location of the bus was described as a tuple containing the locations of

both halves of the bus. This way, by iterating over the list of tuples containing these locations, each bus could have the relevant algorithms applied to it easily. It is important to note that the sum of the car density and twice the bus density must be less than 1. This is because each bus occupies 2 metres, so the maximum bus density is 0.5 buses per metre.

$$\rho_{car} + 2\rho_{bus} \leq 1 \tag{14}$$

The challenge in simulating this model was that the deceleration of each bus depended on whether or not any passengers were getting on or off at the next stop. Furthermore, once all of the passengers had gotten on and off, the bus then had to start accelerating again. The first step to completing this was to add stops which occurred at regular intervals, at a frequency which could be controlled, and to get the buses to stop at these locations. This was done by treating each stop as a vehicle in itself, therefore causing the buses to slow down before it.

However, the real challenge was in deriving a method to get the buses to only stop for a certain amount of time, and for this amount of time to depend on the exchange of passengers. The method used in this simulation to complete this task was to generate a list of tuples (analogous to the method described in section 3.3), where there was one tuple for each bus, and each tuple contained the number of passengers getting off at the next stop and the number of passengers getting on at the next stop.

This tuple was randomly generated based on how many people were on the bus, and how many people were on the stop. The initial number of people on each bus and at each stop was generated to be a random integer between 0 and the maximum capacity of a bus. Every time a bus passed a stop or every time a new bus was generated at the start of the road, a new 'passenger' tuple was assigned to that bus.

Finally, the method used to get the bus to stay at the stop for any a certain amount time is the following; if the passenger tuple was composed of two 0s, then the bus would not slow down due to the stop. However, if the tuple was composed of anything else, then it would have to slow down at the next stop. Then, only when the bus is at the stop, can the exchange of passengers begin. This followed a simple algorithm; if the first index in the tuple (number of passengers getting off) was greater than 0, then the value of this tuple would be decreased by 1 and so would the occupancy of the bus. This new tuple would then be returned and this process would repeated every iteration, making sure that only 1 passenger exchange occurred every second. Then, once the first index in the tuple was equal to zero, if the second index was equal not equal to 0 then this process would repeat, but now, the stop occupancy would decrease and the bus occupancy would increase instead. Then, once the tuple had decreased to (0,0), the bus would be allowed to restart.

4 Results

Due to the contrasting physical nature of each model, different approaches were necessary to present the results of our simulations. In some cases, results were best presented in graphical plots that displayed data qualitatively, as in 1 and 2; In other cases however, results were best shown qualitatively in diagrams representing the roads themselves. The parameters used in each model are described in the caption for each figure. These parameters may include the car density, maximum car velocity, road length and the total time over which each simulation ran.

4.1 Single Lane Case

We thought it best to create space-time-line visualisation plots for the road behaviour (as in [6]), with each dot showing the position of a car in space and time, in Figure 2. As each line corresponds to an aggregate of dots belonging to one car in gradually increasing space-time, these plots are known as world line plots, with one being shown in the following section. The darker regions in the plot occur where the world lines are closely packed together and indicate a high car density in this region of space. Here, the cars are close together and mostly stationary, since their world lines are mostly vertical, indicating no movement in space. Therefore, each dark region corresponds to a 'traffic jam'. As world lines clearly represent car density and motion, they are effective in visualising the way traffic evolves.

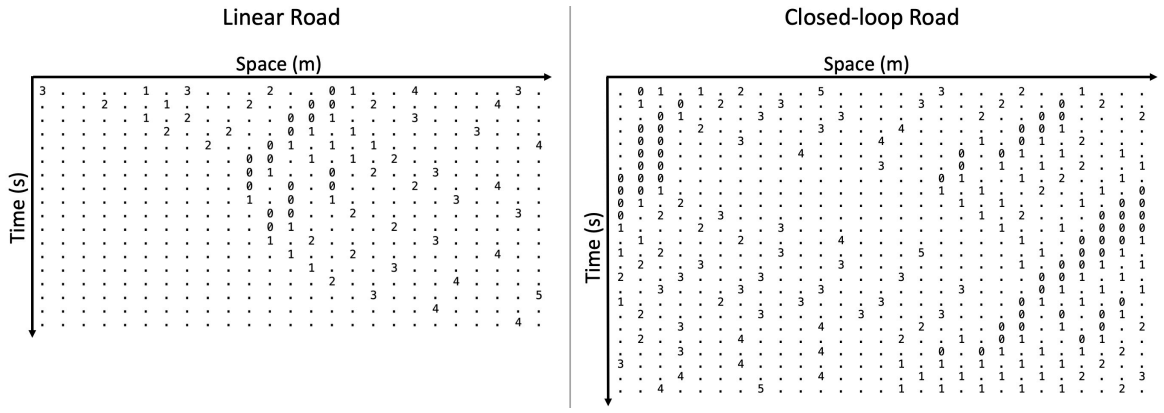

Figure 1: Diagrams of Linear road (left) and Closed-loop road (right), where both roads have a length of 28 metres, an initial density of 0.3 cars per metre and a probability of deceleration of 0.25.

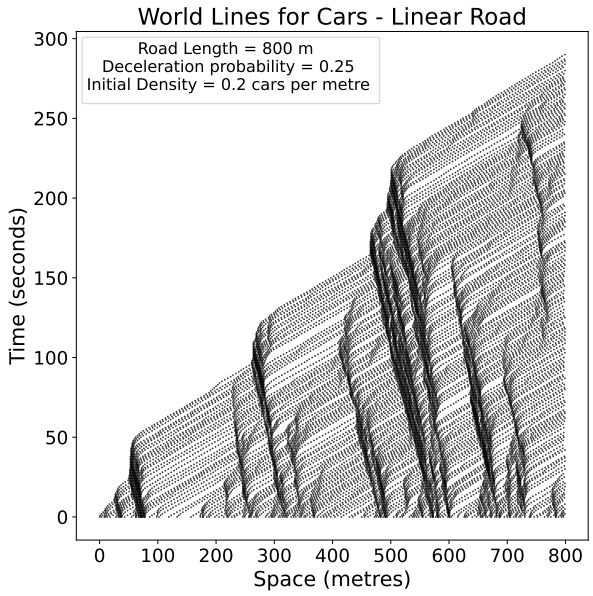

Figure 2: The diagram of the world lines of a linear road with initial car density 0.2 cars per metre, maximum velocity 5 metres per second, probability of deceleration 0.25, road length 800 metres and a total run-time of approximately 300 seconds.

In figure 3, There is a clear peak for the average rate of flow at around a density of 0.1. Before this, the rate of flow is increasing because the number of cars is increasing. The decreasing rate of flow after the peak can be explained due to the fact that after the critical density, the cars start to interact more and therefore slow down, so even though the number of cars is increasing, they are also going at a lower speed, so the rate of flow decreases overall. This trend continues until the density reaches 1 car per metre, i.e., a full road, so none of the cars can move.

4.2 Multiple Lane Case

In this case, the peak for the distribution is much wider, but at a similar height to the single lane case. This is due to the fact that in a single lane system, when there is a car in front of another, the car behind has no other choice but to decelerate and therefore decrease the rate of flow. However, in the 2 lane system, the car

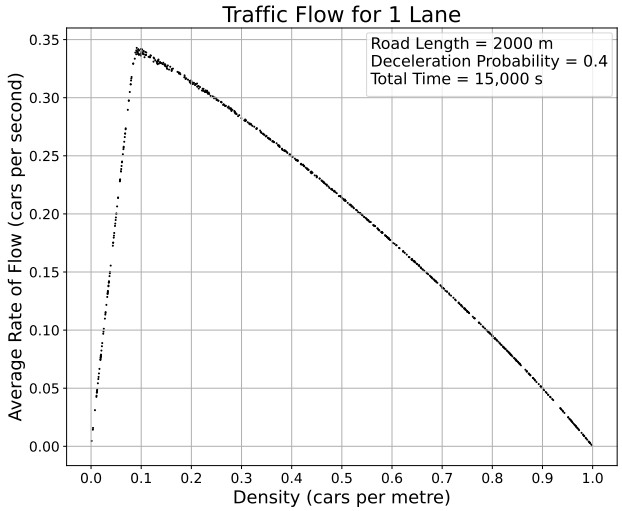

Figure 3: A graph of the relationship between the rate of flow of cars and the density of cars for a road length of 2000 metres, a total run-time of 15,000 seconds, and a probability of deceleration of 0.4.

behind has the opportunity to switch lanes if it has to slow down because of the car in front of it. Therefore, the 2 lane system is able to maintain a higher rate of flow for a larger range of densities, resulting in a wider peak in figure 4.

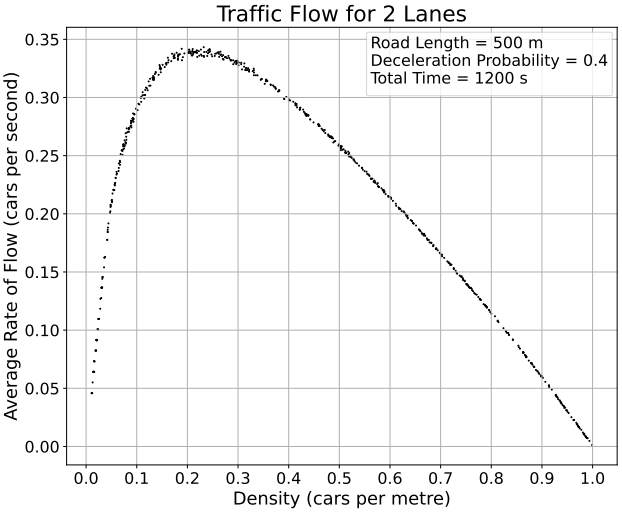

Figure 4: Lane Switching: Plot of the relationship between density of cars and average flow rate for switching between 2 lanes.

4.3 Bottleneck Scenario

Figure 6 shows how the rate of flow of the unobstructed lane remains largely constant until a critical point. This can be understood by considering the fact that, as cars leave the road, other cars switch onto the unobstructed lane. Consequentially, where the rate of flow would usually decrease due to the decreasing density of cars, it remained at approximately 0.25 cars per second. However, once all of the cars on the obstructed lane had

switched onto the other lane, there was no longer a source of cars to replace the ones that were actively leaving the road, so the rate of flow decreased drastically.

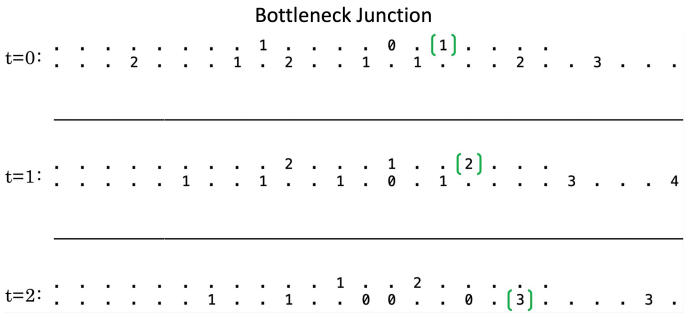

Figure 5: Diagram of Bottleneck. Cars switch to the adjacent lane when nearing the end of the first lane, but only when there is space. The car enclosed by green brackets exhibits this behaviour.

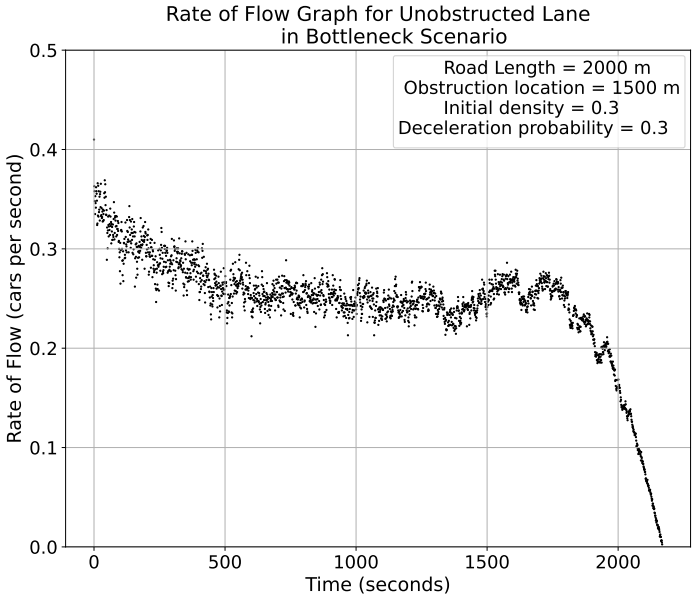

Figure 6: Graph of rate of flow of traffic varying with time in the unobstructed lane in a bottleneck scenario.

4.4 Junctions

Various simulations of traffic junctions were made, involving differing physical models. These junction models included: modelling T-junctions, junctions with and without traffic lights, and those using roundabouts. A diagram detailing the behaviour of a crossroad junction over 4 time steps is shown in Figure 5.

As mentioned in section 2.2.5, the most efficient traffic light interval can be determined from the so called 'emptying' time. In figure 8, the raw data which has been generated by running the simulation various times has been averaged out for each interval time. It can be deduced that the optimal interval for traffic lights in this intersection system is 6 seconds. This can be interpreted in many ways, however an important point to note is that the units slightly distort the picture of reality. In our simulation, the maximum speed of each car is set to 5 metres per second. In reality, for an average road in the UK, a car is likely to be travelling closer to 15 metres per second [3], which means that our model has been scaled down by a factor of approximately 3. This means that a more accurate traffic light interval could be 18 seconds, which sounds more reasonable.

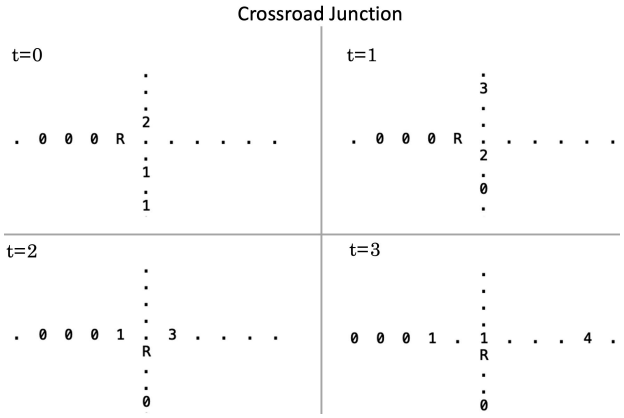

Figure 7: Diagram of a crossroad junction during 4 time steps, where the letter 'R' denotes the traffic light at the intersection being switched on.

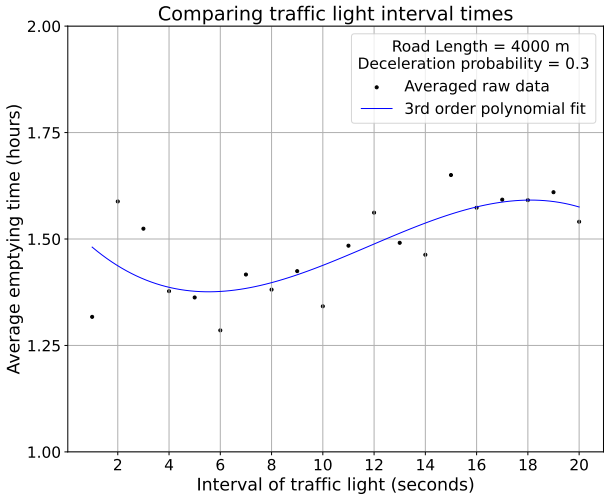

Figure 8: Graph of the emptying time for an intersection with various traffic light intervals.

Many factors affect this scaling factor, as described in Section 5 of [6], where factors such as the density where the distribution peaks in figure 3, and the value of the peak itself play a part in this approximation. Overall, this is why the interval of the traffic light could be considered to be too low. In practice, this value would increase once an appropriate scaling factor had been approximated.

In terms of the comparison between the traffic light system and the roundabout system, our model for the roundabout had an emptying time that exceeded any pragmatic expectation, as it surpassed limits placed by the computer processing power. Even much smaller systems would result in extremely large emptying times, so it was not plausible to make a comparison. Therefore, our roundabout intersection model needs several refinements, this could be one of the tasks to be completed in the future to take this model further.

4.5 Buses

The initial simulation of buses was simply the linear road case, but with buses as well as cars, with the former taking up twice the space as the latter, as shown in Equation 14.

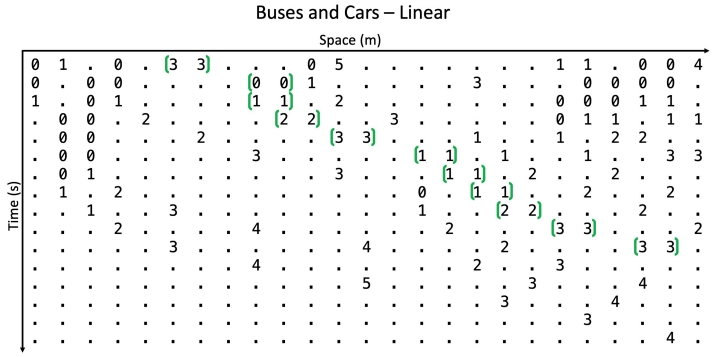

Figure 9: A diagram of buses and cars on the same road. One of the buses has been highlighted in green parentheses. Road length is 25 metres, Total time elapsed is 16 seconds, car density is 0.25, bus density is 0.15.

A diagram showing the process by which the bus stop model behaves is displayed in Figure 10. The phenomenon of 'clumping' at bus stops is evident here and was found in the majority of simulations that were run. This was due to the fact that the bus that first reaches a stop spends a certain amount of time at the stop, then the next bus that reaches that stop will spend less time there because there are less passengers to pick up since the bus that passed before already picked up a fraction of them. Therefore for every bus arriving at a stop, it will spend less time at the stop than its predecessor, therefore closing the gap to the next bus, eventually leading to the buses moving in clusters.

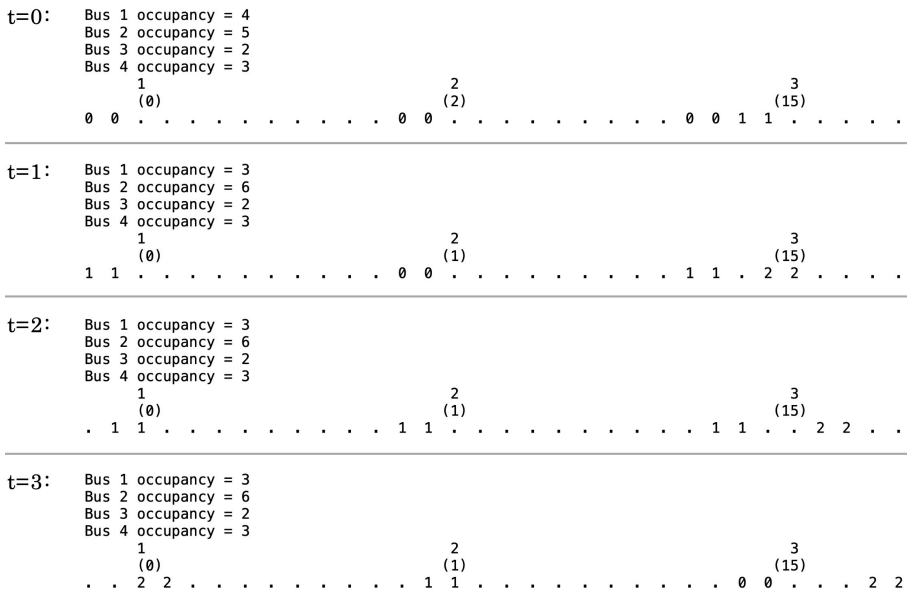

Figure 10: A diagram of the bus stop model, with each integer above the road representing the number of the stop, and the number below it in brackets representing the number of people at that stop.

5 Conclusion

It was beneficial to compare the various road types and junction models simulated for this report, as this enabled the most efficient traffic system to be determined, which may indicate physical, real-word advantages to following certain models. In the traffic light crossroad junction, the optimal traffic light interval time was found to be 6 seconds, however, as discussed previously, this value can be misleading and must be scaled appropriately to match real world observations (UK Government data [3] on car velocity suggested a scaling factor of 3). Furthermore, the cause of bus 'clumping' was determined, as described in section 4.5, as being a phenomenon that depended on the number of passengers at each stop, and therefore, the influx rate of passengers to the stops.

Similarities between traffic systems (including simulation modelling) and both experimental and theoretical fluid dynamics are often highlighted in literature, such as in [6] and [7]. The CA model is convenient for such simulations due to the simplicity of the basic rules which govern the system, and how it can accurately predict motion in irregular geometries.

Possible continuations with this project include expanding the bus stop model, to include a bus lane and a normal traffic lane. Here, only the buses would be allowed into the bus lane, but could still go into the adjacent lane to overtake another bus at a stop if there was no need for it to stop. Another obvious continuation would be to fully refine the intersection roundabout model so that a comparison could be made between the traffic light model and the newly updated roundabout model. Additionally, a further continuation would be to expand the 2 lane switching algorithm to n-lanes. This could be used to see how the graphs in figures 3 and 4 differ from the n-lane graphs. Furthermore, an investigation into bottleneck scenarios could be done to examine the oscillatory behaviour on the time dependence of the rate of flow, as done in [9].

References

[1] Andrew Adamatzky. *Game of life cellular automata*, volume 1. Springer, 2010.

[2] Francesco Berto and Jacopo Tagliabue. Cellular Automata. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2022 edition, 2022.

[3] UK Government; The Highway Code. Speed limits, 2023. Last accessed 28 April 2023.

[4] Heike Emmerich and Ernst Rank. An improved cellular automaton model for traffic flow simulation. *Physica A: Statistical Mechanics and its Applications*, 234(3-4):676–686, 1997.

[5] Sven Maerivoet and Bart De Moor. Cellular automata models of road traffic. *Physics reports*, 419(1):1–64, 2005.

[6] Kai Nagel and Michael Schreckenberg. A cellular automaton model for freeway traffic. *Journal de physique I*, 2(12):2221–2229, 1992.

[7] Daniel H Rothman. Cellular-automaton fluids; a model for flow in porous media. *Geophysics*, 53(4):509–518, 1988.

[8] Palash Sarkar. A brief history of cellular automata. *Acm computing surveys (csur)*, 32(1):80–107, 2000.

[9] Shin-ichi Tadaki, Macoto Kikuchi, Yuki Sugiyama, and Satoshi Yukawa. Coupled map traffic flow simulator based on optimal velocity functions. *Journal of the physical society of Japan*, 67(7):2270–2276, 1998.

[10] A Varas, MD Cornejo, D Mainemer, B Toledo, José Rogan, V Munoz, and JA Valdivia. Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications*, 382(2):631–642, 2007.