2D Traffic Jam Simulation

Group 1

Matej Kucera
s4551192

Andreea Zelko
s4311833

November 18, 2023

1 Introduction

Traffic jams are an inevitable part of road networks and individual transportation. It is therefore interesting to investigate the dynamics of traffic jams, effects of various conditions and possible optimization of those conditions. Many models have been proposed to replicate the behavior of traffic jams and aid in the endeavor of understanding and avoiding them. The aim of this exploratory project will be to investigate some of these models. More precisely, a 1-lane (1-dimensional) cellular automaton model proposed in literature will be replicated as a starting point. Then, the model will be expanded to accommodate 2 or more lanes (2-dimensional). Experiments will be carried out on both versions of the model to identify interesting behaviours, observe the changes cause by different parameters and find possible weaknesses of the model. The experiments will also be used to verify that the 2-dimensional extension is true to its 1-dimensional basis.

The purpose of this investigation is to uncover useful knowledge about the formation, propagation and dissolution of traffic jams on 1- and 2-dimensional roads. If the model is accurate, the simulation will be able to give insight on traffic behavior depending on initial conditions. The results can suggest possible improvements to road networks in order to improve traffic flow. To be able to interpret the results, a series of tests will need to be completed in order to verify the model's accuracy. If the model faithfully represents reality to some extent, the results can be analyzed and conclusion can be drawn about various properties of traffic like response to random slowdowns, impact of car acceleration or maximum allowed speed etc. These results could be used in the future to design roads less prone to traffic jam formation or influence car design policies. This can be achieved by altering road conditions according to the results in a way which leads in better traffic flow as predicted by the model. However there are going to be simplifications in the model and these must be carefully examined in order for the results and predictions to be applicable.

The sections of this paper are explained below.
Section 2 will provide a description of the model to be implemented, as well as the methodology used to approach this task. Research questions will also be enumerated.
Section 3 will further detail the setup of the model and the simulations run on it. It will also list the simplifications made for the sake of implementation.
Section 4 will provide descriptions and observations related to the experiments that were ran, and it will also contain visualisations of the gathered results.
Section 5 and Section 6 together will elaborate on the possible future improvements and the conclusions that were reached during the course of the realisation of this project.

2 Model and methodology

As a starting point for this project, a model will be built to imitate the one used by T. Nagatani as described in [1]. This means that a 1-dimensional cellular automaton traffic simulation will be created. As a result, cars will be simplified to cells (points on a plot), the motion of which can be calculated and visualised. For a more detailed list of the assumptions and simplifications of the model, see Subsection 3.1.

2.1 Cellular automaton model

Cellular automata are discrete computational models used for simulations. A famous example of such a model is Conway's **Game of Life**. Cellular automata have some basic concepts that characterize each model. They are: grid dimension, cell states, neighbours, rules and time steps. The model we have implemented has the following properties:

- grid dimension: 1D, the cars are represented as cells on a line. In the second iteration of the model, we use 2 lanes, so the model is 2-dimensional, consisting of multiple lanes;
- cell states: each cell has a basic state: empty or occupied. The cars have a few states that they can be in, namely, they can be either accelerating, decelerating or stopped;
- neighbours: this refers to the fact that each car is affected by the behaviour of its neighbours. A car's neighbors are defined as the closest cars behind and in front. In the case of the 2D model, this extends to neighboring lanes as well;
- rules: the cars have rules that dictate when they will accelerate and when they will decelerate. Rules for lane changing are added as well in the latter model;
- time steps: the model progresses through time steps, and the position and state of each car is evaluated at each time step;

2.2 Research Questions

Since the aim is exploration, some research questions have been identified. They are:

1. What are some observations that the model produces and do they mirror real-life phenomena?
2. What parameters of the model produce the biggest changes in the traffic jams that are created?
3. What is the best way to replicate the findings in [1]?
4. How can a 1-lane model such as the one proposed in [1] be extended to two or multiple lanes?

3 Simulation setup

3.1 The model

A computer simulation will be used to simulate traffic jam formation and phenomena associated with it. The simulation will be written in Python and run on a 2020 desktop computer. Multiple simplifications are made in order to make the simulation run faster as well as simplify it, namely:

- The road is continuous instead of being made of one car-length cells;

- The cars themselves do not have a length. They are represented as points only by their location and speed;
- The cars are updated one at a time instead of in parallel;
- All cars have identical acceleration, max speed and other characteristics;
- The road is looped around. If a car reaches the end, it is put back at the beginning, maintaining its current speed. This simulates new incoming traffic without the need to generate new cars and discard ones going out of scope;
- The cars do not have a memory. They decide their actions based entirely on the current state of the simulation.

These simplifications are made for the following reasons. The continuous road allows us to track each car, instead of each cell of the read, and it also allows for the position calculations to be more accurate, since we do not have to round to a cell.

The cars do not have a characteristic length because it is not necessary. Since we have the position of each car, and the cars to do not come closer than the given safety distance, we can say that the length of the cars is included in (or implied by) the safety distance.

The cars are not updated in parallel because this would make the computation more complex while providing no gain. The cars being updated sequentially is more accurate to how actual traffic moves since drivers can not communicate and coordinate themselves.

The given parameters are set globally for all cars. This can be seen as a way to observe the impact of the average characteristics of real-life traffic. For example, if we want to model a traffic jam made up of different cars, that have an average maximum acceleration, we can do that by setting the maximum acceleration of this simulation accordingly.

The road is infinite by virtue of having the cars moved to the beginning of the road once they reach the end. This helps reduce the load of the simulation since we do not have to generate new cars. This also allows us to observe the long-term behaviour of the cars, since they will be modeled for the whole duration of the simulation.

The cars do not have a memory of the events that happen during the simulation because that would be out of the scope of cellular automata. This is also because it greatly reduces the computational cost of the model and it allows us to be able to express the logic of the cars through mathematical equations.

The entire simulation code will be published on the students' GitHub. The code is clearly documented and available for use for verification of the results presented in this paper as well as drawing new conclusions. In order to ensure reproducibility, the random number generator is seeded with a constant. All simulations are defined by the parameters used to characterize some part of the behaviour of the cars. These constants are usually kept the same, while only changing one or two in order to observe how that parameter affects the traffic jams that are produced. The baseline of the constants are shown in Listing 1 with one exception being acceleration for which 0.05 and 0.2 are used commonly. For the purpose of reproducibility, the parameters used for every simulation that is mentioned and shown in this report have been added in the appendix of this document, found in Section 7.

```
constants = {
    "initial_density": 0.15,
    "initial_velocity": 0.3,
    "max_velocity": 1.0,
```

```
    "acceleration": 0.05,
    "safety_distance": 7,
    "max_distance": 300,
    "pattern": "random",
    "max_time": 500.0,
    "time_step": 1.0,
    "deceleration": 0.1,
    "dimensions": 1,
    "lane_change_max_speed": 0.0,
    "lane_change_method": "space-based",
    "random_slowdown_probability": 0.0
}
```

Listing 1: The default parameters of the model.

3.2 The 1D CA model implementation

Initially it was decided to implement the model proposed in [1] and explore the effects of various parameters. To achieve this, an implementation of the simulation proposed in the paper was made in Python. One small change had to made to the model. The deceleration had to be set to twice the acceleration constant. This would mimic real car performance since braking performance is usually better than acceleration. This change allows a wider range of acceleration values to be tested without the cars crashing into each other due to relatively small safety distance and maximum speed.

3.2.1 Initial parameters

- initial_density: the initial density of the road. This defines the fraction of the cells being populated by cars;
- initial_velocity: the initial velocity assigned to all cars;
- max_velocity: the maximum allowed speed. Cars are not allowed to go faster than this constant;
- acceleration and deceleration: the rate of change of speed while the car is accelerating and braking respectively;
- safety_distance: the distance between cars to be maintained. If the space between two cars is less than the safety distance, the car behind will start decelerating.
- max_distance: defines the length of the road;
- pattern: defines the initial distribution of cars;
- max_time and time_step: define the length of the simulation and the time step size;
- dimensions: the number of lanes. The simulation implements only 1 and 2- lane options. It could be generalized to any number of lanes quite easily, but that it outside the scope of this project;
- random_slowdown_probability: the probability of random slowdown at each timestep;
- lane_change_max_speed and lane_change_method: added when extending the model to 2D, will be explained in later sections;

3.2.2 Motion equations

The cellular automaton model is governed by a set of rules. These define how the cars move depending on the state of the model. The variables are named as follows: $x_n(t)$ is the position of cars n at time t, a is the acceleration constant, a_d is the deceleration constant. x_c represents the safety distance constant and $\Delta x_n(t)$ is the headway of car n at time t.

$$\Delta x_n(t) = x_{n+1}(t) - x_n(t)$$

$$\frac{d^2 x_n(t)}{dt^2} = \begin{cases} a, & \Delta x_n(t) \geq x_c \\ a_d, & \Delta x_n(t) < x_c \end{cases}$$

At each timestep, acceleration and velocity were integrated in order to find the new position and velocity of each car using the time step size constant ts:

$$v_n(t+1) = v_n(t) + \frac{d^2 x_n(t)}{dt^2} * ts$$

$$x_n(t+1) = x_n(t) + v_n(t+1) * ts$$

The velocity was clamped between 0 and v_{max}:

$$0 \leq v_n(t) \leq v_{max} \quad \forall n, t$$

The positions loop around the simulation once they reach the end, given by the road length l:

$$x_n(t) = x_n(t) \mod l$$

To verify that the created simulation was accurate, two experiments were run with the same parameters as shown in [1] to compare the resulting visualizations. The parameters are listed in Listing 1. The progress of traffic is shown in the visualizations by displaying the position of each car through time, where each car is represented by a single line in the graph. As shown in Figure 1, the simulation replicates the paper accurately. The wave-like propagation of the traffic jams is clearly seen in both simulations. Knowing this, the simulation can be used with varying inputs to explore their effect on the resulting phenomena like the propagation of the traffic jam. After extending the simulation to two lanes, experiments with similar parameters can be run to ensure that the traffic behavior does not change drastically compared to one-lane traffic.

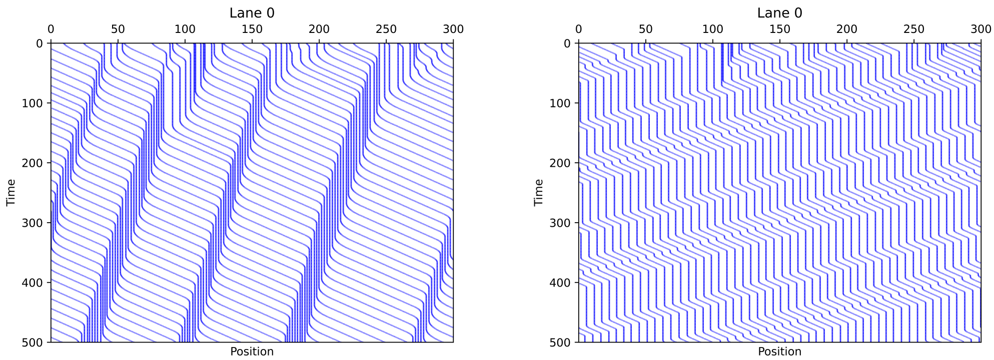

Figure 1: Simulation results with parameters identical to those in [1].

3.3 2D model extension

The next step in understanding traffic jams is attempting to simulate them for multi-lane roads. For this purpose, the model was extended to work with two lanes. Several changes were made in order to achieve this. These changes also required several parameters to be added to the constants the simulation is run on. Some alterations were also made which introduce new functionality unrelated to the two-lane extension. It would be fairly easy to extend the simulation to more than two lanes, but this is not implemented to keep the complexity down and to make the results readable.

The main component of extending the simulation was writing lane-switching strategies. The rest of the simulation could be reused almost exactly. After consideration, two possible lane-change strategies were conceived.

3.3.1 Space-based lane change

This strategy relies purely on checking whether there is enough space in the other lane. The cars are given the ability to look ahead and behind in the other lane. They are allowed to change lanes if the distance to the cars ahead and behind them in the other lane is more than the safety distance. Under this strategy, the cars switch lanes when they're about to slow down in their own lane. There is a input parameter `lane_change_max_speed` which controls this behavior. The parameter represents the speed threshold which the cars will try not to go below, i.e. if the parameter is 0.1, the car will consider switching lanes if its velocity is about to become lower than 0.1.

The lane change behavior can be expressed as an equation, where n is the current car, n + 1 is the car ahead in its own lane, m and m + 1 are cars behind and ahead in the other lane and l_n is 0 or 1 representing the current lane of car n:

$$\Delta x_n(t) = x_{n+1}(t) - x_n(t)$$

$$d_{ahead} = x_{m+1}(t) - x_n(t)$$

$$d_{behind} = x_n(t) - x_m(t)$$

$$l_n = \begin{cases} 1 - l_n, & v_n(t) + \frac{d^2 x_n(t)}{dt^2} * ts < v_{change} \wedge d_{ahead} \geq x_c \wedge d_{behind} \geq x_c \\ l_n & \text{otherwise} \end{cases}$$

where v_{change} is the `lane_change_max_speed` constant. The lane change is calculated first, before velocity and position are updated. This is shown in Algorithm 1.

3.3.2 Speed-based lane change

This strategy stems from real-life observations. In the students' experience, cars switch lanes in traffic jams if they think that the other lane is moving faster than their own lane. To implement this, cars use the ability to look for the car ahead and behind in the other lane. They evaluate the speed of the other lane as the average speed of those two cars. They then compare this to the expected speed of their own lane by taking into account whether they are able to speed up or slow down at the moment. They only switch if the speed of the other lane seems higher and if there is enough space (to avoid collisions).

This lane change procedure can also be represented in equation form using the same notation as subsubsection 3.3.1:

$$v_{otherlane} = \frac{v_m(t) + v_{m+1}(t)}{2}$$

$$l_n = \begin{cases} 1 - l_n, & v_n(t) + \frac{d^2 x_n(t)}{dt^2} * ts < v_{otherlane} \wedge d_{ahead} \geq x_c \wedge d_{behind} \geq x_c \\ l_n & \text{otherwise} \end{cases}$$

The simulation including the lane change algorithms proceeds as follows:

Algorithm 1 The simulation

```
for each time step do
    for each car do
        Attempt to change lane using the defined lane-change method
        Update current velocity
        Update current position based on new velocity
    end for
end for
```

In hindsight, these strategies are actually quite similar to each other. The initial plan was to make speed-based lane change disregard the distance to cars in the other lane, but it was quickly found that this causes crashes extremely often and therefore the distance check had to be added. This in turn made the strategies similar, as will be seen in Section 4. Due to this fact, most experiments are not run with both strategies as the results are very similar.

After implementing these lane switching strategies, some other options were added to the original model. These include:

- Random slowdown - Each car has a probability of randomly slowing down drastically in each time step. If enabled, this introduces jam-generating slowdowns to the simulation. The slowdown is twice the deceleration constant, since otherwise the cars were able to ignore the slowdowns without any major effect on flow. A possible improvement on this is to introduce memory and make a car randomly slow down for multiple time steps in a row which would be closer to reality and more impactful.
- Initial car distribution patterns - The initial distribution can be set to be random, uniform, jammed or jammed-random. The first two options are exactly what their names suggest. The jammed option starts all the cars uniformly spaced half a safety distance apart. The jammed random option starts all the cars randomly positioned within a safety distance from each other. These various starting states can be used to examine different potential situations under various conditions. Lane distribution is always random. A future improvement could be to allow the lane distribution to be defined.

The improved simulation is able to model two-lane traffic. It is up to investigation if it is realistic and whether the two lane switching strategies produce accurate results. There are also some new parameters which can be investigated in order to make the simulation more realistic or be able to mimic more phenomena.

4 Results

4.1 Comparison with the 1D model

To compare the obtained simulation results with the base simulation from [1], the simulation was applied to the same constants as shown in the paper. The results should match the phenomena observed in order to verify that the lane-switching stays true to the nature of the simulation and can be regarded as accurate. In order to visualize multiple lanes, the visualisation was divided into a separate graph per lane and the lanes are given different colors. The results are shown in Figure 2 and Figure 3. It can be observed that these do in fact match the 1D version in the general traffic jam pattern. Some smaller new patterns can be observed which are induced by the lane-switching which will be addressed later in this section.

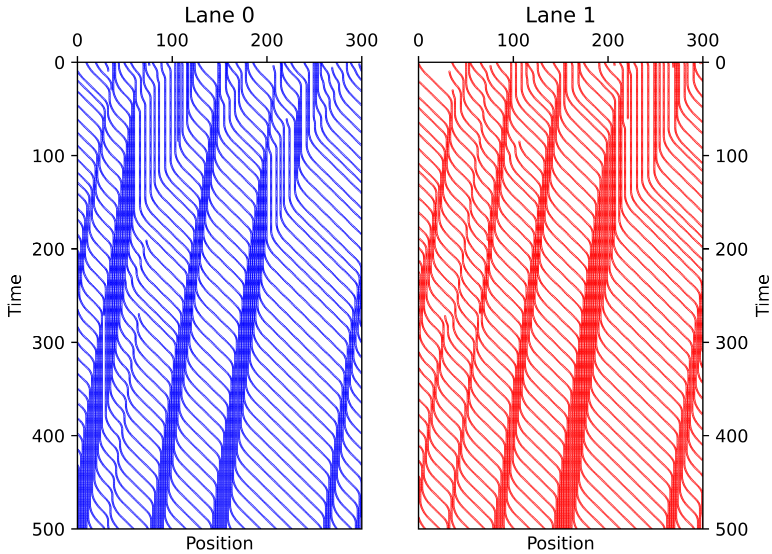

Figure 2: Space-based lane switching with parameters identical to Listing 1.

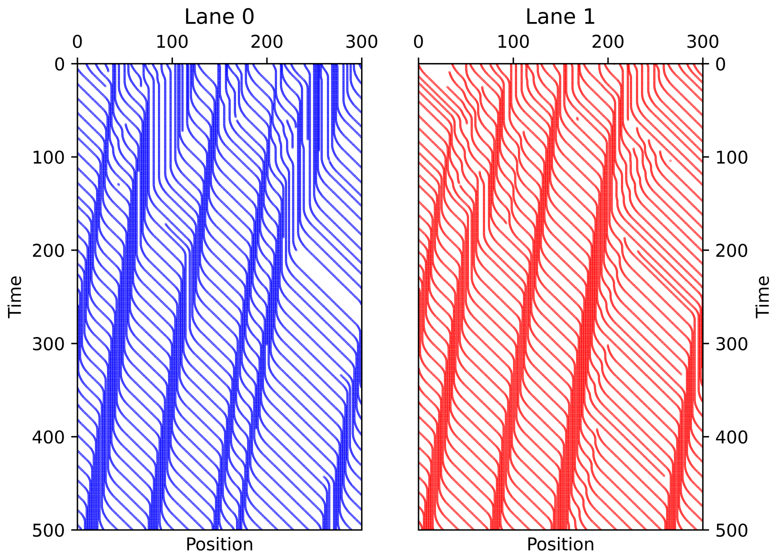

Figure 3: Speed-based lane switching with parameters identical to Listing 1

The simulation was also verified against the other set of parameters [1] provides, namely using Listing 1 and changing the acceleration to 0.2. The results as shown in Figure 4 and Figure 5 also match the 1D results to an extent. Noticeably, one lane comes to a standstill while the other exhibits behavior similar to that of a 1D simulation. This can be explained by the initial random distribution over the two lanes. It can be seen that multiple cars switch to lane 2 at the beginning. Lane 2 clearly starts with a lower density than lane 1. Some cars manage to make the switch in time while the rest get stuck in the congested lane which never moves due to the density being too high. This is realistic given that the simulation wraps around, since it is conceivable that in real life one lane can be moving but the other is stationary for some time. To alleviate this, the same experiment was run with slightly lower density (density = 0.13). The results can be seen in Figure 6. In this one, the left lane exhibits similar behavior to the original simulation while the right moves with almost no interruptions. This indicates that the density of the two lanes is slightly uneven but the distances are distributed in a way that doesn't allow cars to change lanes and balance out.

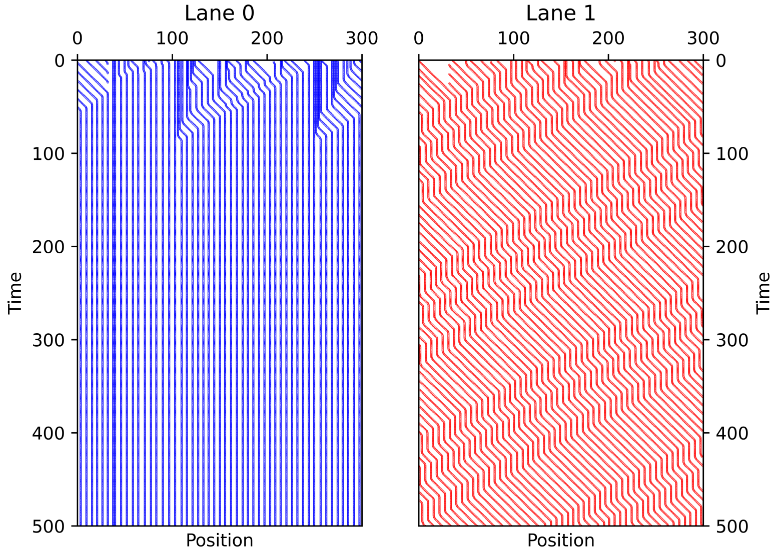

Figure 4: Space-based lane switching with parameters identical to Listing 1 except acceleration = 0.2

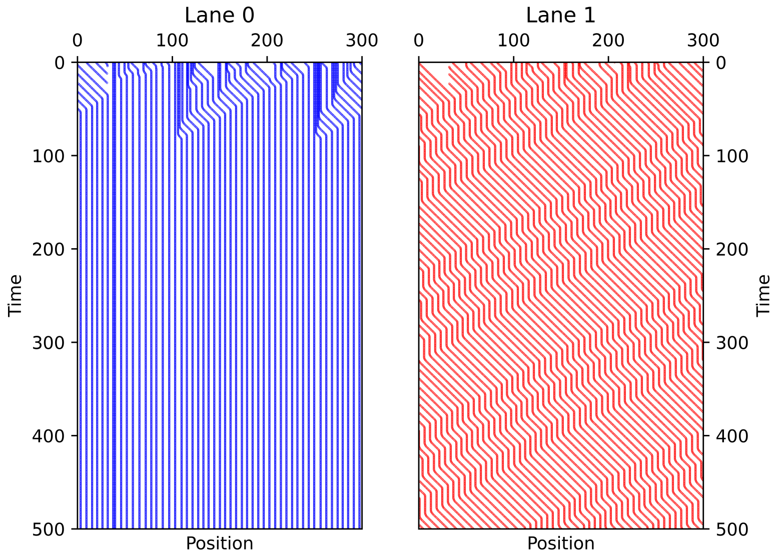

Figure 5: Speed-based lane switching with parameters identical to Listing 1 except acceleration = 0.2

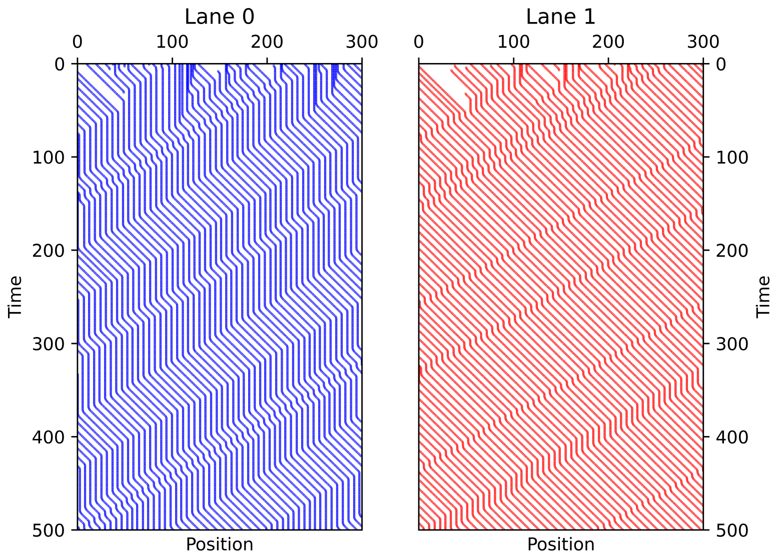

Figure 6: Space-based lane switching with parameters identical to Listing 1 except acceleration = 0.2, density = 0.13

4.2 Observations

Here the two simulations are pitched against each other in order to analyze whether the extension maintains the characteristics of the 1D simulation and mimics the phenomena accurately.

- If the density is set low enough, no traffic jams form. It is intuitively evident since if the road is empty enough, all cars can go at maximum velocity without interruption. This can be seen in Figure 7, and it happens in both the 1-dimensional and 2-dimensional simulations.

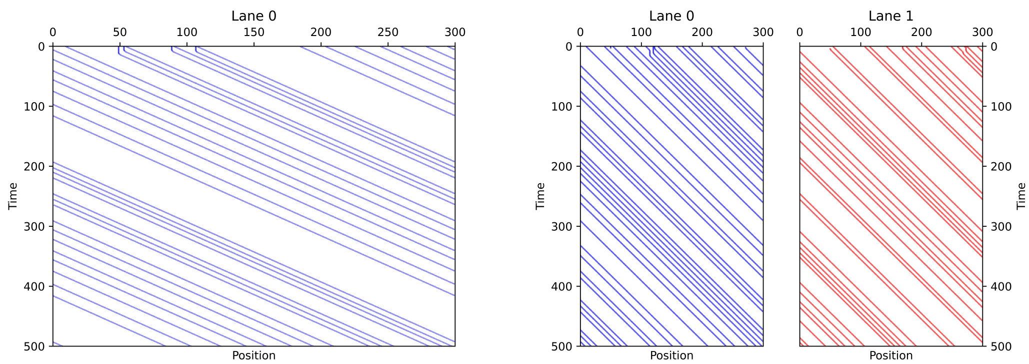

Figure 7: Simulation with low car density where all cars can go at maximum velocity. Left: 1D; Right: 2D.

- If the acceleration is very low, it affects the cars' ability to stop quickly enough. Since this model uses the acceleration to define the speed at which cars decelerate (the deceleration constant is always set to twice the acceleration unless specified otherwise), if the acceleration is too low, the cars can not decelerate fast enough and end up crashing. This behaviour is observed in Figure 8 where the acceleration is set to 0.01 and the safety distance is not big enough for cars to be able to stop if they enter it at max velocity. Due to the way the model is implemented, a crash can be identified on the plot as a car trying to decelerate, the going in front of the car because it could not stop and lastly accelerating again because the obstacle is now behind it. Again, this messy behavior is observed in both 1D and 2D simulations. This is a limitation of this model and has to be avoided by setting the input parameters with some care. Alternatively, this could be avoided by allowing cars to look further ahead and calculate how much time they need to stop, but that would complicate the calculations a lot and was therefore not implemented.

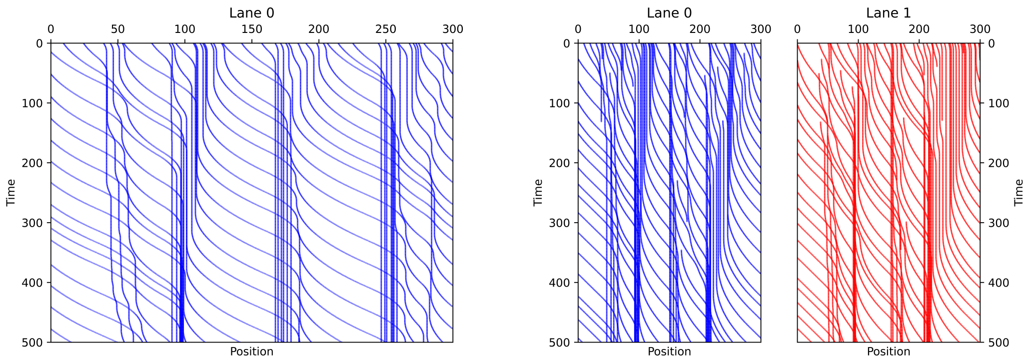

Figure 8: Simulation where the acceleration is too low, causing crashes. Left: 1D; Right:2D.

- If the maximum velocity is set quite low, the cars have the opportunity to evenly distribute themselves along the road. The result of this depends on the safety distance that is set too. If the safety distance is set to the usual value, 7, then all cars arrive at a stand-still, as can be observed in Figure 9 (top left). However, if the safety distance is lowered, the cars arrive in a perfect arrangement where they can all cruise at maximum velocity, which is visible in Figure 9 (bottom left). However, in 2D, interesting behaviour can be observed. If the safety distance is too low in 1D, in 2D, the cars will be stopped on one lane and keep moving an another lane. This is because at the beginning of the simulation, the cars are randomly distributed across the 2 lanes. In the case seen in Figure 9 (top right), there were more cars in Lane 0 and less cars in Lane 1. The difference was enough that the cars in Lane1 were still able to move.

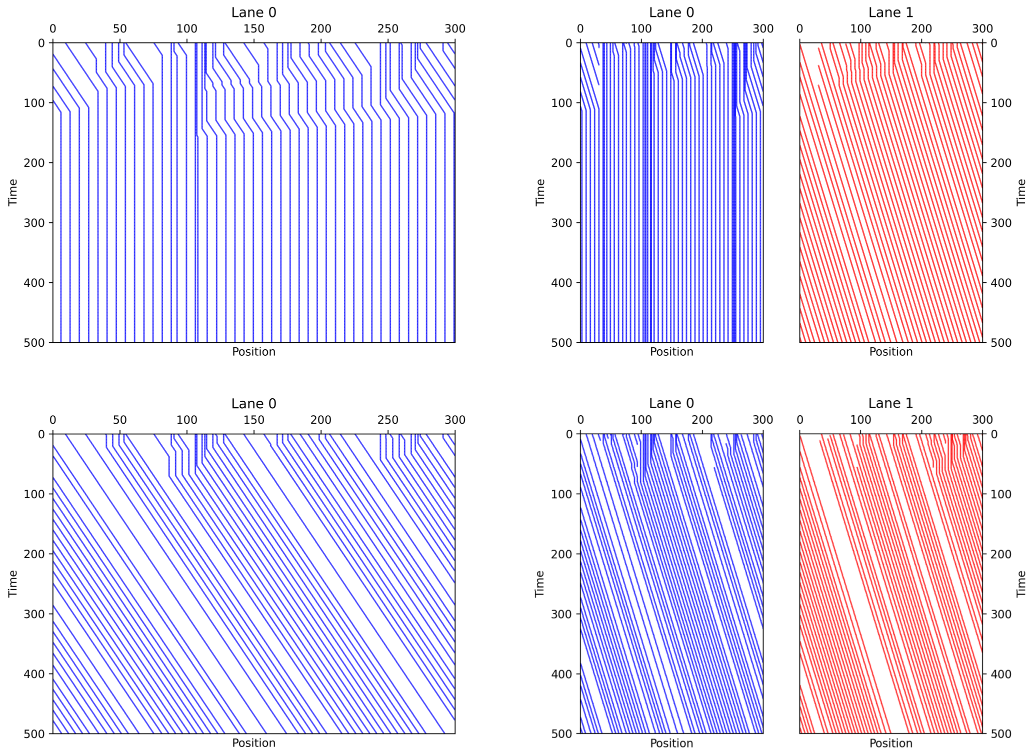

Figure 9: Comparison of changing the max velocity and safety distance in 1D and 2D.
Top left: low safety distance, 1D;
Top right: low safety distance, 2D;
Bottom left: high safety distance, 1D;
Bottom right: high safety distance, 2D.

- If the acceleration is equal to the maximum velocity, we expect to observe start-stop traffic only as the cars are able to stop and accelerate to full speed instantly. This is indeed the case as shown in Figure 10.

  An interesting phenomenon can be observed by letting the starting position be jammed immediately. To achieve this, all cars are spawned less than 1 safety distance from each other. This results is a clear wave-like propagation of the traffic jam as cars leave from the front and new cars join the back of the jam. The effect is shown in Figure 11. This is realistic to how traffic jams propagate in actual traffic. The effect could be improved by adding stochastic slowdown to the cars or otherwise adding realism, however this is an incredibly simple model capable of modelling a realistic traffic jams scenario on a one-lane road.

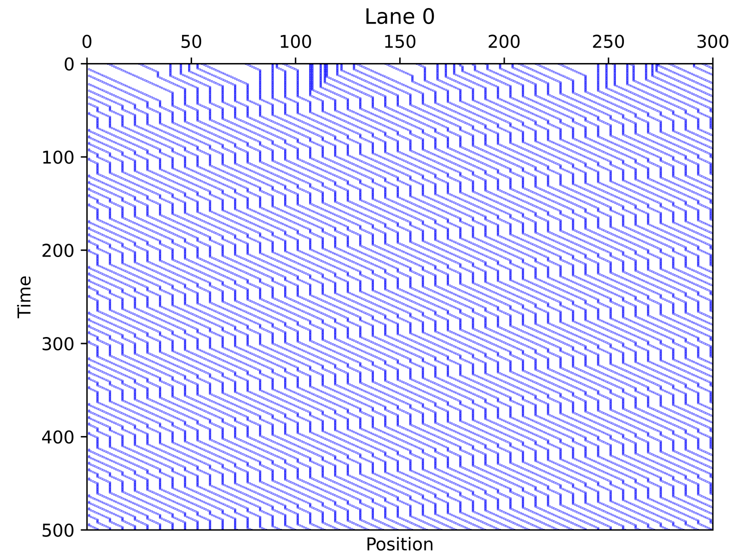

Figure 10: Acceleration equal to maximum velocity

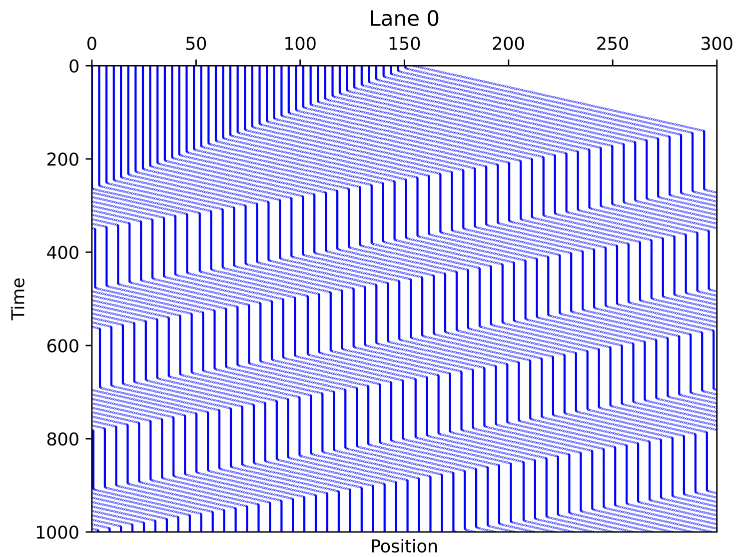

Figure 11: Starting in a jammed state

4.3 Parameter analysis

In order to draw some conclusions from the simulation, some metrics need to be defined in order to measure the impact of changing parameters on the traffic behavior. While these metrics can not be reversed into the parameters of the simulation, they can be used to quickly see the impact that a parameter change has on the model and can be used to draw up conclusions about the parameters themselves. The metrics are:

- Full speed fraction - This metric measures the fraction of cars which were moving at full speed during the simulation. It is measured as the average of the fraction of cars travelling at full speed at each timestep.

$$f_{fs} = \frac{\sum_0^{max\_timestep} \sum_0^{n\_cars} max\_vel\_cars}{max\_timestep * n\_cars}$$

$$max\_vel\_cars = \begin{cases} 1 & \text{if } v_n(t) = v_{max} \\ 0 & \text{if } else \end{cases}$$

- Stopped fraction - This metric measures the opposite of the full speed fraction and is calculated in thee exact same way, taking stopped cars into account only.

- Distance travelled fraction - This is the fraction of the actual distance covered by the cars divided by the maximum possible distance the cars could have travelled during the simulation. This metric is directly correlated to traffic flow. ts is the time step constant like in previous equations.

$$f_{dt} = \frac{\sum_0^{max\_timestep} \sum_0^{n\_cars} v_n(t) * ts}{max\_timestep * n\_cars * v_{max}}$$

One other metric which could be useful in this context is the number of lane changes and some kind of lane change frequency measurement per car. To further investigate the lane-change algorithms and the accuracy of the model, it could be interesting to investigate how often cars change lanes. It could also be investigated whether the same car makes frequent lane changes and if so, why. These metrics could offer further understanding of the model and its applicability.

To simulate the impact of one variable changing, these three metrics were calculated for a range of variable values and plotted. The parameters of these simulations are mostly kept same as Listing 1 except for some exceptions where the circumstances required changes in order to see underlying patterns better. The parameter values for these can be found in Section 7.

1. Density
   The plot can be seen in Figure 12. As expected, the density has clear influence on the metrics and an ideal window can be identified. This is around 0.15, which corresponds to the density used for most experiments. This also makes sense with relation to the safety distance and road length values since if there simply isn't enough space on the road it will become congested inevitably.

2. Acceleration
   It can be seen from Figure 13 that an increase in acceleration tends to favor traffic flow. This is to be expected as higher acceleration allows better driver reactions and smoother flow.

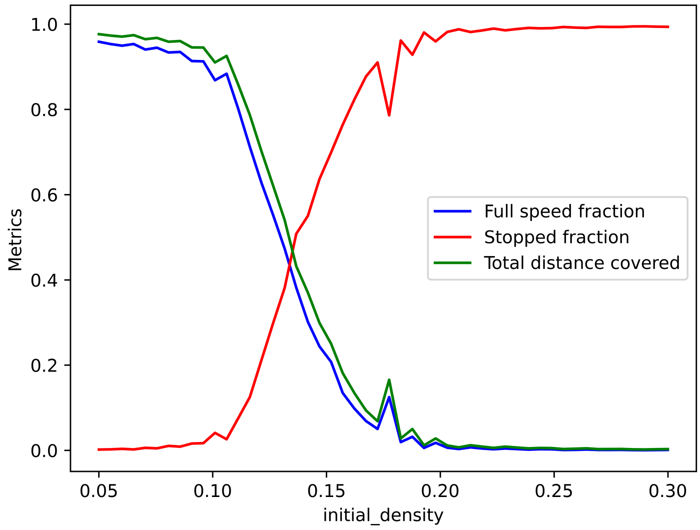

Figure 12: The impact of varying density

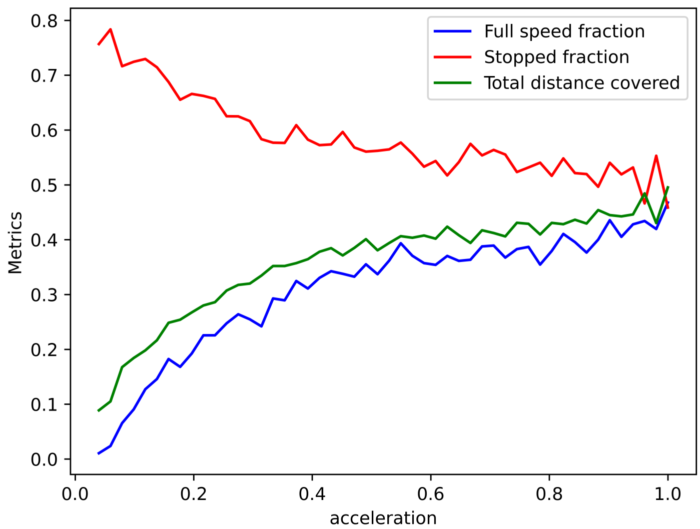

Figure 13: The impact of varying acceleration

3. Space based lane change max speed
   The plot in Figure 14 shows that this parameter actually has no impact on the flow. Except for small variations, it seems that the space-based lane change strategy does not actually get

influenced by the speed at which lane changes occur. This can be explained by the fact that if a car is slowing down, it will eventualyl slow down to a speed at which it will consider changing lanes. Hence, the parameter only makes a small change in the timing of lane changes and does not influence traffic flow.

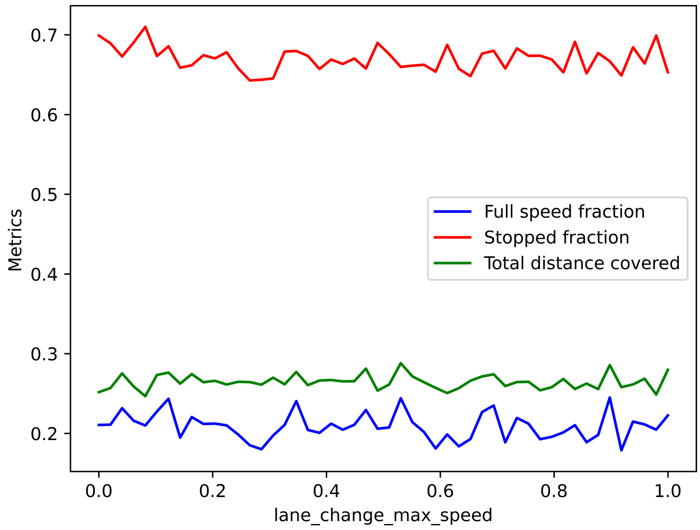

Figure 14: The impact of varying maximum lane change speed

4. Random slowdown probability
   As seen in Figure 15, the impact is as could be expected. Higher probabilities cause traffic flow decrease and even lead to the full speed fraction nearing 0 as the cars struggle to accelerate to full speed before they are randomly slowed down again, especially if the same is happening around them. The total distance shows that the lanes are both still moving.

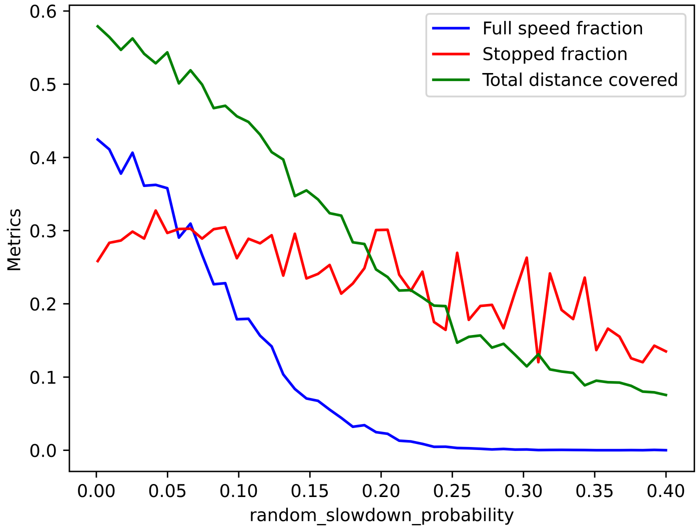

Figure 15: The impact of varying random slowdown probability

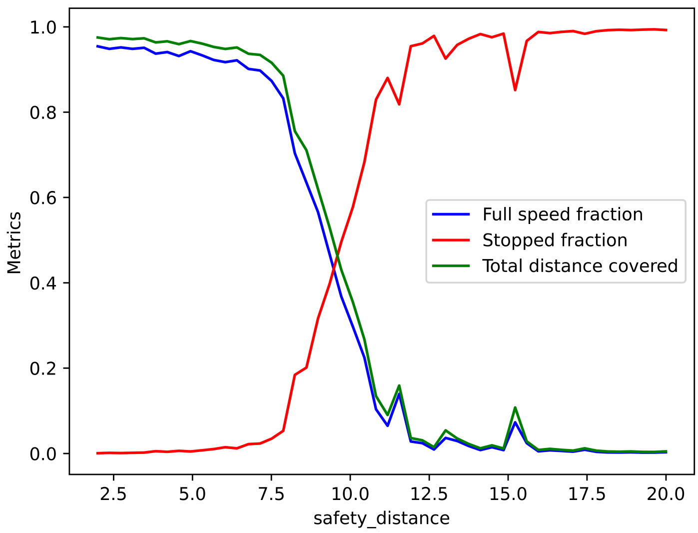

Figure 16: The impact of varying safety distance

5. Safety distance
   This parameter has a similar but inverse effect to density, as the two are related. The effect is shown in Figure 16. This makes sense as lower safety distances allow higher density of cars

without complete congestion.

Overall, it can be seen that there are interesting conclusion to parameter analysis. Conclusions can be made about the space based lane change strategy as well as the impact of acceleration on traffic flow. If the simulation was extended and ran for more cars, more lanes and other increased parameters, it could provide valuable insight into traffic dynamics.

5 Discussion

The model used in this project is quite powerful, especially considering the complexity involved. Even though no parallelization was used, the simulations were able to mimic the results obtained by [1]. Additionally, the model provided enough flexibility to test the behaviour of traffic under different conditions and setups. This testing led to interesting observations. There are several simplifications in the model to allow ease of implementation and efficient running which are discussed in Subsection 3.1. These could be made more realistic in future simulations with more time and computational power.

5.1 Future research and improvements

Below are some ideas of how the model could be improved and expanded upon:

- Parallelizing the movement of the cars;
- Investigating systems of 3 or more lanes;
- Creating new cars instead of wrapping the road around;
- Adding new strategies for line changing;
- Adding new strategies for the movement of the cars (braking, accelerating, etc.).
- Adding driver behavior - defensive drivers, polite drivers, bad drivers etc.

This list is not comprehensive, it is simply a starting point for further investigations into this problem containing a few suggestions.

6 Conclusion

In the beginning phase of this project, a basic model was implemented with the goal of reproducing the results obtained by Nagatani in [1]. Once this model was ready, some experiments were carried out to try to understand the importance of the parameters, and some observations were noted. Later, the model was expanded. The biggest change was the addition of support for 2 lanes (2-dimensional model). Some smaller changes include the addition of the deceleration parameter, as well as overall improvements to the code quality. The 2D model was used to further check the behaviour of the previous observations and to see how they expanded to more dimensions. Parameter studies were also carried out to analyze the influence of some important parameters on traffic flow.

Individual contributions

The work split in this project was equal among the two group members. This was ensured by working on the project concurrently. The code and the report were written by both group members together. Therefore the work split on all parts of the project was 50/50.

References

[1] T. Nagatani, "Effect of car acceleration on traffic flow in 1D stochastic CA model," *Physica A: Statistical Mechanics and its Applications*, vol. 223, no. 1, pp. 137–148, 1996.

7 Appendix

This section contains all the parameter setups used to create the figures shown this report.

Figure 1
Left: "initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.05, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0
Right: "initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0

Figure 2
"initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.05, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.0, "lane change method": "space-based", "random slowdown probability": 0.0, "deceleration": 0.1

Figure 3
"initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.05, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.0, "lane change method": "speed-based", "random slowdown probability": 0.0, "deceleration": 0.1

Figure 4
"initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.0, "lane change method": "space-based", "random slowdown probability": 0.0, "deceleration": 0.4

Figure 5
"initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.0, "lane change method": "speed-based", "random slowdown probability": 0.0, "deceleration": 0.4

Figure 6
"initial density": 0.13, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.0, "lane change method": "space-based", "random slowdown probability": 0.0, "deceleration": 0.4

Figure 7
"initial density": 0.05, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "deceleration": 0.4, "dimensions": 1, "lane change max speed": 0.0, "lane change method": "space-based", "random slowdown probability": 0.0
"initial density": 0.05, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "deceleration": 0.4, "dimensions": 2, "lane change max speed": 0.0, "lane change method": "space-

based”, ”random slowdown probability”: 0.0

Figure 8

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 1.0, ”acceleration”: 0.01, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 0.02, ”dimensions”: 1, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 1.0, ”acceleration”: 0.01, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 0.02, ”dimensions”: 2, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

Figure 9

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 0.3, ”acceleration”: 0.2, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 0.4, ”dimensions”: 1, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 0.3, ”acceleration”: 0.2, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 0.4, ”dimensions”: 2, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 0.3, ”acceleration”: 0.2, ”safety distance”: 5, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 0.4, ”dimensions”: 1, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 0.3, ”acceleration”: 0.2, ”safety distance”: 5, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 0.4, ”dimensions”: 2, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

Figure 10

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 1.0, ”acceleration”: 1.0, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”deceleration”: 2.0, ”dimensions”: 1, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

Figure 11

”initial density”: 0.15, ”initial velocity”: 0.3, ”max velocity”: 1.0, ”acceleration”: 0.2, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”jammed”, ”max time”: 1000.0, ”time step”: 1.0, ”deceleration”: 0.4, ”dimensions”: 1, ”lane change max speed”: 0.0, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.0

Figure 12

”initial density”: 0.3, ”initial velocity”: 0.3, ”max velocity”: 1.0, ”acceleration”: 0.2, ”safety distance”: 7, ”max distance”: 300, ”pattern”: ”random”, ”max time”: 500.0, ”time step”: 1.0, ”dimensions”: 2, ”lane change max speed”: 0.1, ”lane change method”: ”space-based”, ”random slowdown probability”: 0.01, ”deceleration”: 0.4

Figure 13
"initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 1.0, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.1, "lane change method": "space-based", "random slowdown probability": 0.01, "deceleration": 0.4

Figure 14
"initial density": 0.15, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 1.0, "lane change method": "space-based", "random slowdown probability": 0.01, "deceleration": 0.4

Figure 15
"initial density": 0.1, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.05, "safety distance": 7, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.1, "lane change method": "space-based", "random slowdown probability": 0.4, "deceleration": 0.1

Figure 16
"initial density": 0.1, "initial velocity": 0.3, "max velocity": 1.0, "acceleration": 0.2, "safety distance": 20.0, "max distance": 300, "pattern": "random", "max time": 500.0, "time step": 1.0, "dimensions": 2, "lane change max speed": 0.1, "lane change method": "space-based", "random slowdown probability": 0.01, "deceleration": 0.4