Touring Machines;
On the effects of on ramp traffic light timer duration on throughput.
GitHub page
Trello board
Online documentation

Stan Meyberg,[1] Thijs van den Berg,[1] Casper Smet[1]

[1]Institute for ICT, HU University of Applied Sciences, Utrecht

2019
December

Abstract

With world poverty rates declining and access to private transportation methods increasing, road congestion is an ever-growing problem. Using the conducted experiment, we attempted to answer the following question: How does a traffic light on an on ramp and its timing affect throughput on a highway? In order to do this, we implemented and altered the Nagel-Schreckenberg congestion model.

Increasing the timer duration of the traffic light showed a positive effect on the throughput of the main road. Increasing the timer duration also caused the buildup of a large queue on the on ramp. Further research may focus on not only congestion on the highway, but also that of the on ramp. Finding the right balance between the two will hopefully improve throughput for the road network as a whole.

Contents

1 Introduction

Road congestion is getting worse and worse with the ever expanding population size and growing wealth of this modern world. With the growing wealth a lot more people can afford the luxury of owning a car. this results in an increase in road use, and, most likely, a decrease in overall throughput on the road. The consequences of the diminishing throughput include an increase in greenhouse gas emission and an increase in travel time. [1]

In our experiment we want to model a 1-lane road based on the Nagel-Schreckenberg model. This model will eventually be expanded with an on ramp and a traffic light. The experiment will simulate the throughput with and without the addition of a traffic light. The results will provide more insight about the differences in throughput between the different scenarios and hopefully can help with improving throughput in real life scenarios.

2 Research Question

Using the simulation, we will try to answer the following research question:

> How does a traffic light on an on ramp and its timing affect throughput on a highway?

The following two hypotheses will help with answering the research question:

> H0: A traffic light does not have a positive effect on throughput
> H1: A traffic light does have a positive effect on throughput

For this experiment, throughput is defined as the average velocity of all the agents on the main road. A positive effect being a heightened average velocity, and a negative effect being a lowered average velocity.

3 Method of Approach

3.1 Data collection

While the simulation is running, it will generate a lot of data. Eventually we want to evaluate our experiment with this generated data. To be able to that we need to collect and store some data from the simulation. We're most interested in the following data:

- The velocity of every agent per time step
- The amount of agents (cars) on the road per time step
- The positions of every agent per time step

[1] http://theconversation.com/remove-car-lanes-restrict-vehicles-and-improve-transit-to-reduce-traffic-congestion-127873

- The average velocity of all agents

With this information we ought to be able to formulate an answer to our research question.

3.2 Environment setup

In the simulation, agents will travel and can only travel by road. The road is represented by a grid of cells. This grid of cells defines the Nagel-Schreckenberg model as a cellular automaton. Each cell contains zero or one agent. The grid of cells is one by n long. n is defined as the length of the road.

The road initially consists of one lane, thus, as just mentioned one wide. According to the Nagel-Schreckenberg model, there is no end of the road. The visual end of the road will be connected to the beginning of the road. This will form a continuous highway with no ending.

Time in the environment will develop in discrete time steps. Every time step, each agent will perceive the environment and possibly change their position on the grid. The behaviour of the cars will be further explained at the *Car behaviour*-section.

If time permits, an on ramp will be added to the road. This ramp shall contain a traffic light. The traffic light will regulate the throughput of cars joining the main road. The behaviour of the traffic light will be further explained at the *Basic traffic light behaviour*-section.

3.3 Car behaviour

The basic behaviour of a car in the Nagel-Schreckenberg model is defined as such:

- A car perceives its current velocity. When a car is not at its maximum velocity, it accelerates. Each time step, velocity increases by one unit.
- A car perceives the distance between it and the nearest car in front of it. It is always able to perceive the nearest car. If the distance is smaller than its velocity, it brakes. The car's new velocity then equals the distance.
- Each time step, a car may randomly reduce its velocity by one with a probability of p[2]. Velocity may not be reduced to 0.
- A car then moves forward in cells equal to its velocity.

These four steps are conducted in order from first to last. If time permits, the following behaviours and alterations to behaviour will be added to this model.

[2]In the original model, $p = 0.5$.

3.3.1 Lane merging behaviour

While a car tries to merge on the main road the following set of rules apply:

- The agent on the on ramp can only switch to the lane on his left hand side
- The agent on the on ramp can't switch lanes if there is another car on the lane the agent wants to switch to
- When there is no space on the road to merge into the agent has to wait until such space presents itself.

An optional addition will be to let the agents that are already on the road perceive that some cars want to merge, and that these agents will attempt to give them room.

3.4 Traffic light behaviour

The traffic light placed on the on ramp will either be on or off.

If the traffic light is off, the agent on the on ramp will directly try to join the highway. The agent on the on ramp will only succeed in joining the highway *if* any of the cells connected to the on ramp are empty. If the agent on the on ramp is not able to join the highway, the agent will have to wait until such space presents itself.

While the traffic light is on, the agents on the on ramp will stop until the light switches off again. The traffic light will have a timer with an interval to regulate the throughput to the main road. The lane merging of agents is described in the *Lane merging behaviour*-section.

The timings of the traffic light will be easy to change. During the batch run, multiple traffic light timings will be tried and analysed.

3.5 Batch run

In order to properly analyse a simulation with a variety of settings, the simulation must be run a multitude of times.

Ideally, you do not have to reset and rerun a simulation for each new set of settings.

This requires a 'Batch run' functionality. For this simulation, the setting in need of testing is the timing of the stoplight.

3.6 GUI

The GUI will not only serve as a pleasure to the eyes but will also serve as a handy tool to easily change the parameters of the simulation. The GUI will also serve as a visual way to verify that the model and underlying scripts are operating as intended.

3.7 Deliverables

Eventually we will have to model the environment and the behaviour of the agents that exists in that environment. In our method of approach we described what each specific module contains. In our deliverables we will state the specific functions/classes we eventually want to deliver/build.

3.7.1 Data collection

- A function to collect the chosen data and store it

3.7.2 Environment setup

- A single road represented by a grid of cells

3.7.3 Car behaviour

- Car class with basic behaviour according to the Nagel-Schreckenberg model
- Car merging behaviour

3.7.4 Traffic light behaviour

- A function that regulates the amount of agents that join the highway
- A function that regulates the throughput of the light.
- A function that controls the timing of the traffic light

3.7.5 Batch run

- Function to run the simulation with different settings

3.7.6 GUI

- A visual representation of the road with the agents on it
- Sliders to control the amount of cars on the road
- A plot to show the average velocity per step

4 Tool selection

For this particular model, defining agent behaviour is not a point of contention. The two most important aspects of this simulation are:

1. Data collection
2. Environment

The behaviour an agent exhibits is fairly simple, and can be implemented easily. The data collection functionality is necessary to be able to come to any conclusions based on the simulation's results. Without a proper grid-based space or discrete time functionality, this model would be nearly impossible to implement.

These will be the deciding factor in choosing an ABMS tool.

At the start of this assignment we were given the choice between three different tools to use to construct our simulation. These tools were:

1. Mesa
2. Unity
3. NetLogo

For each module, except for the behaviour modules, the tool will be given a rating out of five based on their suitability. Each tool will also be given a rating out of five based on their feasibility.

4.1 Mesa

4.1.1 Suitability

Data collection **** The data collection requirements are easy to implement in Mesa. Mesa offers the 'Data Collection' module. This module allows for easy model- and agent-level variable saving. The only required parameter being a dictionary containing the names of the variable one wishes to save.

It stores these in Pandas data frames. Pandas being a famously fast module for transforming and analysing large amounts of data.

Environment ****

Environment; Space **** The environment would be easy to implement in Mesa. Mesa offers the 'Space' module. This contains classes which perfectly fit the spacial requirements. See *space.Grid* and *space.SingleGrid*.

Environment; Time **** Mesa also offers the 'Time' module, which again, offers classes which perfectly fit the chronal requirements. See *time.SimultaneousActivation* and *time.StagedActivation*.

Batch run ***** In order to run a simulation many times with differing settings, Mesa offers the 'batchrunner'. Batchrunners allow batch runs. 'Batchrunner' also works well with the 'Data Collection' module. This can even be done using multiprocessing, making the process significantly faster.

GUI *** In order to dynamically alter settings and render simulations, Mesa also offers the 'Visualisation' module. Using this module, one can generate simple web-based visualisations.

This web interface also allows for sliders for any set of settings. It also updates in real time based on the script it is executing on, eliminating the tedium of restarting your interface every time a script is altered.

Perhaps the greatest pro for this implementation of a GUI is how easy it is. A GUI with three sliders, a grid and a graph with live animations took slightly less than 70 lines of code.

4.1.2 Feasibility *****

Due to the prior experience of this team, Mesa should be an easy to use tool. Mesa is used in Python, and all three members of this team have considerable experience in this programming language.

4.2 Unity

4.2.1 Suitability

Data collection *** There is not a standard or build-in solution to data collection in Unity. C# is also not a common language for data collection or data farming. It is possible, however, to read the values of specific agents every frame of the simulation. These variables can then be saved to e.g. a .CSV file.

Environment ***

Environment; Space *** Most projects in Unity are 3D. For the simulation however, we would like to use a 2D grid. Unity offers a way to make 2D projects. Most 2D Unity projects are side-scrollers. For a cellular automaton model it would be more common to use a top-down view. To make the Nagel-Schreckenberg model as a side-scroller would however be very original. With some workarounds it will be possible to implement a grid in Unity.

Environment; Time **** The functions Perceive, Update and Action will be executed every frame of the simulation. This makes time in Unity discrete. These functions shall be executed parallel for multiple agents in one frame.

Batch run ** Just like Data Collection, there is no default solution for batch running the simulation in Unity. Unity offers the ability to change the values of the agents and the environment while the simulation is running. For the batch run to be useful, data during the different configurations needs to be collected and saved. As discussed at Data Collection, this will be very difficult.

GUI **** The GUI is one of Unity' strengths. Unity offers incredible graphics with little to no effort. It is possible to make sliders to control values as the amount of cars on the road or the maximum velocity of the agents. The underlying code however will need to be written manually and could be rather difficult.

4.2.2 Feasibility ***

Unity is a very powerful game engine which offers a good basis for graphical stunning simulations. A lot of things specific to making simulations however are missing and need to be written manually. The learning curve of Unity is very steep and all the team members have little experience working with Unity and coding in C#.

4.3 NetLogo

4.3.1 Suitability

Data collection *** There isn't really a build-in method for gathering info about a process or the agents. However, the user can make use of global variables which by updating them every tick can eventually be used to plot the data. Furthermore the user can make use of the $file-close$ and $file-open$ commands to write away the values the variables or agents hold.

Environment ****

Environment; Space ***** The needed environment would be relatively easy to implement in NetLogo, because NetLogo already works with a grid of cells called patches.

Environment; Time *** The functions that define an agent will be executed every tick. However these agents will not operate parallel to each other. NetLogo is at its core a synchronous system.

Batch run **** In order to perform experiments with models and to run a model many times with different settings NetLogo has BehaviorSpace. BehaviorSpace is a tool integrated with NetLogo that allows you to run a model many times, systematically varying the settings of the model and recording the results of each model. It lets you explore all the possible behaviours of a model.

GUI **** The GUI of NetLogo isn't that extensive, but it all works rather intuitive. NetLogo takes a lot of hassle of displaying the simulation away. The user can literally type $create - turtles20$ and NetLogo will display 20 newly created agents on the screen.

Furthermore with NetLogo the user can create all kinds of inputs, sliders, buttons, etc on the screen to let the end-user choose some parameters for itself.

The style of the GUI of NetLogo may be a little old-fashioned, nonetheless it does its job well and correct.

4.3.2 Feasibility ****

NetLogo was designed for both research and education purposes. Therefore it has a nice environment for building multi-agent simulations and is user friendly. The learning curve is much lower than say for Unity. That said there is no prior experience in the team with NetLogo and its language. Therefore the development of the simulation can perhaps take more time.

4.4 Chosen tool

	Mesa	**Unity**	**NetLogo**
Data collection	****	***	***
Environment	****	***	****
Batch run	*****	**	****
GUI	***	****	****
Feasibility	*****	***	****

As can be seen at the ratings per tool above, Mesa was the highest scoring tool of the three available tools. Therefore we have chosen Mesa to use for building the simulation.

5 Design of the experiment

The simulation used for the experiment consists of the functions mentioned in the deliverables. For the experiment, we have chosen five different settings for the timing of the traffic light and three different settings for the number of initial cars. Each combination of traffic light timing and amount of cars will be simulated. There will be a total of $6 \cdot 3 = 18$ different scenarios tested. Each unique combination is then simulated 20 times. This will further enhance the reliability of the experiment. Each simulation runs for 100 time steps. In total, $6 \cdot 3 \cdot 20 = 360$ different simulations will be executed.

The following settings will be tested in the experiment:

Timing of the traffic light:

- 0 time steps
- 1 time steps
- 2 time steps
- 4 time steps

- 6 time steps
- 101 time steps[3]

The timing of the traffic light involves the interval of steps on which the agents are permitted to the on ramp. The number of agents that are permitted to the on ramp are one agent per interval.

Number of initial cars on the main road:

- 10 cars
- 25 cars
- 35 cars

6 Results of the experiment

For all three variations of initial road density, the greatest average velocity occurred at a timer duration of six.

This implies that there is positive correlation between timer duration and average (average) velocity. The amount of correlation appears to decrease at greater levels of initial road density.

Timer	N	Average Velocity	Average Standard Deviation
6	10	3.76	1.36
6	25	1.83	1.23
6	35	1.38	0.87

Table 1: The greatest Average Average Velocity for each N

This is not an entirely surprising result. At greater timer durations, less cars get the opportunity to merge. This, in turn, lowers the increase in road density over time. Road density negatively correlates with average velocity.

Do note that what is measured here is average velocity on the highway over time, not the average velocity of all agents. It does not include the agents waiting in a queue, that would most likely lower the average velocity significantly.

The t-test compares two means and tells the user if they are different from each other. Furthermore the t-test also tells how significant the differences are.

As can be seen in the created table the t-values between the base and intervals at a low N are significantly higher than the t-values at higher N's. This means that the differences between the groups at a lower N are greater than the differences at a higher N and that there is a more significant difference between these two groups.
Furthermore the p-values are extremely low which means that our sample data has a high probability that it occurred by chance.

[3]On ramp cannot be used, equals original Nagel-Schreckenberg

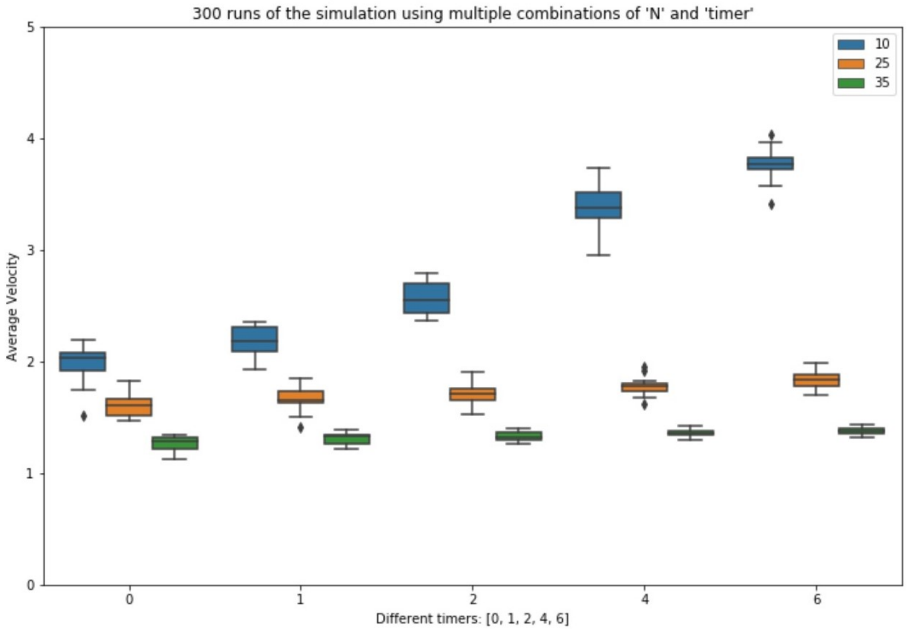

Figure 1: Boxplot showcasing timer length and initial road density on average velocity

7 Conclusion

The results of the experiment give us a better understanding of the model and helps to explore the connections between the individual agents' behaviour and the patterns that emerge from these interactions.
Using the simulation we tried to answer the following research question:

> How does a traffic light and its timing affect throughput on a highway?

As well as a research question we have defined the following two hypotheses that can help with answering the research question:

> H0: A traffic light does not have a positive effect on throughput
> H1: A traffic light does have a positive effect on throughput

As mentioned earlier, we have defined throughput as the average velocity of all the agents. A positive effect being a heightened average velocity, and a negative effect being a lowered average velocity.

As can be seen at the boxplot above in the results there is a significant difference between the amount of cars on the road, the timer and the average velocity.

N	Base	Interval	t-value	p-value
10	101	0	67.97	5.66e−36
10	101	1	76.96	5.31e−42
10	101	2	49.90	2.43e−30
10	101	4	27.97	1.97e−22
10	101	6	14.56	4.87e−14
25	101	0	18.32	8.88e−16
25	101	1	18.41	6.34e−16
25	101	2	22.25	9.31e−20
25	101	4	19, 39	7.27e−19
25	101	6	22.94	8.40e−24
35	101	0	19.05	3.12e−15
35	101	1	22.45	8.56e−18
35	101	2	22.30	3.80e−18
35	101	4	25.55	2.71e−22
35	101	6	19.39	4.90e−18

Table 2: T-test results, see **Simulation Analysis.ipynb** and Discussion for explanation

When the total amount of cars on the road is too great, the timing of the traffic light has no influence on the throughput of the road. Oppositely when the total amount of cars on the road becomes fewer, the timing of the traffic light does matter. When the total amount of cars on the road is low, the length of the interval of the timer of the traffic light has a positive effect on the throughput. This means that the greater the interval of the timer becomes the better the throughput gets.

This is not an entirely surprising result. When less cars get the opportunity to try to merge into the main road there are less cars that can hinder the cars that are already driving on the main road. Nonetheless, based on our current results we have to conclude that when the timing of the traffic light becomes greater the throughput becomes greater, but the timing only has a significant influence when the total amount of cars that already exist on the road is low.

Based on our preliminary results, we reject the null hypothesis under the circumstances of a low initial road density, and high timer duration.

8 Discussion

8.1 Missing measurements

As hypothesised in the section "Results of the experiment", the timer duration positively correlates with throughput on the highway. We did not, however, measure two very important data points:

- The throughput on the on ramp
- The length of the queue forming at the traffic light

Without measuring these two data points, any conclusion about throughput would be incomplete. The current conclusion essentially assumes that only the cars on the highway matter. This is highly unrealistic. It would be worthwhile to include these in the future.

8.2 Braking

Internally, we were not able to come to an agreement about this topic. Two complaints were levelled about the current braking mechanic.

8.2.1 Size of the decrease

If a car is at its maximum velocity of five, and it brakes, its speed gets reduced by one. That is a decrease of 20%. At lower current velocities, this decrease is even steeper.

Some members of our team believe this is unrealistic. They believe it is too great of an decrease.

This large decrease in speed makes the Nagel-Schreckenberg model extremely volatile. While this is, seemingly, beneficial for the regular Nagel-Scheckenberg model[4]. It could negatively affect a more complex simulation like the one implemented for this paper.

Due to the nature of Nagel-Schreckenberg as a cellular automaton, velocities need to be defined as natural numbers. A speed reduction of one is then the lowest one can possibly get. If we used a greater maximum velocity, the percentile decrease would be smaller, and the volatility would decrease.

This would, however, require a larger grid and more steps to get significant results.

The sudden addition of an agent to the road would then be a less intrusive event, which could lead to more accurate results.

8.2.2 Timing of braking

The second complaint levelled about the current braking mechanic is its timing. Currently, each car brakes randomly with $p = 0.5$. According to some members of our team, this is unrealistic. They believe it happens too often.

The $p = 0.5$ is taken directly from the Nagel-Schreckenberg. They also use $p = 0.3$ in other variations of their model. How realistic this value is, is unknown. $p = 0.5$ was implemented simply because it is the standard value.

8.3 Traffic light

Another topic of contention were the dynamics of the traffic light.

[4]The volatility allows results in less steps

8.3.1 Amount of cars allowed to merge

In the current model, the traffic light allows a maximum of one car to merge per time step. In real life scenarios, an on ramp may allow many more cars to merge. This would most likely have a negative throughput on the highway, while having a positive effect on the throughput on the on ramp.

The effects of allowing more than one car would be an interesting avenue to explore in the future.

8.3.2 Road density

Adding behaviour to the traffic light based on road density would be another interesting avenue to explore. The existence of a 'smart' traffic light might lead to better throughput on both highway and on ramp.

8.3.3 Rate of adding cars

Currently, during each time step two cars gets added to the traffic light's queue. If allowed at all, only one car may enter the on ramp's queue. This means that, no matter what, the traffic light's queue grows uncontrollably.

The uncontrollable growing of the queues needs to be solved. This might be a property where some measure of non-determinism could come in useful. It is unlikely that this many cars wish to merge at each step, and that the amount is constant. Further research into the amount of cars that make use of an on ramp is required.

8.4 Removal of cars

In the current model, whenever a car originated from the on ramp reaches the loop in the Torus, it gets removed. Cars placed during initialisation never leave the model. This temporarily breaks up the congestion between the start of the Torus[5] and the start of the on ramp.

This ensures that, eventually, all cars that merge onto the highway also leave it. This also helps to manage the amount of cars that exist on the road and will eventually prevent the over-flooding of the road. The rate cars get added per time step, and the rate they get removed are not at all equal. If the timer duration is small, it causes instant congestion.

While the way cars leave the model is not necessarily the problem here (the way we currently add cars is), but altering our current method *could* alleviate the issues caused by the main problem.

[5] Effectively an off ramp

8.5 T-test

To give more certainty about our results we have tried to do what's called a *Student's t-test*.[6] This test compare two means and tells you if they are different from each other and if so how significant these differences are.

However since we're inexperienced with this type of testing, there could be a possibility that we've made an error while computing the t-values or p-values. Therefore the possibility exists that the computed results of our t-test are flawed.

The results from the executed T-test implied a very small significance to our results. We are, however, not sure if this is due to a bad implementation of the T-test, or due to the results themselves.

If we look at the averages, however, there does seem to be some significant difference between settings. That is why we decided to reject our null hypothesis and accept its alternative.

8.6 Realism

The final point of discussion is realism. Nagel-Schreckenberg, and our adaption of it, leans more toward a minimalist simulation. As many events and behaviours are abstracted away.

Some members of our team again argue that this lack of realism negatively affects the simulation. Others argue that the minimalist nature of Nagel-Schreckenberg is in fact its strength.

The value of Nagel-Schreckenberg itself aside, the question remains if expanding Nagel-Schreckenberg the way we did even makes sense. If any element of the model's definition is excluded, it would result in an immediate loss of crucial properties of traffic. That might also entail that Nagel-Schreckenberg is too minimalist to be properly expanded upon.

Adding comparatively complex behaviour to a simple system without first expanding the original behaviour, might cause incompatibility between the new behaviour and the original system.

[6] `https://en.wikipedia.org/wiki/Student%27s_t-test#Independent_two-sample_t-test`