SCHOOL OF PHYSICS AND ASTRONOMY

YEAR 3 FINAL PROJECT REPORT
SESSION 2020-2021

Name:	**William Hughes**
Student Number:	**c1803626**
Degree Programme:	**Math and Physics**
Project Title:	**Using Cellular Automata traffic systems to demonstrate the emergent complexity of Cellular Automata systems, and to predict Gridlock in road networks**
Supervisor:	**Dr G. Pisano**
Assessor	**Dr A Papageorgiou**

Declaration:

I have read and understand Appendix 2 in the Student Handbook: "Some advice on the avoidance of plagiarism".
I hereby declare that the attached report is exclusively my own work, that no part of the work has previously been submitted for assessment (although do note that material in "Interim Report" may be re-used in the final "Project Report" as it is considered part of the same assessment), and that I have not knowingly allowed it to be copied by another person.

Using Cellular Automata traffic systems to demonstrate the emergent complexity of Cellular Automata systems, and to predict Gridlock in road networks

Abstract

This project examines the nature and study of Cellular Automata, and uses a Cellular Automata model of a traffic network to show that Cellular Automata systems can be complex. An emergent property of the traffic model, namely, the time period that is required for gridlock of the network is analysed quantitatively and compared to principle variables associated with the network to build a formula for predicting its behaviour.

Contents

1 – Theory, Context and Aims

1.1 - The Theory of Cellular Automata

A Cellular Automaton (CA) is a program, whose functioning can be understood in three steps:

1. Declare a field of cells
2. Update that field based on the CA rules
3. Re-define the field as its updated version

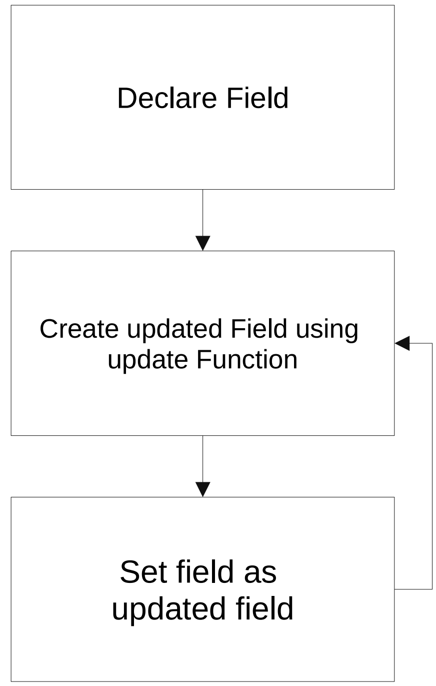

Figure 1: Flow Chart of Cellular Automata process

The field can be treated computationally as array, its only requirement is that it is split into discrete portions, the cells. Each cell has a location that relates it to the wider system, and a state, which is what the update function changes. Cells can have many different states, but for many models two states suffice, which can be treated as on or off, 1 or 0, or in the case of the traffic model in this paper, car or no car.

The complexity in a CA arises from the rule it uses to update, its update function. This function technically takes the entire field of the CA as an input, and generates a new field as an output, but for both ease of application and understanding, the function is usually defined across neighbourhoods, small regions of cells inside the field. This definition of the function acts on a specific cell, taking the neighbourhood of a cell as an input, and outputting a new state for the cell . The update function is then applied to every possible neighbourhood across the field, and so results in a totally new field being produced.

1.2 - Complexity

CA models are excellent for simulating complex phenomena with computational ease, because their discrete foundation reduces the number of calculations required to process changes in the system. They also have inherent complexity, which is something computer systems often struggle to simulate efficiently. Complexity is a property that a system can have. A system is complex if from simple, localised rules across small components, larger, less directly predictable properties emerge. In particular, Warren Weaver's idea of disorganised complexity (1948) is looked at, wherein the systems emergent properties arise from interactions between stochastic values and the small-scale predictable rules. The classic example of a disorganised complex system is a sample of gas, where Newtonian Mechanics are the simple underlying rules, and the stochastic values are the energies of the gas molecules .

CA like that approached in this model inherently posses the first characteristic; in being defined across a small neighbourhood, the update function can be considered as the simple localised rule. However, the second condition for complexity, an emergent property, is not guaranteed. To indicate complexity, emergent properties cannot be completely random, but they also cannot be dependent on the underlying rule; in the case of CA, they must depend on some macroscopic property of the system, rather than the update rule.

1.3 - The Study of Cellular Automata

The study of CA sits at an intersection of three fundamental fields: Mathematics, Computer Science and Physics, and in some ways represents a microcosm of all three. In the same way that mathematics informs computer science, which is then used to better understand and apply physical principles, CA research started in the realms of the theoretical, then graduated into simple, obvious application, before becoming profound and generating results in fields that one could have expected.

The study of CA began with the efforts of John von Neumann to design a theoretical system capable of replicating itself, which resulted in the first known Turing complete cellular automaton, which was a 29-state 2D system. A thorough account of this is that of Palash Sarkar (2000), which then goes on to describe the work of Stephen Wolfram, who took von Neumann's ideas and attempted to create an elementary framework for CA, through rigorous research. He detailed this very extensively in his book, A New Kind of Science (2002), where he presented his discovery of the fundamental 1D, 2D and 3D CA, and then went on to further argue that CA models are so-called "simple programs", and should be studied in a field of their own, in the same way that physics or chemistry is studied, in an effort to discover more about how natural phenomena arise.

This philosophical direction is rare in the field of CA, as is the sheer detail in A New Kind of Science. Instead, most of the focus is on the practical models that have been built using Cellular Automata, such as the Nagel-Schreckenberg model for traffic flow [Jason Merritt, (2012)], or Lattice Gas automata [Bostjan Mavric, (2013)].

A New Kind of Science is considered a seminal work in the field of CA study, in part because it represents the culmination of almost two decades of work on the part of Stephen Wolfram, but also because it is the first work that represents CA models in a truly accessible way. This treatment was uncommon, even for Wolfram himself before he published his book, and the vast majority of discussion around CA models until then was focused on their mathematical characteristics, particularly whether they were Turing complete. Most of the papers from this period are now compiled into textbooks, like that of E.F. Codd (1968), which are rigorous to a mathematically compliant extent, but this process of forming mathematical understanding continues into recent times; one of Wolfram's elementary rules, rule 110, was shown to be Turing Complete within the last twenty years[Cook, (2004)].

However, most simple CA models have been considered mathematically at this stage, and so theoretical attention has now turned to the possibilities of creating CA models in the new computing paradigm, Quantum Computing (QC). There is much range in the discussion of this field, from theoretical introductions [Joshua Horowitz (2008)], to the classic focus of trying to prove Turing Universality for quantum CA models [Tougaw et. al, (2020)]. As quantum computing is in its infancy, practically at least, the results of these theories are yet to be borne out, so it is difficult to comment on their significance, but they show at least that the study of CA is a field still driven by significant theoretical enterprise.

This is quickly changing though. The applications of CA models are quickly being shown to be broad and profound, and with increasing computing power more readily available than ever, research applying CA models to real situations is becoming more and more common. Many efforts model straightforward phenomena that are already cellular in nature in real life, such as biological life. This research sometimes focuses on the spread of disease through the immune system [Beauchemin et al (2004)], or through a certain organ [Zhu et al, (2004)]. The success of these models varies; the more complex the model, the more accurate to real life data it ends up being, but they struggle to generate large volumes of data, as they use far more computing power than the simpler models. Often all these models are able to show is general rules, rather than numerically significant results.

In addition to modelling inherently cellular phenomena, CA models can also be used to understand less obvious situations. CA models compute an entire time step at once, and so can simulate complex interactions that don't conform to simple mathematical rules very well. Situations like this are very difficult to approach computationally, as they require exponential amounts of processing power. The obvious example is fluid flow, where unbelievable numbers of interactions result in complex and borderline chaotic behaviour. However, using CA, these processes can be made more accessible, and can be simulated very efficiently [Salles et al, (2006)], allowing for practical applications. This also allows stochastic phenomena such as flooding, to be analysed and predicted [Douvinet et al, (2013)] which is of the utmost importance. A less obvious example of this type of research is in studying the behaviour of groups of people. CA models have been used to understand the actions of small groups of pedestrians heading en-masse to a point, like a door [Yamamoto et al (2007)], all the way up to entire populations urban spread [Xiaoping et al. (2013)]. The advantage of models like this is in their ability to simulate complex phenomena in a computable and accessible way. They all struggle to some extent with accuracy though, as they inherently sacrifice detail by approximating practically continuous phenomena as discrete.

1.4 – Aims

This project aims to build a quantitative understanding of a novel and original CA designed for simulating traffic, in particular the predictable phenomena that emerge from its underlying simple rules. The model built should not be computationally intensive, and should be able to yield large amounts of data. This data will then be analysed to look at the trends in how the model behaves, and to understand any emergent properties that the network may have, and the variables that affect these emergent properties. These affecting variables should be properties of the system as whole, rather than related to the underlying update rule, which will demonstrate that the system possess disorganised complexity.

2 – Methodology

2.1 – Rule 184

As previously mentioned, the study of traffic using CA has been a promising field for a little while. The most famous model is the Nagel-Schreckenberg model (1992), which works using a 1D CA field, and a four step update function. This model can produce many varied traffic phenomena, but is computationally complex to implement, requiring multiple passes of the field to update, as well as branching conditional rules, and up to five cell states. This means it can't be used to form 2D networks of roads in a computationally efficient way, and so simulating more complex traffic networks with it is not possible. However, when simplified, the Nagel-Schreckenberg model can be reduced to Wolfram's elementary rule 184, a computationally simple rule, requiring none of the processes that make Nagel-Schreckenberg computationally intensive, allowing for much more complex models to be produced.

The use of Rule 184 for the model in this paper takes into account the nature of traffic flow into both its field and update function. The field of the CA is 1D, and so can only change in one direction, simulating the movement of traffic on a road in the real world, where cars essentially move linearly unless they leave the road as a whole. While on the road, cars in the real world follow a fairly simple rule: they move forward if there is a space to move into, and stay in place if there is not a space. When Rule 184 operates on a cell, it takes the two adjacent cells as the neighbourhood, in the same way that a driver is only aware of the cars in front or behind them.

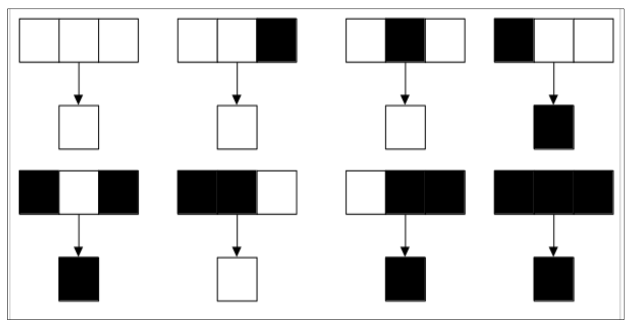

Figure 2: Rule 184. The groups of three cells are the neighbourhood, while the result of the update function is the single cell underneath

The cell diagram of the rule is given in Figure 2, where each possible neighbourhood configuration is mapped to the state of the cell that the update function outputs. More simply though, the rule can be understood as follows:

1. If a cell is on, and its right-hand neighbour is off, then the cell turns off.
2. If the cell is off, and its left-hand neighbour is on, then the cell turns on.
3. Otherwise, the cell retains the state it was in before.

Considering each cell as a segment of road, Step 1 simulates a car leaving a space to move forward, Step 2 simulates a car moving into a space, and Step 3 simulates a car staying still. This CA exhibits all the properties required of it; upon activating a cell at one end, the cells will all activate in sequence, until the final cell in the CA is activated, which then turns itself off. These moving activations behave like cars on a road. Blockages in the road can be simulated by keeping a cell permanently turned on; any time the first cell is activated, the "car" will move across the field until it reaches the blockage, at which point it will stop until the blockage is removed. Blockages can result in Rule 184 CA becoming congested, where no new cars can be added until the blockage is removed, a state called gridlock. Figure 3 shows the progression of two Rule 184 CA models. In the top one, the "car", or activated cell moves along the CA unimpeded, whereas in the second one, the car reaches a blockage in time step 4 and can no longer progress. The activated cell representing the car will stay activated until the blockage is removed.

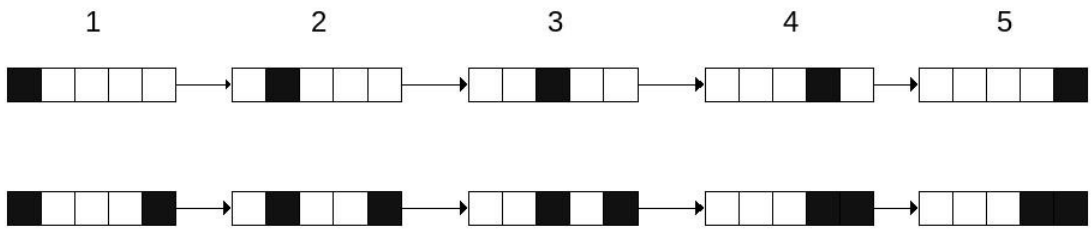

Figure 3: Diagram of two rule 184 CA over 5 time steps, one with a blockage, and one without

2.2 - Road Networks

In isolation, rule 184 CA are useful for simulating the motion of cars, but don't necessarily exhibit complex behaviour. However, in the real world, road networks are often made up of multiple roads, so to model one of these road systems, a method wherein these CA can be combined must be utilised. To do this, an object called an intersection is incorporated, which has a single property, a numerical value.

When a car reaches the end of a CA, representing a segment of road, the intersection receives the car, and increments its own value. The next time step, it re-allocates the car to another road by activating the first cell of a random road that attaches to the intersection, and then decrementing its own value. With the addition of these intersections, a network is established, made up of a grid of the Rule 184 CA, that simulate roads. Cars, simulated by activated cells,

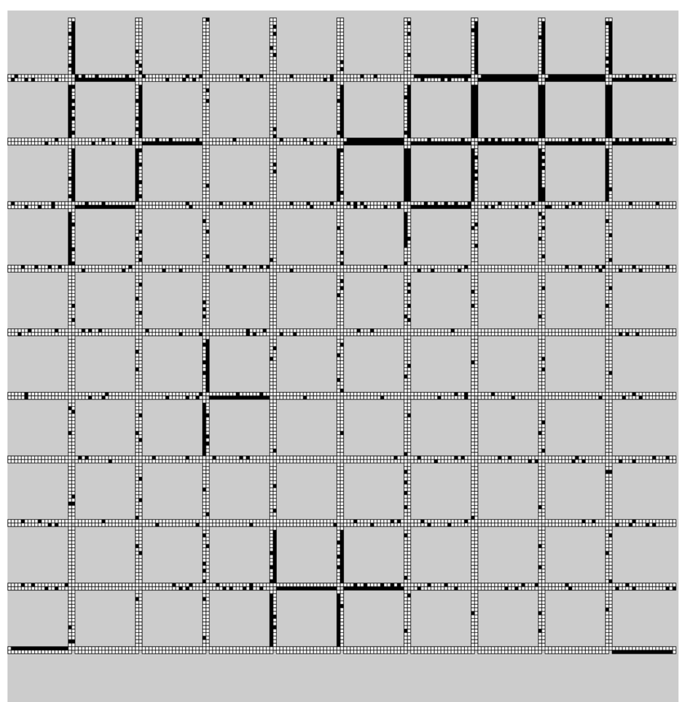

Figure 4: A frame from a visual model of a network of Rule 184 CA. The black squares are cars, and the congested roads have begun to develop. This network is a 10 by 10 network.

move across this network according to Rule 184, randomly moving from road to road via intersections. The first cells in the roads that sit at the edge of the network are randomly activated, adding cars to the network, that then travel until they reach the end of the roads that end at the edge of the network. Cars that reach this point are then removed. Figure 4 is a frame from a visual demonstration of the model, which can also be seen in action by clicking here if this report is being read digitally.

2.3 – Congestion and Gridlock

Intersections are given a capacity, the maximum number of cars that they can hold. If the intersection is full, then a road cannot allocate the car at its end to said intersection, and so that car is held there until the intersection becomes free again. In figure 3, there are some roads in the network that are entirely populated, or are congested, because the intersections they lead to are at capacity. This is a natural progression of all networks that have cars being added, so long as there are more cars arriving than leaving, the number of congested roads will increase over time. The network will eventually reach a state where no cars can pass through it, at which point the road system will have failed, a situation called gridlock. When designing roads, this state of gridlock is to be avoided, and so being able to predict this phenomenon would be of great value to road network planners in the real world. On a deeper level, gridlock is a predictable state emerging from a stochastic system governed by simple rules, and so is an indicator for complexity. However, it must be shown that it is indeed predictable and that it depends on properties of the entire system to show that complexity is indeed arising here. And so, a mathematical model for gridlock of the network must be found. To do this, it is necessary to analyse what is happening at the CA scale of the network to make inferences about the values that could affect gridlock, as well as how it can be measured.

A given road in the network will become congested if it is unable to place as many cars on to the intersection it leads to as it takes in. In keeping with real world intersections, the model allows up to four cars on an intersection at a given time. Once a road becomes congested, the intersection that adds cars to it can no longer do so, and initially localised congestion spreads throughout the network. This implies two principles:

1. Full intersections can be used to measure the level of congestion, and so whether gridlock has been reached
2. The spread of congestion means that it is inevitable that after a certain critical amount of congestion has been reached, the system will inevitably reach gridlock

The first idea implies that it is possible to find the number of time steps required to reach gridlock, here to after referred to as T(Gridlock), by checking the number of intersections that are occupied. The second idea means that once a certain number of intersections is full, then it is known that the network is guaranteed to become gridlocked. This critical proportion of filled intersections is not the focus of this investigation; so long as the value used in the experiment overshoots the actual value, then the results relevant to this projects aims will be unchanged, and so it is taken as 0.25, i.e. once the number of filled intersections is greater than a quarter of the total intersections, then the system is considered gridlocked:

$$T(\text{Gridlock}) = \text{Time step at which a quarter of the intersections are full}$$

2.4 – Affecting factors of T(Gridlock)

Gridlock is a product of capacity; if the network contains the maximum number of cars then it is gridlocked. As such, its relationship with the key factor that affects the number of cars on the network, P(Car), can be studied:

$$P(\text{Car}) = \text{Probability of a new car being added to a given road on the edge of the network}$$

In this investigation P(Car) is fixed for each time T(Gridlock) is found, and so over a given time step, the total number of cars arriving at the road is simply P(Car) multiplied by the number of roads at the edge of the network. So for a given network it is reasonable to expect T(Gridlock) to decrease as P(Car) increases, as the number of cars being added has grown, and so the system will head to capacity more quickly. However, larger networks have more intersections, and so it is would take longer for cars to fill the network. T(Gridlock) is expected to be increased for these larger networks when P(Car) is kept the same.

Finding T(Gridlock) is a simple matter of running the model with a given P(Car), and measuring how many intersections are at capacity each time step. Once the number of intersections is equal to, or exceeds, a quarter of the total number of intersections, then the test stops, and the number of time steps that have passed since the test started is recorded. Repeating this across different networks and different P(Car) will give indications as to the behaviour of T(Gridlock).

3 – Results and Analysis

3.1 – T(Gridlock) against P(Car)

The above test investigating the relationship between T(Gridlock) and P(Car) was carried out for square networks, i.e. networks with the same number of roads going down as across, from P(Car)=0.01 to P(Car)=0.99, with interval 0.01. For each P(Car), the simulation was run seventy five times, and the T(Gridlock) of each of those seventy five simulations was averaged. If a given simulation reached 15,000 time-steps without reaching gridlock, then it was stopped, and was listed as a DNF result. If 10 repeats for a given P(Car) went to DNF, then that probability's testing was stopped, and the next probability was tested. This only occurred for low probabilities (P(Car)<0.05), although there were a mix of gridlocked finishes and DNF finishes for P(car) up to 0.1.

The general relationship between P(Car) and T(Gridlock) is exactly as expected (Figure 5). When the probability is sufficiently low (<0.05), the system operates continuously without reaching gridlock, which is shown as the flat region before log(P(Car))=-1.5 on Figure 6. This is the situation where as many cars leave the network as arrive each time-step, and so the road network never reaches capacity. Once P(Car) gets large enough to cause gridlock however, T(Gridlock) decreases exponentially as P(Car) grows, implying that:

$$T(Gridlock)=\frac{k}{P(Car)^{\alpha}}$$

By calculating the gradient of a log/log graph of T(Gridlock) against P(Car) [Figure 5] using the standard linear regression formula, negative values for the power, α, for each network looked at can be found. These values are given in table 1, with the standard error of regression.

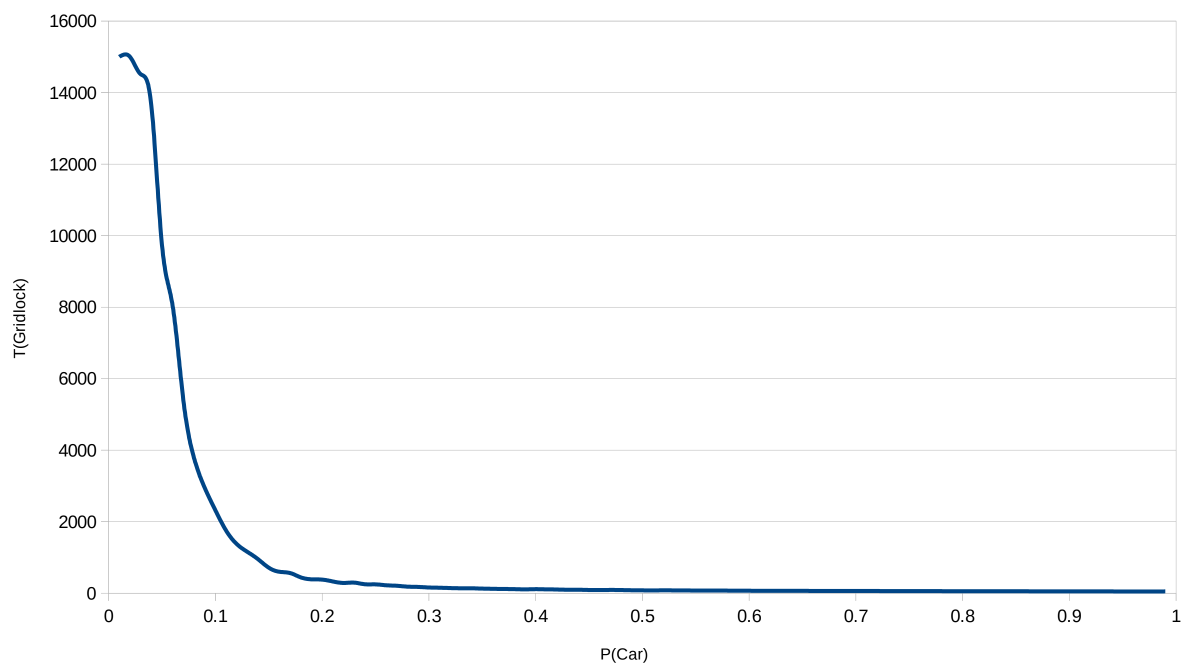

Figure 5: The relationship between T(Gridlock) and P(Car) is shown here for a network with 6 roads going across, and 6 going down

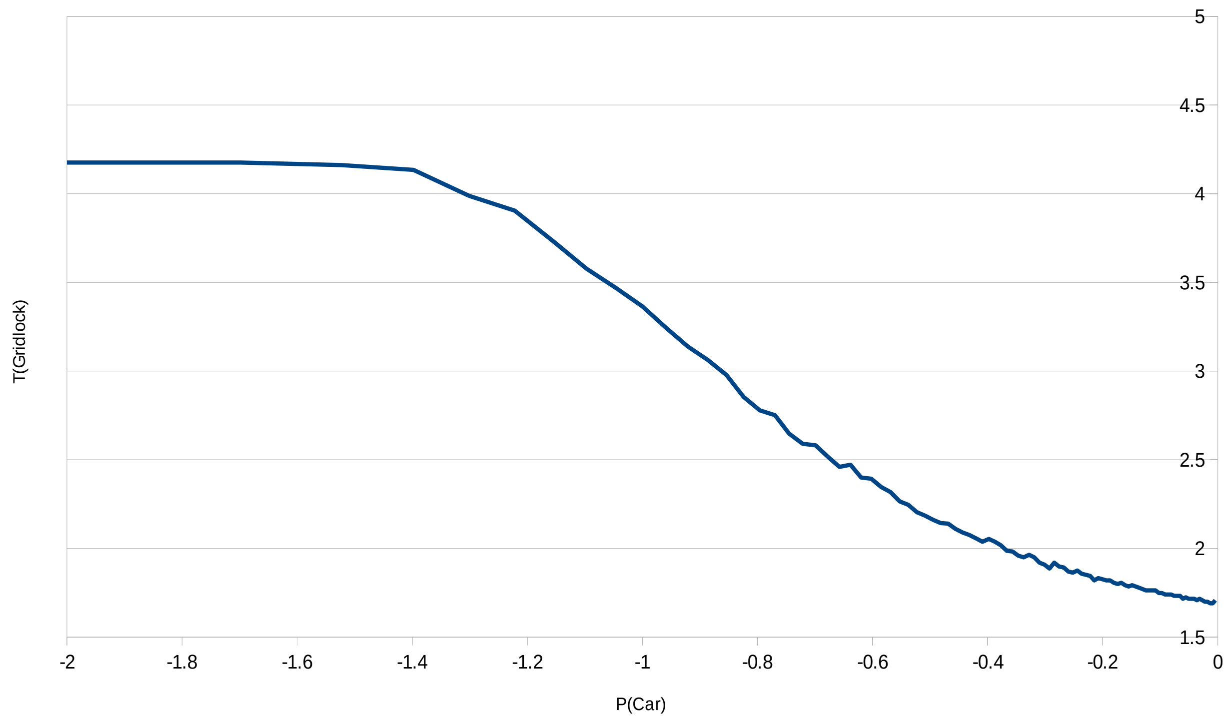

Figure 6: Graph Showing the linear relationship between Log(T(Gridlock)) and Log(P(Car)). The initial flat section corresponds to the period where gridlock did not occur

Table 1	
Road Network	**α**
4 by 4 network	1.76±0.21
5 by 5 network	1.67±0.17
6 by 6 network	1.61±0.14
7 by 7 network	1.57±0.12
8 by 8 network	1.55±0.12
9 by 9 network	1.53±0.11
10 by 10 network	1.48±0.10
11 by 11 network	1.50±0.10
12 by 12 network	1.49±0.14

The power, α, slightly decreases as the size of the network grows, approaching α=1.5. The behaviour of the networks at low probabilities is very random; for the six by six network who's behaviour was described in Figure 4, the standard deviation of the first probabilities after the networks started becoming gridlocked is in Table 2:

Table 2		
P(Car)	**Average T(Gridlock)**	**Standard Deviation of the Average T(Gridlock)**
0.06	9731	4006.47
0.07	8028	4120.48
0.08	5396	3162.09
0.1	3779	2166.09
0.11	2944	1540.74

These early probabilities have very high variance, and are responsible for much of the randomness in the calculated values of the constants for the formula for T(Gridlock), and are the reason that α approaches 1.5 as the size of the network increase, rather than actually being at that value for all networks. The larger networks are less susceptible to random factors than the smaller ones, and so they are more reliable sources of indication as to the actual nature of the relationship between T(Gridlock) and P(Car). For any theoretical values produced in this project, α is taken as 1.5.

3.2 – Congestion Constants

Using the value of α, it is possible to find the value k, the "Congestion Constant", for each network, using the intercept of the linear regression of the log/log graph of T(Gridlock) against P(Car). The values of k for all networks tested are listed in table 3, with their standard deviations:

Table 3	
Road Network	**k**
4 by 4 network	14.64±10.00
5 by 5 network	25.08±12.48
6 by 6 network	33.49±12.87
7 by 7 network	43.65±13.95
8 by 8 network	51.89±15.12
9 by 9 network	64.10±17.28
10 by 10 network	78.52±18.88
11 by 11 network	138.9±39.82
12 by 12 network	338.81±139.78

The origin of the value of k is not immediately obvious, but when compared to the number of intersections (referred to as I) that each network has, a relationship emerges. For a square network, with n roads across and down, there are $(n-1)^2$ intersections[1], and so the following is acquired:

Table 4		
Road Network	**Intersections (I)**	**k**
4 by 4 network	9	14.64±10.00
5 by 5 network	16	25.08±12.48
6 by 6 network	25	33.49±12.87
7 by 7 network	36	43.65±13.95
8 by 8 network	49	51.89±15.12
9 by 9 network	64	64.10±17.28
10 by 10 network	81	78.52±18.88
11 by 11 network	100	138.9±39.82
12 by 12 network	121	338.81±139.78

The graph of log(k) against log(I) has a linear regression gradient of 1.001, strongly indicating a linear relationship between the two values, in turn implying that T(Gridlock) is linearly dependent on the number of intersections in the network. Using the standard linear regression formula, the constant of proportionality relating the value of k to the number of intersections can be found to be m=2.230. However, the values of k for the first and last networks, the two most extreme values, are discounted this constant drops down to m=1.188. In turn, the intercept of the log/log graph can also be used to get a third, and different value, for the constant, m=1.361.

3.3 – A formula for T(Gridlock)

Returning to the originally posited equation for T(Gridlock) in terms of P(Car), its form can be updated with the dependency on the number of intersections added:

$$T(\mathrm{Gridlock}) = \frac{mI}{P(\mathrm{Car})^{1.5}}$$

where I=number of Intersections
m=Constant

1 It is $(n-1)^2$ rather than n^2 because the final road for each row/column of roads doesn't lead to an intersection, and so one must be subtracted from the value being squared.

As previously seen, the value m is not reliably acquired from the data; three possibilities for it were immediately obvious. Despite the fact that m is difficult to imply, it is possible to test it against values from the original data. Using

- m_1= 1.188
- m_2= 1.361
- m_3= 2.230

graphs of the theoretical values that the above model produces, alongside the average values from the simulation (Still using the 6 by 6 network) against P(Car) can be made (Figure 8). From the graphs, two things are clear; the value of α=1.5 is correct; the straight, theoretical lines on the log/log graph, which have a gradient of α, follow the trend of the curved simulation line very accurately. In addition to this, all of the possible m, which are the y-offsets of the straight lines, could be potentially correct; the simulation line sits closely with all of them. The only region where this is not the case is when P(Car) is low (<0.1), a reflection of the increased randomness that the system demonstrates when it is moving from never going into gridlock to becoming gridlocked.

To find which value of m is the correct one, more simulations would have to be carried out, with more repeats of individual P(Car) values to ensure accuracy of the value k, and more networks would have to be tested to broaden the data points to be found when looking at m.

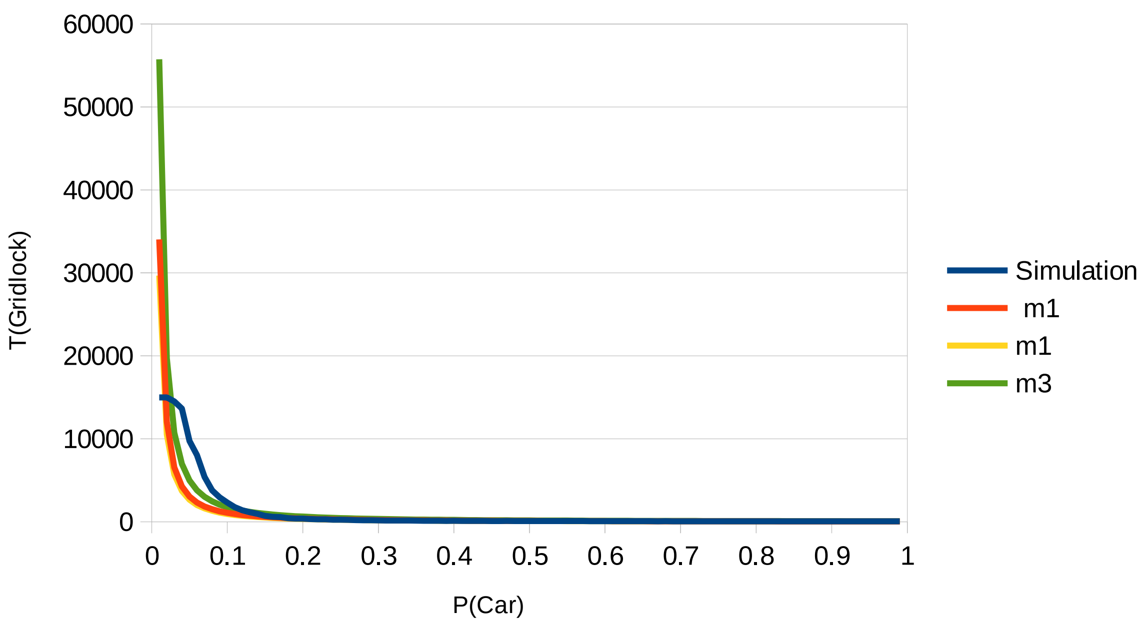

Figure 7: Graph showing the Average Simulation T(Gridlock) against the Theoretical T(Gridlock), using the three potential values of m, all against P(Car)

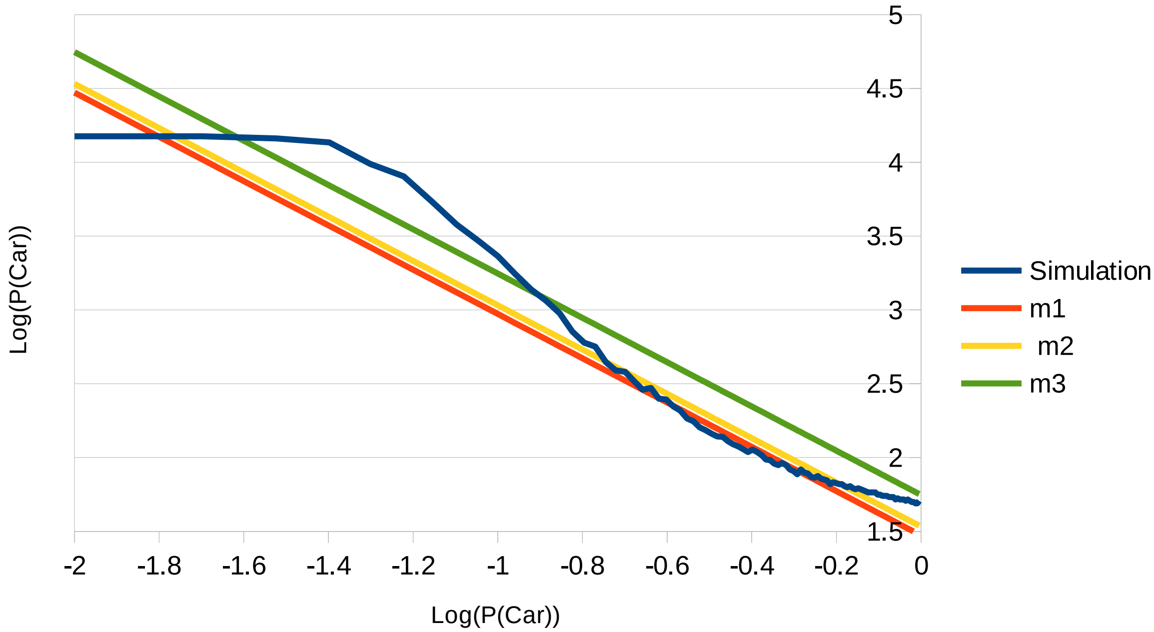

Figure 8: Log/Log Graph showing the log of the Average Simulation T(Gridlock) against the log of the Theoretical T(Gridlock) values, using the three potential values of m, all against P(Car)

4 – Conclusions

The road network described in this paper demonstrates clearly the potential of cellular automata to form complex systems, and simulate them efficiently. The model was built and run on an Intel I3 processor, and the simulation was run 6750 times without crashing or breaking once. On a computer with more power, it would be possible to test each P(Car) thousands of times, and produce even more compelling data than that presented in this paper.

The individual and simple rule 184 CA are combined to form a network with emergent characteristics that bear no connection to the simple starting rules. Upon analysis of the model, and the data it produced, the emergent property T(Gridlock) can be observed. T(Gridlock), the time at which a network with a given number of intersections becomes unable to function was measured in relation P(Car), the probability of a car arriving on the network. By testing the square networks (networks with the same number of roads going across as going down) the following relationship was found:

$$T(Gridlock) = \frac{mI}{P(Car)^{1.5}}$$

where I=Number of Intersections and m=Constant

Further analysis of the data led to several values for the constant m being calculated, and while they all generally predicted the behaviour of T(Gridlock), more data would have to be collected to definitively state its value. However, it is clear that T(Gridlock) depends not on Rule 184, but on wider properties of the system, namely P(Car) and the number of intersections I. This means that this system is indeed complex: it has an underlying rule, rule 184 with intersections, and an emergent property, T(Gridlock), which depends on other macroscopic factors rather than the underlying rule. The implication in turn is that it is possible for CA models to be complex. However, this only shows disorganised complexity, i.e. there is a random effect also at play, the allocation of cars on to the network, and between roads by intersections. A potential direction to take this project would be to look at whether it has organised complexity, by making these random values predictable instead, and then looking at the behaviour of the system.

Another avenue this project could go down would be to carry out more simulations to further establish the relationships discovered here. Specifically, testing the value of $\alpha=1.5$ is important. The propensity of smaller networks to become congested more easily at lower P(Car) meant that some of the measurements of α were inaccurate. More simulations at each P(Car) would eliminate this uncertainty, and give greater confidence in the model's general trend. On eliminating the dependency on P(Car), it was observed that the dependency of T(Gridlock) on the other major factor of each network, the number of intersections it had (I), was linear. The gradient of the log/log graph of these values against each other was almost exactly one strongly implying the linear dependency of T(Gridlock) on I.

Finally, the constant of proportionality between T(Gridlock) and I was investigated. However, too few networks had been tested to reliably establish its value. Referred to as m in the above equation, three very different values for it were found, and when placed next to the data from the simulation, all three values could have been correct; they all predicted the general behaviour of T(Gridlock), but the simulation values were too variable to pin down the actual value of m. To get a better value of m, more networks would need to be tested, in particular non-square networks, which would require no additional computational power, but would serve to reinforce the rules already established, get more accurate values of m, and to highlight any influence that the layout of networks has on the system.

When T(Gridlock) was first defined for this project, it was taken that if the proportion of intersections occupied was higher than 0.25, then the the system was guaranteed to end up in a state where every road in the network was fully occupied by the cars. This was deliberately an attempt to over-estimate the proportion required for the network to stop functioning, but it would be an interesting line of enquiry to establish both whether this value is indeed sufficient to imply gridlock in the networks future, and also to find the smallest possible proportion such that gridlock is in the networks future.

Using a CA road network to demonstrate complexity has broadly been successful. While there is still uncertainty about the specifics of the emergent behaviour of the network, the existence of this emergent behaviour is undeniable, as are its dependencies on inherent factors in the network. This in turns shows that this network is complex, and that CA models in general can be complex.

References

Warren Weaver, 1948, *Science and Complexity,* American Scientist, 1948-10, Vol. 36(4) , pp. 536-544, DOI: Unavailable

Palash Sakar, 2002, *A Brief History of Cellular Automata,* ACM computing surveys, 2000-03-01, Vol.32 (1), p.80-107, DOI: 10.1145/349194.349202

Stephen Wolfram, 2000, *A New Kind of Science,* Wolfram Media Inc. 2000

Jason Merrit, 2013, *Cellular automata as emergent systems and models of physical behavior,* Available at merritt.pdf (uiuc.edu), Accessed 18/12/2020

Bostjan Mavric, 2012, *Physical Applications of Cellular Automata,* Available at cellautom.pdf (uni-lj.si), Accessed 18/12/2020

E.F. Codd, 1968, *Cellular Automata,* New York : Academic Press 1968.

Cook, 2004, *Universality in Elementary Cellular Automata,* Complex Systems. 15: 1–40. DOI: Unavailable

Joshua Horowitz (2008), *An Introduction to Quantum Cellular Automata,* Available at download (psu.edu), Accessed 18/12/2020

Tougaw et al, (2020), *Designing a Turing-complete cellular automata system using quantum-dot cellular automata,* Journal of computational electronics, 2020-09, Vol.19 (3), p.1337-1343, DOI: 10.1007/s10825-020-01518-1

Beauchemin et al, (2004), *A simple cellular automaton model for influenza A viral infections*, Journal of theoretical biology, 2005, Vol.232 (2), p.223-234, DOI: 10.1016/j.jtbi.2004.08.001

Zhu et al, (2004), *Facilitating arrhythmia simulation: the method of quantitative cellular automata modelling and parallel running,* Biomedical engineering online, 2004-08-30, Vol.3 (1), p.29-29, DOI: 10.1186/1475-925X-3-29

Salles et al, (2006), *Cellular automata model of density currents,* Geomorphology (Amsterdam, Netherlands), 2007-07, Vol.88 (1-2), p.1-20, DOI: 10.1016/j.geomorph.2006.10.016

Douvinet et al, (2013), *A flash flood hazard assessment in dry valleys (northern France) by cellular automata modelling,* Natural Hazards, 2015-02, Vol.75 (3), p.2905-2929, DOI: 10.1007/s11069-014-1470-3

Yamamoto et al (2007), *Simulation for pedestrian dynamics by real-coded cellular automata (RCA),* Physica A, 2007, Vol.379 (2), p.654-660, DOI: 10.1016/j.physa.2007.02.040

Xiaoping et al. (2013), *Simulating urban growth by integrating landscape expansion index (LEI) and cellular automata,* International journal of geographical information science : IJGIS, 2014-01-02, Vol.28 (1), p.148-163, DOI: 10.1080/13658816.2013.831097

Kai Nagel and Michael Schreckenberg (1992), *A cellular automaton model for freeway traffic*, J. Phys. I France, 1992-12, Vol.2(12), p.2221 – 2229 DOI:10.1051/jp1:1992277