Improved Macroscopic Traffic Flow Modelling with Cellular Automata

Noah Gardner*
ngardn10@students.kennesaw.edu
College of Computing and Software Engineering
Marietta, Georgia, USA

Andy Vu*
avu5@students.kennesaw.edu
College of Computing and Software Engineering
Marietta, Georgia, USA

Olivier Tran*
otran@students.kennesaw.edu
College of Computing and Software Engineering
Marietta, Georgia, USA

Sanjeev Khemani*
skheman1@students.kennesaw.edu
College of Computing and Software Engineering
Marietta, Georgia, USA

Abstract

This paper proposes a method cellular automata-based method in which city traffic and its affection by factors such as distributed occurrence, evanescence, and time variation is simulated on a two-dimensional grid. The variables used in this new experiment include diagonal paths, one-way streets, pre-generated flows, varying speeds, and different types of roads. With these additional factors in the experiment, the results of the new simulation may yield results that more accurately reflect the environment of real-world traffic.

Keywords: traffic flow, cellular automata

ACM Reference Format:
Noah Gardner, Andy Vu, Olivier Tran, and Sanjeev Khemani. 2018. Improved Macroscopic Traffic Flow Modelling with Cellular Automata. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 Research Statement

There are a few interesting changes we could make to the original algorithm in order to explore the impact of different traffic environments and flows. For example, the original paper randomly generates *traffic flows* which are an abstracted representation of the traffic that actually represents a group of vehicles as a random amount of traffic. Additionally, when each traffic flow is generated, the destination of the flow is also generated randomly. After the traffic flow is generated and the destination is chosen, the traffic flow chooses it's path automatically as the shortest distance from the starting cell to the current cell. Finally, the original paper uses one traffic environment with simple two-way roads that are placed on the grid laterally and vertically.

One improvement could allow for modelling of different types of roads. For example, rather than the flow taking the shortest path from the source to the destination, each cell could have a parameter for the weight of adding that cell to the path of the traffic flow. If the weight of adding the cell to the path is low, that cell may be modeled as part of a road that can handle more traffic such as a highway. The original experiment also did not test diagonal or one-way streets, so that may be valuable for a traffic flow model as well.

2 Related Works

Pang and Yang use a cellular automata model to simulate traffic flow on a two-dimensional grid [5]. Traffic flows are randomly generated that have a certain amount of traffic (low or heavy) and also have a random destination. There exists a grid with cells filled in which are meant to model the roads for the traffic flow to move on. On each timestep, the traffic flows move towards their destination using a minimum distance method. The cells can hold a certain amount of traffic capacity. When the traffic capacity for a cell is reached, the cell will not allow new traffic flows to enter the cell. The cells also have a traffic volume, which helps the programmer to see at a glance which cells have high and low volumes of traffic.

Nagel and Schreckenberg have gone away from the typical fluid dynamical approaches to traffic flow studies during this time to the use of a more revolutionary technique [4]. The need for this was to introduce a factor of laminar start-stop waves with increasing vehicle density with an emphasis on human behavior a traffic environment. In their model, it is represented as a single dimensional array where each cell may be a vehicle with functions to accelerate, slow down, as well as randomized velocities. These functions are to represent the randomness in human behavior and varying conditions that are expected during traffic. The authors concluded that a computational way to simulate traffic flow,

*All authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

was more advantagous to a fluid dynamical approach that incorporated real world behavior of a driver but also retained the key aspects of fluid dynamic studies.

Nagatani implements a two lane model as an extension of a one dimensional cellular automaton to monitor lane changing [3]. Their model is essentially two 1-dimensional lattices to represent a two lane roadway. Rules using arrows within the lattice were deteremined to allow cars to change lanes at certain times such as when a car in front was blocking the path to progress forward from either the right to left direction or vice versa. These rules were dependent on the time step. Overall the results showed that as the cars general velocity increased, there was a usage of lane shifting without cars obstructing progress. As density increases, the maximal velocity decreases. There was a "sweet-spot" indicating the optimal amount of lane changing within a set of density of traffic where lane changing peaked with a critical value of density. If the density fell or rose too high, this optimal amount of lane changing dropped.

Instead of car and vehicle traffic like the other papers, Jia et al. focus on mixed bicycle flow using a multi valued cellular automata [2]. The authors implemented two bicycles with different maximum speeds. The usage of bicycles to compare to vehicles have an emphasis on the behavioral and personality aspect that can be difficult to analyze with cars. The authors explain that young riders tend to ride at high speeds while older riders ride at a lower speed. These differences show that there is no set maxium velocity. In this model, bicycles can move to their next open site with faster bicycles moving with priority over slower bicycles. It was shown similarily to vehicles that slow bicycles that congregate and occupy sites, block faster moving bicycles that cannot overtake and thus move in a platoon order. Increasing slow bicycle density will congest and bottleneck the simulation. One of the points that Jia et al. mention that cannot be properly analysed is the nature of riders that carpool, thus this simulation does have some flaws. Increasing randomization of the locations and density of slow riders tended to allow more free flow and less bottlenecking.

Simon and Gutowitz have built upon the paper referenced above by Nagel and Schreckenberg. [6]. Here they are implementing a bidrectional traffic using a two lane road with traffic moving in opposite direction in comparison to the original one lane cellular automaton. They have incorporated inteteractions to simulate passing, as well as a distribution of varying vehicle speeds. Once again, Simon and gutowitz, like their reference model, had an emphasis on approximating the behavior of real traffic and human behavior. They wanted to address the issue of a one lane cellular atomaton where all vehicles have a maximum velocity and thus the model unrealistically follows a lead slow car, hence the need and emphasis on passing. The bidirectional model varried in types where there was varrying rules regarding passing. The researchers have found that passing greatly increases fluidity in traffic not seen in single dimensional cellular automaton and greatly resembles real world traffic dynamics and flow.

Hafstein uses cellular automata to visualize and simulate real word traffic conditions using inputs that are detected from the North Rhine-Westphalia area in Germany [1]. Loop detectors are inductive electrical devices that are put into the road which can record vehicles that are either passing or sitting upon them. By utilizing these detectors, Hafstein and others were able to determine the vehicle type, such as truck or car, the average velocity of these vehicles, and the amount that passed the loop detector within a time window. Once this data is collected, the data is input into a high-resolution cellular automaton traffic simulator. Hafstein's goal was to improve Nagel's simulation design with such improvements such as whether a vehicle should brake or not depending on the distance to the vehicle in the front. By using a smaller cell and higher density, it allowed more realistic acceleration and speed bins compared to using larger cells of that in other simulations. Lastly, regarding the output of the simulation, a graphical 3D interface was designed to allow city planners to visualize and plan construction around the highway.

3 Methods

For this project, we are using Python to implement our cellular automata traffic flow model. We also included a simple GUI to visualize the traffic flow on each time step. Some of the libaries that we are using include: Seaborn, Pillow, and Numpy.

4 Progress

As of now at this middle update, we have successfully implemented the original algorithm. Our matrix road network is different from the original paper. Instead of an outside loop, with several intersections, our implementation so far is a simple three vertical paths, with one horizontal path (Fig. 1). With this we have generated cells of varying capacity and density which traverse stepping through possible moves through the matrix to reach their randomized terminal destination where from here the cells are removed from system. We can expand the matrix to any size with any number of starting instances of traffic flows, however, so far, we have worked with a 10 x 10 matrix for simplicity.

On top of this, we have also implemented our first step in branching from the original paper. Here we have developed a weighting system that adds weights to cells. By introducing weights to the cells, we able to coerce the traffic cells to take different routes other than the shortest possible path. We are able to see if the route of taking a lower weighted route such as a longer highway with a high traffic capacity would be quicker than a shorter path that allows less traffic flow with a smaller capacity value. Since the weight is simply the inverse of the capacity of the cell we can find an optimal path to the destination (Fig. 2). On top of this, we have designed a simple

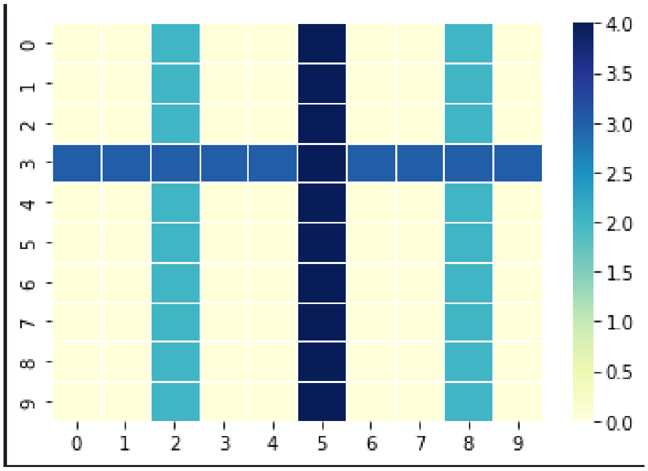

Figure 1. An example traffic grid.

gif to illustrate the traffic flows in addition to an output that shows corresponding coordinates that show past and future moves to allow us to better understand the direction and flow of these cells. This GUI heatmap allowed us to see an intentional design choice in the original paper that is further discussed in the results section.

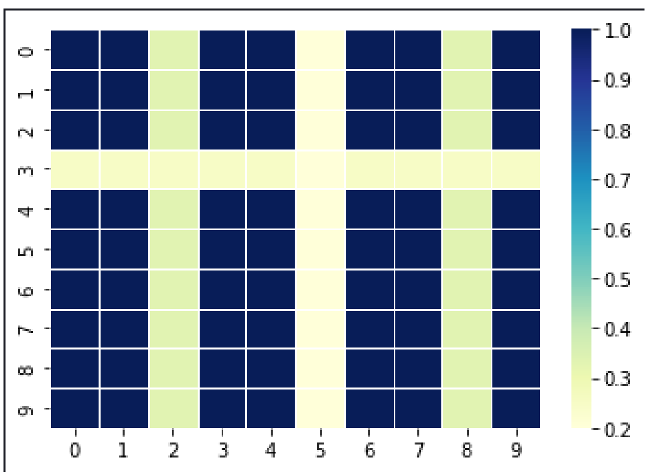

Figure 2. An example traffic grid, showing the weighted matrix.

Regarding the destination priorities that was included in our initial project proposal as a brainstorming idea to branch from the original paper, we have decided to omit the development of it. We had decided that it would not create much of an impact in the simulation in terms of priority along the development timeline. Addressing the weights and their respective functions came as a priority in addition to the issue found with cells getting stuck without a circular outside edge path of the matrix.

Some of the issues that we have run into is the use of technology such as Python, IDE (VS Code) and Github with several of the group members. This presents a challenge as there is a disparity in some of these skills between members. However, despite this, our members are continuously eager to learn and teach one another. In addition to this, members contribute to their strengths while others cover their weaknesses. With this, there has been a constant amount of communication from each member and harmonious work ethic that each member brings to the table to accomplish the task at hand. Through this mutual cooperation, we have been able to steadily progress through our milestones.

5 Preliminary Results and Analyses

From our implementation of the original algorithm, we are able to visualize the flow of traffic. Depending on the number of instances of traffic cells generated, we can see that the cells are able to move to their destination within the shortest path given the traffic capacity is accommodating. It is interesting to see how a cell can quickly reach a high density of traffic once multiple flows of traffic converge at that matrix coordinate. In addition to this, it seems that in the original paper [5], an outer lying circular route at the edge of the matrix was included to accommodate to the rules of the traffic flow. Without the outer route, it would be more cost effective for a cell to stay in its position than to backtrack into another route. By implementing a circular route along the edges of the matrix, the cells are able to continuously move in their desired direction to reach the final destination. This approach to the trafic simulation has its own shortcomings in that not every traffic environment will have this type of roadwork. By implementing our route pathing further discussed in the future works section, we are able overcome this issue with higher traffic simulation fidelity.

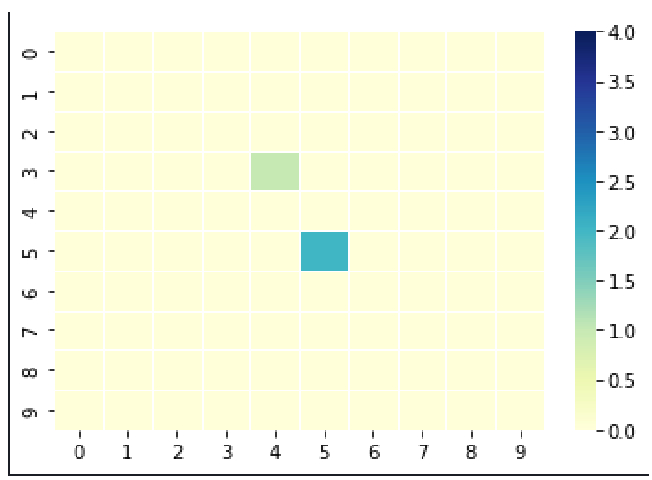

Figure 3. An example of two traffic cells inside the traffic matrix.

6 Conclusion and Future Work

We would like to implement a cost-effective route decision that takes accuracy as the highest priority to prevent the likely hood of a cell staying in single position that was discussed in the results section. By implementing this design, we would be able to have cells reach their destination no matter the cost of the route, without the use of circular route along the edges of the matrix. This was a decision we had come up with that presented itself as an issue not foreseen in the initial project proposal. In addition to this, we are still projected to model different traffic environments such as one-way streets and further do a deeper analysis on the traffic simulation model. Most likely an improvement to the matrix to represent a real-world model would greatly reflect the accuracy of the simulation. A local road or area where we could compare it with traffic from another source, we be a great test comparison. Lastly, with time permitting we would like to further polish our GUI to better visualize the traffic flow simulation than already accomplished.

References

[1] Sigurur F. Hafstein, Roland Chrobok, Andreas Pottmeier, Michael Schreckenberg, and Florian C. Mazur. 2004. A High-Resolution Cellular Automata Traffic Simulation Model with Application in a Freeway Traffic Information System. *Computer-Aided Civil and Infrastructure Engineering* 19, 5 (Sept. 2004), 338–350. https://doi.org/10.1111/j.1467-8667.2004.00361.x

[2] B. Jia, X.-G. Li, R. Jiang, and Z.-Y. Gao. 2007. Multi-value cellular automata model for mixed bicycle flow. *The European Physical Journal B* 56, 3 (April 2007), 247–252. https://doi.org/10.1140/epjb/e2007-00116-5

[3] T Nagatani. 1993. Self-organization and phase transition in traffic-flow model of a two-lane roadway. *Journal of Physics A: Mathematical and General* 26, 17 (Sept. 1993), L781–L787. https://doi.org/10.1088/0305-4470/26/17/005

[4] Kai Nagel and Michael Schreckenberg. 1992. A cellular automaton model for freeway traffic. *Journal de Physique I* 2, 12 (Dec. 1992), 2221–2229. https://doi.org/10.1051/jp1:1992277

[5] Hali Pang and Xu Yang. 2019. Simulation of Urban Macro-Traffic Flow Based on Cellular Automata. In *2019 Chinese Control And Decision Conference (CCDC)*. 520–524. https://doi.org/10.1109/CCDC.2019.8832372 ISSN: 1948-9447.

[6] P. M. Simon and H. A. Gutowitz. 1998. Cellular automaton model for bidirectional traffic. *Physical Review E* 57, 2 (Feb. 1998), 2441–2444. https://doi.org/10.1103/PhysRevE.57.2441