Summary

Team 71663

In this paper, we analyze the shape, size, and merging pattern of the toll plaza system and innovate a traffic model which simulates human drivers' interaction with the environment and other cars, models car movement with realistic assumptions, and collects data to predict an optimal toll plaza.

Our main approach is to find a useful discrete simulation to model the toll booth system that can help verify our hypotheses. We advanced one of the mainstream models called the cellular automata model, which assumes a discrete representation of tollbooths, cars, and road. Our model takes into account the number of tolls, number of roads, maximum velocity in the toll plaza, initial velocity of cars leaving a tollbooth, and the shape of the toll plaza as additional parameters to model more realistic conditions.

We focus our model in heavy traffic because the problem is much more interesting in this respect and we take into account autonomous vehicles as ideal cells in a modified cellular automata model. By simulating different toll booths, with various random variables, we are able to simulate non-uniform behavior of cars at the toll plaza in order to understand the the effects of waves of traffic that is inhomogeneous.

From our model, we found that separating tolls into two main sections at different starting points would result in the optimal configuration of the toll plaza in terms of efficiency and accident prevention. This type of toll plaza configuration promotes an idea of using multiple toll booths at different starting points rather than the conventional toll booths which usually start at the same place.

Letter

Dear New Jersey Turnpike Authority,

We tried simulating cars movement based on data pulled from New Jersey Turnpike's toll and traffic data and models of Thailand's toll plaza. From our simulation, we believe that the shape and structure of Thailand's toll plaza is adaptable to the U.S. toll system. In other words, the model for representing very high traffic in Thailand is a perfect replica for the U.S. New Jersey Turnpike.

Based on the data from Numbeo [1], Thailand has a high relative traffic index compared to the United States, which indicates that people in Thailand spend more time transitioning from one place to another. Many people in Thailand, as a result, decide to use the tolls in order to avoid traffic jams. This suggests that the toll model in Thailand needs to be able to support the high volume of traffic flowing into Thailand toll system.

Describing the model for traffic flow, we notice that the model in Thailand divides the highway into two main sections in which 80% of cars go through the main toll plaza while 20% of the cars skip the toll plaza to go to a toll plaza further in. This way, we can use the same amount of space in a road, but we delay the actual time needed to get into the toll booth. The desired recommendation is pictured below

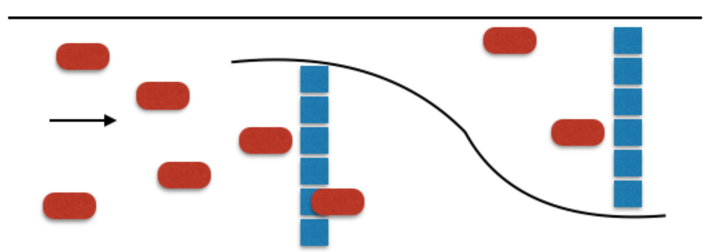

Figure 1: Thailand's Toll Booth Model

As such, the model allows for the optimal path by restricting the set of cars that enter through the toll plaza at a given time. By separating the proportion of cars that go through the two different toll plazas, we employ our simulation that divides the total number of cars into separate queues. The restriction of the total number of cars takes into account the ideas of traffic flow and considers accident prevention, throughput, and cost to help produce a more efficient toll booth system. Therefore, our recommendation is to use a system as shown above. Please refer to the model in our paper to understand more about the intricacies at hand.

Sincerely,
Team 71663

Toll-Booth Performance Improvements

MCM Contest Question B

Team # 71663

January 23, 2017

Contents

1 Statement of Problem

We are asked to consider the shape, size, and merging patterns of vehicles passing through toll booths during light and heavy traffic. By creating a discrete simulation of the toll booth system, we design an algorithm for each car to determine its optimal path to the exit. We use a large rectangular grid to find the locations that are the least frequently visited in each car's optimal path. From prioritizing the merging pattern, we find a useful shape and size for an optimal fanning pattern.

For self-driving cars, we imitate their mechanism and design a different model to find optimal paths, which summarize the most significant difference between human drivers and AIs. We also take into consideration the different types of toll booths by studying the difference in the speed, appearing probability, and other features of cars at different toll booths.

We essentially use cellular automata with a more meticulous partitioning scheme compared to the standard model [6].

Note, we are considering only the latter half of the toll system to determine the better method of merging lanes. From considering the latter half of the toll system, we also make sure to consider collision methods for accident prevention in which we employ an acceleration/deceleration model.

2 History and Context of the Problem

Toll booths exist to fund road maintenance and construction. Although they help recover the cost of constructing roads, they have also caused problems for the general public in the form of traffic congestion. An interesting approach to solve the congestion is to use a better tolling system. For example, there are three main types of toll booths to consider:

1. **Human Tolls**, where human officers obtain our money and return change

2. **Automated Change tolls**, where we deposit money into a machine

3. **Transponder Tolls**, where there is a device that allows cars to drive through the tolls without stopping, albeit at slower speeds

In our model, we can consider these toll booths by assigning different velocities to cars from different exit tolls.

Our task, however, takes the toll booth problem one step further by expanding on the idea of the shape, size, and merging patterns involved, so we can find a better solution to improve flow at a cost-reduced price.

One of our initial thoughts to solve heavy traffic congestion is to increase the cost dramatically of toll booths, which would directly reduce the number of cars willing to go through the toll booths. This method of improving traffic flow, however, will probably elicit a lot of angry complaints from the drivers on the road.

3 Definitions and Conventions

Here, we define some useful definition and conventions that we later use in describing our model.

- the number of tolls, B
- the number of roads, L
- starting velocity, V_{start}
- id of car, *id*
- maximum velocity, V_{Max}
- acceleration rate, A
- angle, θ
- collision ratio, col_{ratio}
- boolean for collision likelihood, *mode*
- distance between toll and exit, d_{BL}
- relative x-position from the leftmost toll and leftmost exit, d_x
- traffic index, t_{index}
- resolution, *res*

4 Assumptions and Justifications

Some statements below are not assumptions.

- Average width of a car 1.8 m – We use this for our average model of a car.
- Average length of a car 4.5m – We use this for our average model of a car.
- Average width of a lane 3.6 m – We use this for the lanes based on the typical lane width.
- Discrete time steps – We use this to model how the cars move with time.
- Length of Toll plaza merge lanes – We assume that the merge lanes are long enough such that a car can complete a complete merging to all the exit lanes.
- At each toll station, we assume that there is only one car that moves through at a time.
- The drivers want to move as fast as possible within the speed limits.
- All the cars in our model will never drive above the maximum speed.

5 Modeling Traffic in Toll Plaza

5.1 Overview

A toll plaza is a complex system which is difficult to model. However, we can divide the whole modeling into several smaller tasks to gradually build our model. Our model building process consists of the following steps.

- Represent toll booths, road, exits and cars with grids
- Simulate human drivers' decisions and behaviors
- Model cars' movement based on last step

- Collect information about the optimal design of the toll plaza

To make our model more precise and realistic, we extend it to a flexible and applicable version. After applying the common cellular model, we adapted it so that it works best for modeling the behavior of toll plaza. The major differences of our proposed model compared to the original model of cellular automata are:

- Cars can be represented as multiple cells, enabling the cars to be tilted.
- Each cell represents the state that can be occupied by multiple cars at the same time indicating the traffic collision or construction site.
- Each car has its own acceleration, deceleration, and parameters that cannot be adopted by the original cellular automata model.

5.2 Original Cellular Automata Model

To model such a complicated traffic situation, we need to refer to the classical models used in history. One popular modeling theory of the traffic model is the cellular automota model. Each car is represented by a single square in the gridding which ignores the difference in cars' size, shape and the continuity of cars' velocity.

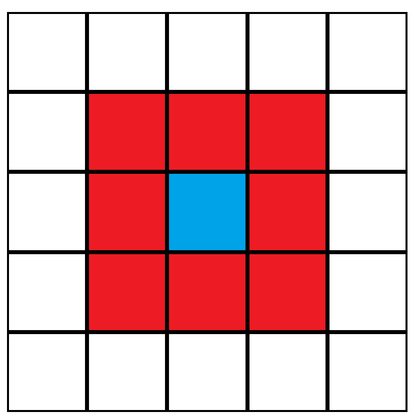

Figure 2: Classical Automaton Model with Moore's neighborhood

Theoretical cellular automata is a discrete model which is pervasively studied in computability theory, mathematics, and physics modeling for modeling a cell on a grid. A grid can have any dimensions, but in the context of traffic flow, we limit the scope to a two-dimensional grid, which can essentially represent the view of the toll plaza from the above. Within the grid, there are cells with a finite number of states. In our case, the cell represents a car, which has different states such as likely to collide, collide for sure, empty, and one car [3].The cells have the property of changing state. Such states are affected by the surrounding neighborhood, which enlists a set of rules for the cell.

Figure 4 shows an example of the Moore neighborhood, which considers its surrounding as its neighborhood [5]. This is one of the main neighborhoods that cellular automata models consider. From observing the nearby neighbors in the surround 9 squares, we create a set of rules that allow our cell to move and adjust its state.

5.3 Adapted Cellular Automata Model

The adapted cellular automata model encapsulates ways of representing cars flowing out from toll booths, merging patterns, and shape of the toll plaza. Figure 3 shows the basic configuration of the adapted cellular automata model. In our proposed model, each car can be represented by connected cells determined by the position of its top-left corner, width, and length of the car. The state of each cell is defined by a function mapping from a cell position to a set that contains the car's id in the cell. The empty cell represents a cell containing no cars, i.e. no cars is currently on the cell of interest. The dead cells represents barriers such as walls, immobile crashed cars.

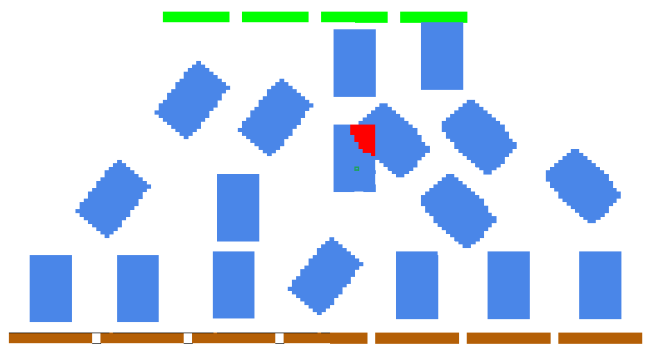

Figure 3: Toll Booth Plaza with $B = 7$, and $L = 4$. The cars are labeled in blue, and the collision are labeled in red.

As shown above, brown bars represent toll booths; blue cells represent moving cars; red cells represent crashed part of two cars; green cells represent the exits. When the resolution, i.e. the number of cells per unit length, is large enough, we can even model tilted cars.

5.4 Human Drivers' Decision Making

Generating Cars from Toll Booths

In the model, cars are generated at the toll booths and are moving towards the assigned exits according to its optimal path. The optimal path is calculated based on the position of the toll the car are in and the position of the car's assigned exit. Then, the car will move along the simplified model as depicted in Figure 5. After the car reaches the exit, the car gets removed from the 2-dimensional grid. In the case of collision, we mark the cell, where the place of collision occurs, and the cells covered by colliding cars, with a colliding state that represents that no cars would be able to go into these cells.

The rules for producing cars from toll booths are based on the random variable, whether there is a car right in front of the toll booth or not, and a parameter determining how heavy traffic it is at the current time of the simulation. The random variable, X, is pulled from Uniform Distribution $X \sim \text{Uniform}(0, 1)$. Obtaining values from the random variable, we compare it with the threshold parameter determining heavy traffic for determining whether a car will be created at a toll booth of interest or not.

After a car is generated, the car will be assigned information about its velocity, acceleration, assigned exit, optimal path, and deceleration rate (in case of preventing collision). Most of the parameters will be randomized parameters varying around their respective average value.

Find the Optimal Path

The optimal path is generated when a car is created at a toll booth. The path is calculated based on the current position of a car and its destination, i.e. an assigned exit. The optimal path is updated accordingly as the car moves in the grid. Figure 5 shows the calculation of the optimal path.

The rules of making an optimal path for human drivers are as following

- Consider all possible exits, positions to begin making a turn, positions to end a turn
- Calculate the number of collisions, i.e. the number of cars lying on the path.
- Calculate the length of the path
- Balance between the number of collisions and the length of the path, choose the optimal path.

Waiting Policy

In our model, we also include the rules for cars to accelerate or decelerate based on the behavior of cars' surroundings. The rules are determined by the direction of the cars, their velocity, and their acceleration. Considering a car at a given time in the simulation, we will accelerate or decelerate based on 4 following rules:

Case 1: there is no cars in its front and rear. In this case, we will accelerate the car towards the exit following the optimal path we have calculated.

Case 2: there is a car in its front. In this case, we will follow the behavior of the car in front. In other words, if the car in its front is accelerating, the car of interest will also be accelerating.

Case 3: there is a car in its rear. In this case, we consider the direction of the car in its rear and our current car. Based on the optimal path, we predict whether these two cars are going to collide or not. In the case of collision, we prioritize the car moving straight, e.g. cars with $\theta = 0$ according to 4. The car moving straight will be accelerating, and the other car will be decelerating. In the case of no collision, we will not accelerate nor decelerate.

Case 4: there are cars in its front and rear. In this case, we follow the behavior of the cars in front according to Case 2.

5.5 Modeling Cars' Movements

As shown below, a car is represented by its upper-left corner's coordinates, angle of its front wheels, angle of its body and velocity.

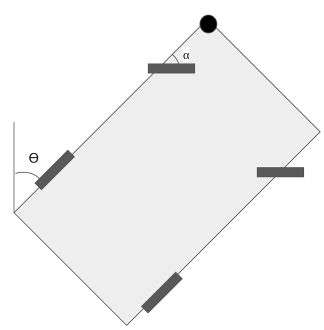

Figure 4: Cellular Automaton. Here is a representation of a single car in our grid.

For a car going to make a turn, we assume that the driver always directs his front wheels to the destination. With the parameters noted below, we can write down several differential equations.

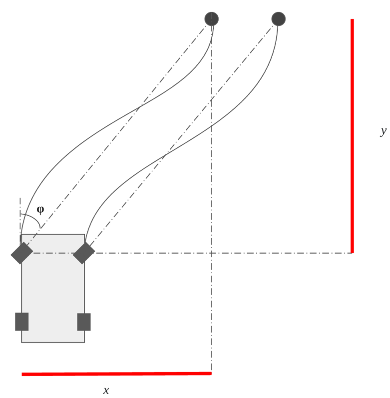

Figure 5: Adapted Driving Movement Model. Here is an example of the optimal path a car needs to take to reach a desired exit.

5.5.1 Differential Equations of Cars' Movements

$$\theta + \alpha = \arctan \frac{y_0}{x_0}$$

$$\dot{x} = v \sin(\alpha + \theta)$$

$$\dot{y} = v \cos(\alpha + \theta)$$

$$\dot{\theta} = \frac{v \cos \alpha}{l}$$

5.5.2 Boundary Conditions

$$\theta(0) = 0$$

$$x(0) = 0$$

$$y(0) = 0$$

$$x(t) = x_0$$

$$y(t) = y_0$$

With the differential equation set above, we can solve for the front wheels' movements.

5.5.3 Movements of the Trailing Wheels

Since the trailing wheels cannot be tilted from the body, its angle with respect to the lanes will always be θ. So the trailing wheels will travel in this direction until the distance between the front wheels and the trailing wheels is met. Hence we also have a way to model the movements of the trailing wheels.

5.6 Data Collection

Given our modeling of the toll plaza traffic, we can simulate cars passing through for plaza of any shape and any size. Below is a screen shot of our simulation for a rectangular plaza.

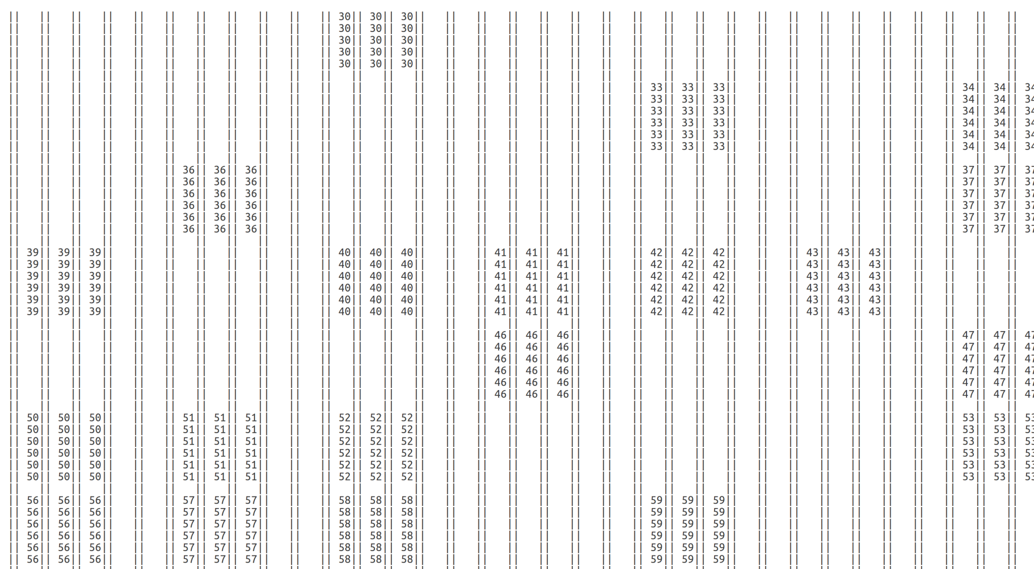

Figure 6: Data Collection

We will run the simulation for the rectangular plaza and calculate each grid's frequency of being occupied. Those grids whose frequency is low enough will be deleted from the rectangular plaza and only the cells useful for the traffic will be left. In this way, we can always find the optimal shape of the plaza.

6 Sensitivity to Parameters

6.1 Light and heavy traffic

From the results, as the traffic becomes heavier, the passing time greatly increases. Our model tries to focus on heavy traffic more carefully than light traffic because we make the assumption that when cars move well in heavy traffic, the worst case for light traffic must be less than or equal to the most optimal case for heavy traffic. Naturally, we conclude that a threshold exists for the great leap in passing time from light and heavy traffic. In our model, we reflect the light and heavy traffic by modifying the traffic index ratio, which is a parameter scaled from 0 to 1. When the parameter is 0, we assume that there is no traffic. When the parameter is 1, then we assume that we are at max flow in which there are cars generated at every single toll plaza at each discrete time step.

6.2 Autonomous vehicles

From reading papers on dynamic bayesian networks, we assume that self-driving cars behave optimally in our system [4]. The self-driving cars fit very closely to the cellular automata model because they focus on their local neighborhood by using sensors to predict movements based on subtle changes as we also see reflected by defining the neighborhood of autonomous vehicles.

As the number of autonomous vehicles increases, the general trend of vehicles at the toll plaza will increasingly behave towards the optimal model. This likely suggests that the increasing number of autonomous vehicles would improve the quality, e.g. speed and safety, at the toll plaza.

6.3 Proportions of tollbooths

The algorithm in our model is based on the ratio given by the paper [2], which suggests 2 manual lanes, 1 automated lane, and 1 fast pass lane. As we varied the parameters in our lanes, we found that this ratio was a decent base for our model. We recommend in the future works to vary this ratio for further optimization to produce a more successful model.

7 Strengths and Weaknesses

7.1 Strengths

- The increase in resolution guarantees limitless precision
- By including an acceleration/deceleration model, we account for the different merging patterns in car movement.
- Our model is extendable, ready for all kinds of improvements.
- We visualize our model to make our result more intuitive.

7.2 Weaknesses

- When simulating human drivers' devision making process, our assumptions are too simple to fully recur the complicated human psychology.
- We assume that the driver can turn the front wheels very quickly, which may not be the case
- We lack consideration for mechanisms of self-driving cars other than following.
- Our simulated front wheels can only drive along straight lines, which is unrealistic.

8 Conclusion and Recommendations

From our experiments, we determine that the most optimal pattern for fanning in is satisfied when the exits are placed in the middle. The exact shape of the toll plaza will be determined by our real-time simulation, since no analytic result can be provided.

Self-driving cars will affect the traffic due to their different mechanisms of driving - they will follow other cars instead of determining an optimal paths for themselves. But they will not greatly affect our results since their behaviors are still similar to human drivers. Different type of toll booths will affect our result in that they will affect the initial velocity. The electronic booths will ensure the highest initial speed, resulting in the shortest passing time for the cars.

9 Future Improvements

- Increase the cases for the cars to change velocity, making our simulation more realistic
- Take self-driving cars into consideration by adding the ”following” mode
- Take different types of the toll booths into consideration by setting different initial values for cars out of different toll booths.
- Run more simulations to increase the size of our data collection
- Calculate the safety risk for different shapes of the toll booths

References

[1] Traffic. URL `https://www.numbeo.com/traffic/indices_explained.jsp`.

[2] Dan Gulotta Andrew Spann, Daniel Kane. Lane changes and close following: Troublesome tollbooth traffic. *The UMAP Journal*, 26(3):251–264.

[3] B Chopard and M Droz. *Cellular automata*. Springer, 1998.

[4] Tobias Gindele, Sebastian Brechtel, and Rüdiger Dillmann. A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1625–1631. IEEE, 2010.

[5] Lemont B Kier, Cheng Chao Kun, and Paul G Seybold. *Modeling Chemical Systems Using Cellular Automata: A textbook and laboratory manual*. Springer, Dordrecht, 2005. URL `https://cds.cern.ch/record/997542`.

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