Discrete Event Simulations of Traffic Flow

CSE 6730 - Modeling and Simulations

Bryan Crumpton
bcrumpton3@gatech.edu

Manikanta Reddy Dornala
mani@gatech.edu

Woradorn Kamolpornwijit
wk@gatech.edu

ABSTRACT

The goal of this project is to model vehicles driving along Peachtree Street. The model is only taking into account traffic in one direction, and spanning from the 10th street intersection to the 14th street intersection. We will then analyze the average travel time for a vehicle to go from the origination zone to the destination zone in the area of analysis (show in Figure 1). As with all models, some simplifications and assumptions will be made, to be described in Section 4-6, under the respective simulation type. Simulations in this area are extremely useful for their potential to impact multitudes of lives, based on the insight provided from the simulation. Additional lights can be tested, turn lanes added, and even adjustments of traffic light timing can be tested before real world implementation. In addition, new scenarios can be tested, like entirely motorcycle or truck traffic, how flow of traffic can affect travel time and how peak times can affect overall travel times.

1 INPUT ANALYSIS

There are many variables available for tuning within the problem. These can be broken down into the following areas:

- Traffic
- Traffic Lights
- Vehicle Parameters

These parameters have been explored in order to develop a realistic simulator, and can always be expanded upon further in future expansions to the project. Note that all exploration discussed below involved partitioning the NGSIM data set received into two groups, one to analyze and a smaller section in order to compare our simulators to some real life examples. All data analyzed referred only to traffic moving in our chosen direction of traffic, Northbound.

1.1 Traffic

One of the largest inputs dealing with traffic alone is the inter-arrival time for vehicles. The team has analyzed the input data set partition, and found that the arrival times closely match a Poisson distribution, as seen in Figure 4. This data will be further discussed in the following subsection. The inter arrival time was utilized during the generation of vehicles and the logic to determine when another vehicle should be spawned. Another input available for tweaking with traffic are the Origin and Destination Zones. There is currently an implementation of an empirical generator, based on our aforementioned grouping of test data.

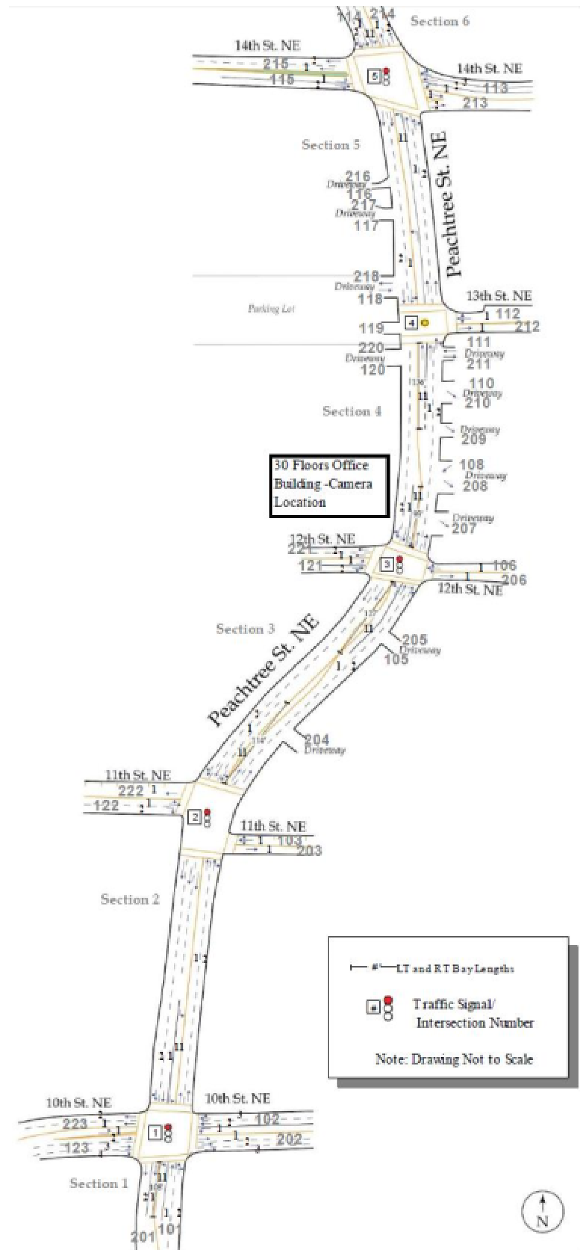
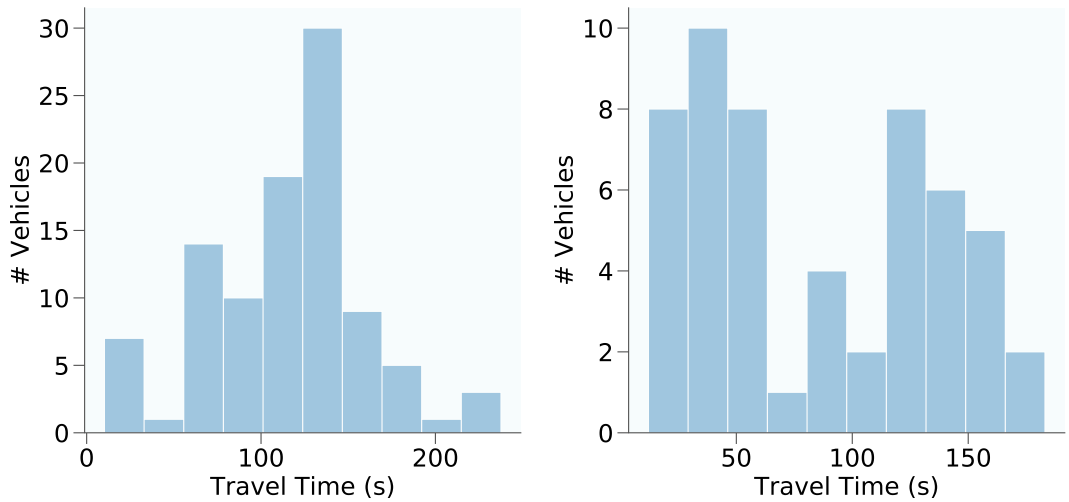
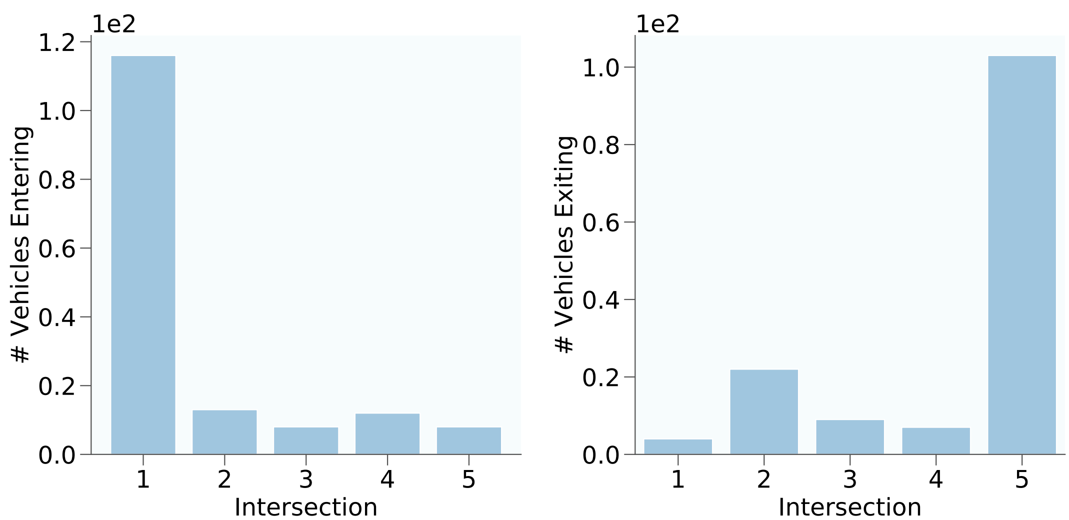


Figure 1: Depiction of the Peachtree Street Corridor, from 10th Street to 14th Street



(a) Train Split **(b) Test Split**

Figure 2: Travel Time Distribution



(a) Distribution of Vehicles Entering **(b) Distribution of Vehicles Entering**

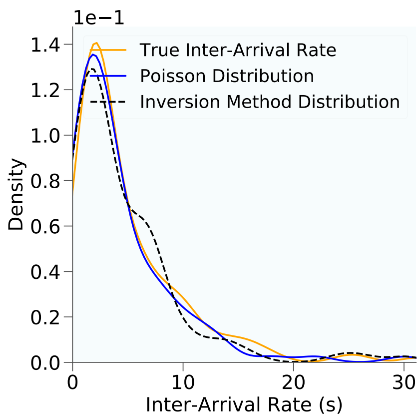
Figure 3: Input data Distribution

1.1.1 Inter-Arrival Times

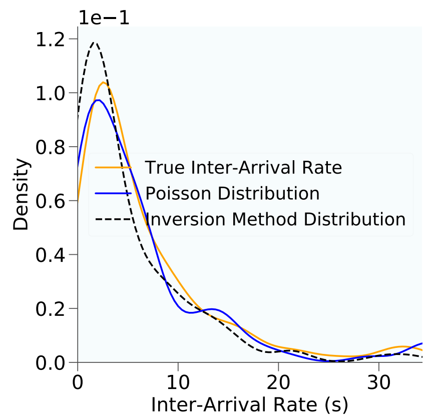
Each individual graph in Figure 4 shows how the real distribution from the 75% data split compares to a Poisson Distribution with the same mean as the true distribution. The Inversion method distribution is generated from the inverse IDF curve of the True Inter-Arrival Rate of the training data split. This is also utilized for the implementation of vehicle inter arrival times, providing a different set of results that can be compared and contrasted.

1.1.2 Vehicle Origin and Destination Zones

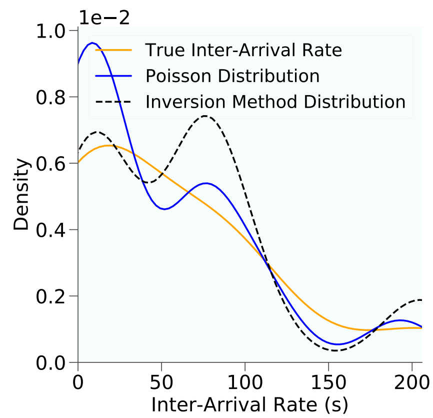
We can see from Figure 3a and Figure 3b that the majority of vehicles enter at Intersection 1 and exit at Intersection 5. To ease in the development of the simulation while still maintaining a realistic level of traffic, some implementations of our simulators only generate traffic going directly from one end of the corridor to the other. In addition, these simulators disregard lateral traffic. As shown in Sections 4-7, even with these simplifications we are able to achieve a realistic measure of success.



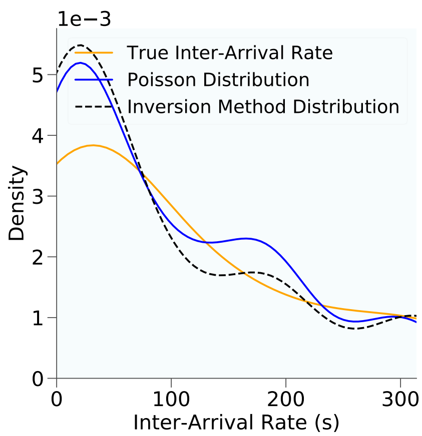
(a) Overall Inter arrival Time



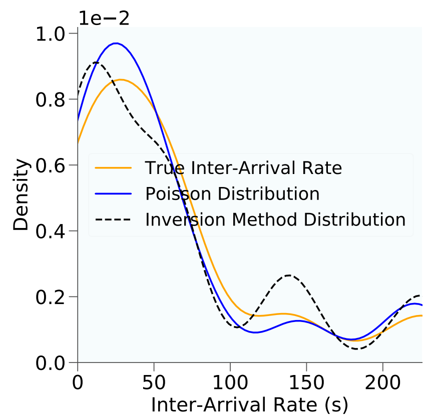
(b) Intersection 1



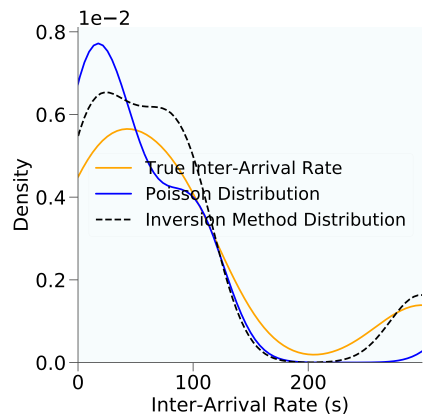
(c) Intersection 2



(d) Intersection 3



(e) Intersection 4



(f) Intersection 5

Figure 4: Inter Arrival Times - Comparing Actual Distribution, Poisson and IDF generated distribution

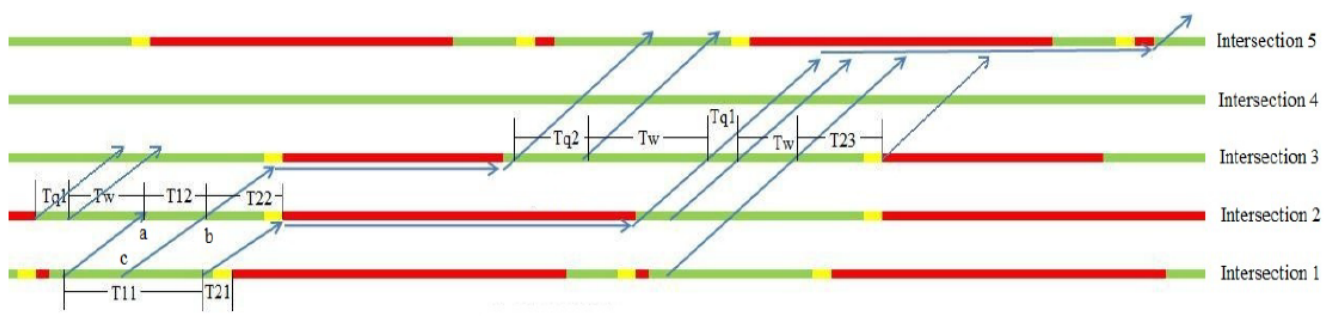


Figure 5: Cycles of Traffic Lights

1.2 Traffic Lights

The main parameter that can be adjusted here is the initial state of the lights. The transition times of the lights are highly detailed in source material, and is implemented in our project. Based on the initial state of the lights, you may see a large variation of vehicle travel times though the Peachtree corridor. Figure 5 depicts the process graphically.

1.3 Vehicle Parameters

There are many vehicle parameters that can be adjusted to complicate the project more. Some are not necessary to achieve realistic levels of performance of our model during simulation.

- Vehicle Length
- Vehicle Acceleration
- Vehicle Reaction time

- Vehicle stop location at intersections
- Safety buffer size

Some of these parameters are examined in the two reports given to the class, and can be utilized as such. Others will be further explored in the future of this project, and will evolve from the currently implemented uniform distributions or static pieces, into more stochastic processes, matched to distributions seen in the NGSIM data-set.

1.4 Overall Input Discussion

Some overarching questions can be raised regarding the data set, and will be under discussion in this section.These questions are as follows:

- Inter-arrival time of every intersection
- Independence Tests
- Whether the data is Stationary or Non-stationary
- Relevance of lateral traffic.

1.4.1 Inter-arrival time of every intersection

As depicted in Figure 4, there exists a vast difference in the arrival times for specific intersections vs the overall arrival time. Previously discussed in the Vehicle Origin and Destination Zones subsection, the majority of vehicles end up spawned at Intersection 1. All of the individual intersections still roughly follow a Poisson distribution.

1.4.2 Independence Tests

An important question should be raised on whether data points are independent, or correlated to one another. Because vehicles that take longer to travel through the corridor will occupy road segments for more time, you can see that the data points are indeed correlated to one another, based on average travel time. If there are simply more vehicles, or even the same amount of vehicles but entering the corridor in closer proximity to one another, they will all begin delaying one another much more and account for overall longer travel times in the corridor. Visual analysis of the scatter plot in figure 6b reveals how number of cars arriving at time $t+1$ is independent of the number of cars arriving at t in our data set. Hence our inter arrival duration is an independent random variable.

1.4.3 Stationary vs Non-Stationary

The NGSIM data set given to us covers 4:00-4:15 pm. Given such a small window of time, it is easy to see how the data is considered stationary. If instead, the data set covered multiple hours, days, or weeks, it is trivial to see how the data would become Non-Stationary. This could be due to factors such as

- Work Schedules
- Meal times
- Holidays
- Popular Events

To have a statistical basis for stationarity we performed the Dickey Fueller test on our time series of vehicle counts. This resulted in a p-value of 0.95 and a test statistic of 0.0, with critical values being -4.6(1%), -3.4 (5%) and -2.8 (10%) showing that the is significant since the critical values are less than the test statistic. [1]

2 CONCEPTUAL MODEL

As all 3 implementations of models take various liberties between assumptions, the specific assumptions key to each model will be discussed separately in their own sections. All the simulation model only one direction of traffic along the corridor.

3 ACTIVITY SCANNING

The activity scanning model is very similar to event driven simulation. All entities are now viewed to be taking part in activities, which are still transitioned between using events. A uniform time step is utilized between all simulation tics, currently set at one second.

3.1 Assumptions

There are a few assumptions utilized in the implementation of the activity scanning model. Things like predetermined vehicle length, instantaneous acceleration/deceleration and simplified driver behaviors allow for an easily implemented model while still resulting in realistic outputs. In addition, inter arrival time are currently set to be drawn from a Poisson distribution, as discussed in Input Analysis section. Vehicles are currently able to be spawned from any potential Origin Zone, with any potential northbound Destination Zone also being able to be spawned.

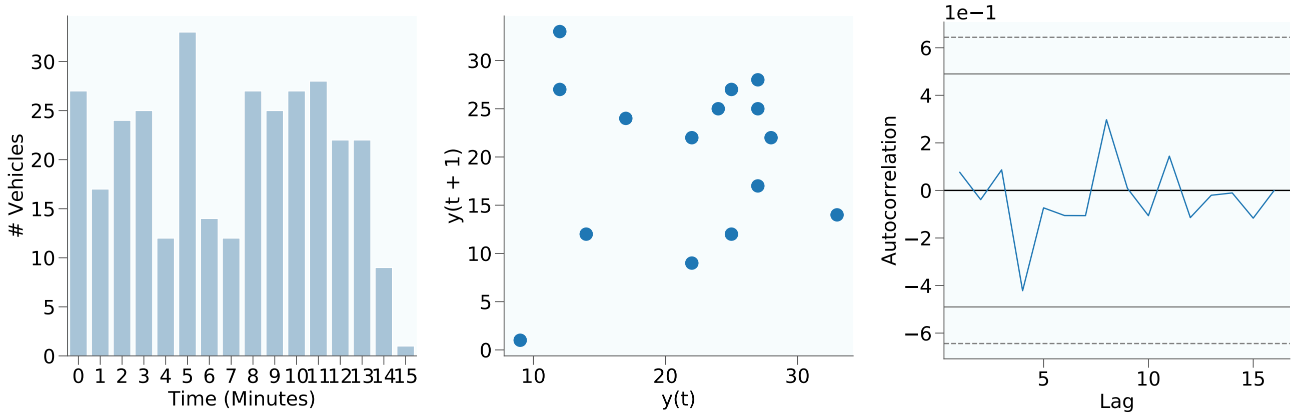
3.2 Implementation

The activity scanning model is implemented in the inefficient, non priority queue methodology. Instead, a program loop must iterate over a list of activities multiple times per simulation tick in order to ensure all potential activities are executed at a moment in time. Although Activity Scanning models allow for B and C type activities, only C type activities currently exist in the implemented model as all potential activities have a predicate requirement (for example, a road space being unoccupied to move). A data structure of potential road segments is utilized to check the availability for movement of all vehicles. To contain all C type activities, there exists a typical python list that gets added to and removed from during each time execution.

3.3 Simulator Applicability and Limitations

This simulator has tried to be developed in a modular method that allows for easy adaptation of many scenarios. One shortcoming of the method of implementation is with set vehicle length and large road segment lengths. This makes the simulator unable to cope currently with varying travel speeds, vehicle lengths and inefficiently uses the available road space. As a future expansion, the road could instead be split up into 1 foot segments, allowing for more efficient layout of vehicles

[1]https://en.wikipedia.org/wiki/Dickey-Fuller_test



(a) # Vehicles passing through **(b) Time Independence Test** **(c) Independence Test with varying lag**

Figure 6: Input Analysis

and adaptable velocity. This would in turn allow the adaption of the simulator into a system of simulators, allowing for models covering driver aggression, stopping distance and following distance.

4 CELLULAR AUTOMATA

4.1 General model description

The model implemented in this section is a cellular automata model of traffic flow. Each cells in the cellular automata represents a vehicle, while the grid represents the street. Each cell in the grid can either be empty or occupied by a single vehicle. In each timestep, each vehicles observe the state of the street in front of it and react by either staying still or moving forward.

4.2 Assumptions

The additional assumptions implemented into the cellular automata model are as follows:

- All vehicles in the same run are the same length (1 block) of either 8, 16, or 32 feet long.
- Vehicle speed is either stopped, 16ft/s, or 32 ft/s.
- All drivers will try to maintain at least 1 block of gap to the preceding vehicle while moving.
- All vehicles exits the system at 214. In other words, no vehicle turns off of the street.
- No lane changes occurs.
- Vehicle inter-arrival time is assumed to be a Poisson distribution with probability such that the average rate of vehicle spawning matches the rate found in the data set.

4.3 Modifications

The first modification is that instead of storing whether each cells contains or does not contain a vehicle, each cells actually contains a reference to a specific vehicle instance. This allows more detailed tracking of the vehicle to be implemented. The vehicle are tracked in terms of its location, time it enters the road system, and velocity. This information is used to calculate the amount of time the vehicle spent in the road system before exiting. This allows the system to avoid processing the empty cells and improve performance while remaining mathematically identical to direct cellular automata simulation.

The second modification is that another grid of traffic lights are also tracked. A red light behaves like an occupied cell, whereas a green light behaves like an empty spot. This allows the vehicle to react to the state of the traffic light appropriately.

4.4 Simulator limitations

The assumptions used in the construction of the simulator causes the following limitations:

- The system can't simulate traffic with mixed vehicle length.
- The system assume all vehicle accelerate at the same constant rate.
- The system does not allow vehicle to turn off of the street.
- The system does not allow for lane changes. This may cause traffic imbalance between lanes.

5 EVENT ORIENTED QUEUING

In this model of traffic simulation, movement of traffic is modeled as discrete events being triggered at different times in the forward march of time.

5.1 Assumptions

For the purposes of this simulator built certain assumptions have been made.

- All vehicles are the same.
- They move with a constant speed and accelerate or decelerate instantly.
- The traffic lights between the sections have only two signals (red or green). This decision was made because

the duration of yellow light was very low compared to green or red signals as per input data. Figure 5.

- Vehicles can only go forward, although lights for right and left turn have been implemented. This is a justified assumption based on Figure 3. We have previously seen that the number of vehicles entering or exiting through right or left sections is very less as compared to the forward traffic. Hence a decision has been made to exclude the lateral traffic.
- Sections of roads have a fixed travel time. This is based on the actual lengths of the sections from map data and a fixed constant velocity of the vehicles estimated as the average from input data.
- Sections of roads have a fixed capacity (maximum number of vehicles that can be present in that section at any moment in time). This is decided by the lengths of the sections and length of the vehicles.

5.2 Components

The event driven model has two components, the application layer and the executive layer.

The executive layer is the core engine, that is agnostic of our application layer and can be used for any other simulation. It manages the future event list as a priority queue and schedules events as per their time stamps. It also maintains the clock of the system. The engine also runs the simulation by executing events by picking them from future event lists in the order of their priority (timestamp). Whenever an event is removed from the FEL, an event handler provided by the application layer is called to execute simulation logic. These events can be of different types and each may have a different event handler associated with them.

The application layer consists of all the logic regarding event creations and entity relationships.

5.3 Entities

We have three entity types in the simulation 1) Car, 2) Traffic Light and 3) Road section. There are multiple instance of each entity type in the simulation.

For each road section there are different attributes associated with it. 1) Travel Time 2) Capacity 3) Current number of vehicles. The capacity of section 6, the final section, is set to infinity and a car entering this section is considered to be exiting the simulation. All cars begin their journey in section 1.

There are traffic lights in the system between each of the consecutive sections. The traffic lights at each of the intersections have been modeled as various state variables that are functions of time as opposed to be ones managed by the simulator engine. In this approach traffic lights are deterministic and we know exactly what the signal will be at any given moment in time, hence are functions of time. In the other method, traffic signals changes are modeled as events that are scheduled by the engine and are changed by the handler function when executed. The behavior of both is equivalent. Both were considered and this simulator implements the first approach.

Cars are generic entities that do not have any defining attributes of their own at the moment except for their id, the section they are currently in, the section they first entered the simulation and the section they exited the simulation and corresponding time stamps.

5.4 Implementation

One defining parameter of the simulation is the *inter-arrival* δ time between cars reaching the first intersection. However this interval is stochastic and is randomly generated. This is modeled by creating an event scheduled at current time with a handler for handling arrival of the car the first intersection. Then another event is scheduled at a future time that is δ ahead of current time for new arrival event. This is repeated for every new arrival. Thus cars keep arriving at first intersection separated by a random time δ. This simulator utilizes the Poisson distribution at every intersection as was discussed in Figure 4a.

Irrespective of which intersection it is, when an event for car arriving at the intersection is executed it checks if the current light is green and if the section ahead is not full, then the car moves from the current section to the next one. Then a new event is scheduled in future for this car for arriving at the next intersection after τ time, equivalent to the travel time of the section as discussed previously. If the light is red, then the car cannot move forward and will have to wait till the light turns green, hence a new event is scheduled in to the future for this car when the light turns green. If the light is green and the section ahead is full then this car will have to wait till someone moves from the section ahead, this happens when the light in the section ahead this turns to green and a corresponding event is scheduled in future.

Section 6 has infinite capacity and cars cross over when light turns green at the intersection. When a car enters section 6 it exits the simulation and no future events are scheduled for this car.

We have defined the warmup time of the simulation as the time till when the first exit happens. This ensures that all sections of the roads have some vehicles in them.

The entire simulation has a fixed end time and when the clock reaches this time the simulation ends. This is a user defined parameter.

5.5 Limitations

This simulator doesn't account for stochastic nature of vehicles and physical phenomenon like appreciable acceleration and deceleration. Effect of different kinds of vehicles (like cars, trucks, cycles) in a real world is very different. This simulator cannot take these into consideration as it assumes they are all the same. Other random processes like vehicle breakdowns, distracted driving are not accounted in these simulations and may have a pronounced effect.

6 VERIFICATION

Verification of the simulator is simply assumed to be the absence of infinite loops and crashes in the code. The simulator also match expectations set out in this section of this work. It seeks to match simulator results to the subsection of the NGSIM data set discussed in Section 1 within a margin of error. Some of the general validation methodology included the creation of unit tests for each simulation and utility.

6.1 Activity Scanning

An example tester can be seen in the GenerationTester.py, which simply generates a number of vehicles where you can validate the distributions of origin/destination zone, vehicle type and could be expanded to further test additional metrics if more were added to the vehicle.

6.2 Cellular Automata

The verification of the cellular automata model was performed using both manual inspection of the result and unit testing. Due to the stochastic nature of the model, automated testing is impractical to implement. For this reason, the verification of the code is mainly done by running the simulator and outputting the state of the simulator at each timestep. These output are then manually inspected for errors. Unit testing are also used where practical. This can be found in CA_sim_test.py.

6.3 Event Oriented

The verification of the event oriented model is primarily for the correct functioning of the event scheduler engine. This was done to multiple manual unit tests to ensure that the priority queue is correctly maintained and events are popped in a timely fashion without breaking any laws of casualty.

Other verification tasks included correct event handler callbacks for various kinds of events so that only authorized events can call vehicle generation and departure handling.

7 VALIDATION

The mean travel time over multiple simulations ends up following a nice normal distribution curve, as seen in Figure 9. As seen in Figures **??**, 9, 8 our simulators are in line with each other, and are also realistic when compared to the overall average travel time 2. All of the simulators were able to be compared against one another, as well as against the true data in order to ensure validity.

8 DESIGN OF EXPERIMENTS AND OUTPUT ANALYSIS

The outputs that the team is concerned with are almost entirely all Derived scalar output variables (DSOV). These DSOVs are averaged values from a list of Point set output variables (PSOV). In order to get a PSOV, you must do a single run of the simulator. Our experiments revolved around running multiple iterations of the simulator, and then interpreting the data using 95% confidence intervals. As seen in Figure 9a,Figure 9c and Figure 9b the confidence intervals are very tight and match the potential value calculated in the sample reports given to the team. All experiments were designed for individual simulators to have similar parameters (for example, number of vehicles to spawn and the inter arrival time).

In addition, as a part of our analysis, we have designed and expanded upon individual experiments that are discussed in Section 9.

8.1 Warm up Period

A warm up period is the phenomena of an initial set of readings as being different from the average, due to the initial state of the simulation being empty. In the case of this traffic simulator, the first few vehicles will have low travel times due to the roads being "empty".

8.1.1 Activity scanning

The warm up period generally will depend on the inter-arrival time of vehicles. By utilizing our Poisson distribution, discussed in Section 1, we are able to keep the warm up period to around 20 vehicles for the activity scanning simulation. In order to still get realistic readings, we disregard those 20 initial vehicles and will be sure to spawn extra vehicles to achieve the number of readings we were targeting.

8.1.2 Cellular Automata

the warm-up period of the cellular automata was derived from manual inspection of the vehicle travel time distribution. This data is shown in figure 7. From the graph, the system can be observed to be cyclical in nature. For this reason, the warm-up period of the system is considered to be 1000 cycles. The data is then recorded for 2000 cycles afterwards, giving 2000 seconds of data. Due to the timing logic of the simulator, the simulator also has considerable cooldown time at the end of the simulation. for this reason, the simulation is allowed to run for 500 more cycles before termination. The data is then cropped to the 2000 cycle section in the middle, discarding the warmup and cooldown section of the data.

8.1.3 Event Oriented

We have defined the warmup time of the simulation as the time till when the first exit happens. This ensures that all sections of the roads have some vehicles in them. This translates to about 95s in the simulation time. Once the events before warmup have been fitered out the results of the simulation are fairly stationary and approximate the real dataset very well.

8.2 Simulation Results

8.2.1 Activity Scanning

For an individual simulation run, the travel times can be seen in Figure 8. This can be compared to 2, showing the individual travel times for the real life data collection. For Activity scanning, it can be seen that there is a very binomial distributed looking set of results. This is due in part to the initial settings of the traffic lights. The initial period determines if all lights will be green together, or staggered etc. In addition, there is a large group of vehicles with small travel times. This is due to the vehicles spawned extremely close to the ending/destination zone. After the execution of multiple

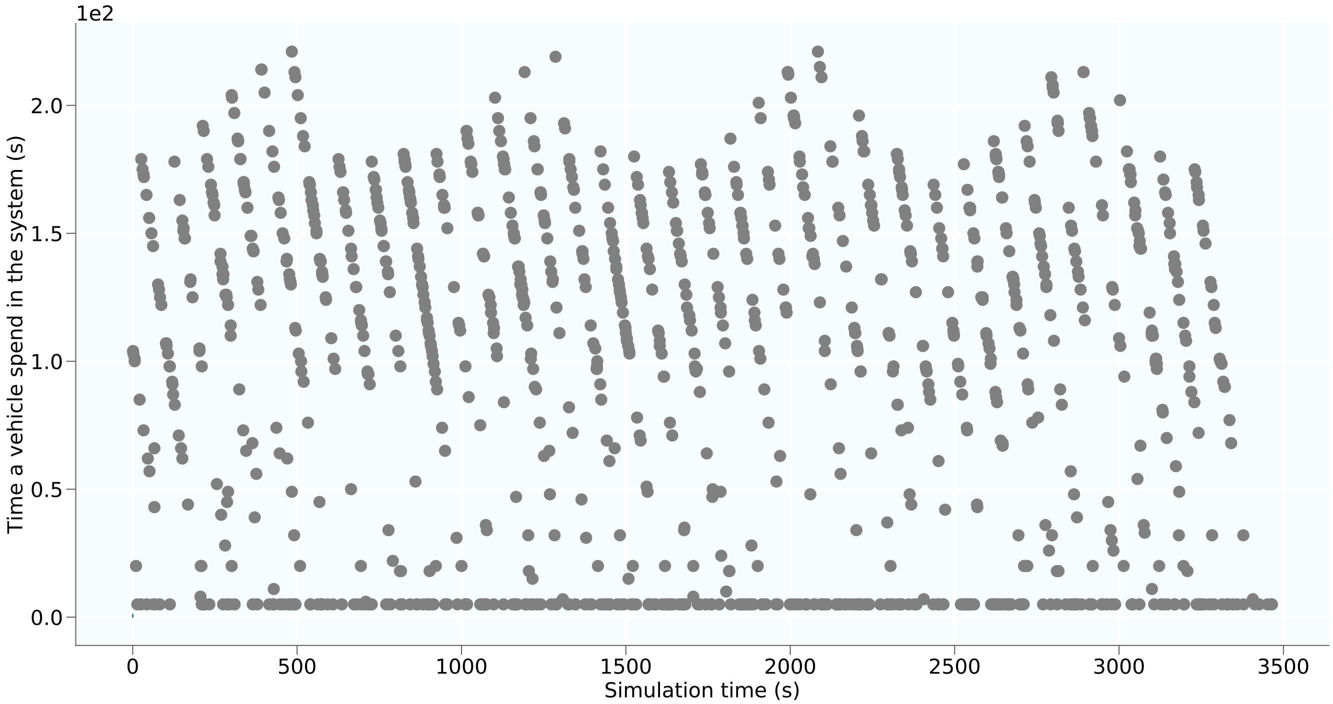
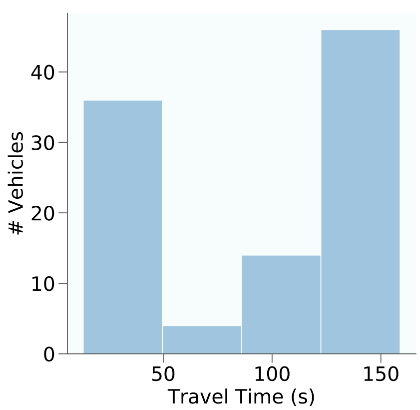
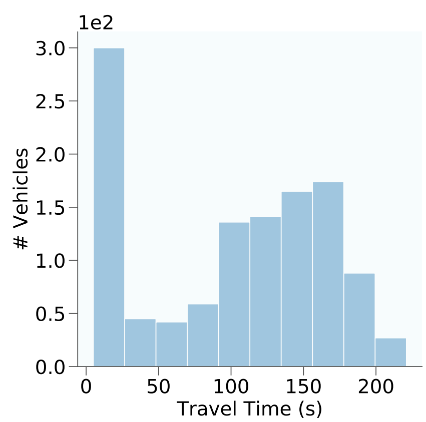


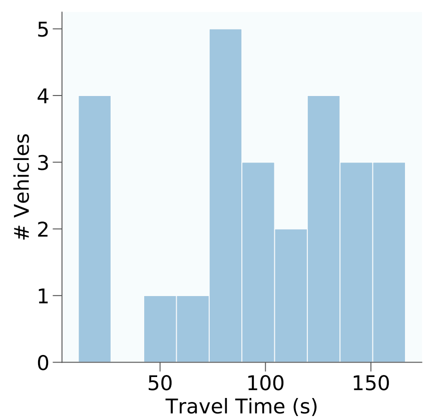
Figure 7: Vehicle travel time as a function of time the vehicle enters the system for Cellular Automata simulation.



(a) Activity Scanning



(b) Cellular Automata



(c) Event Driven

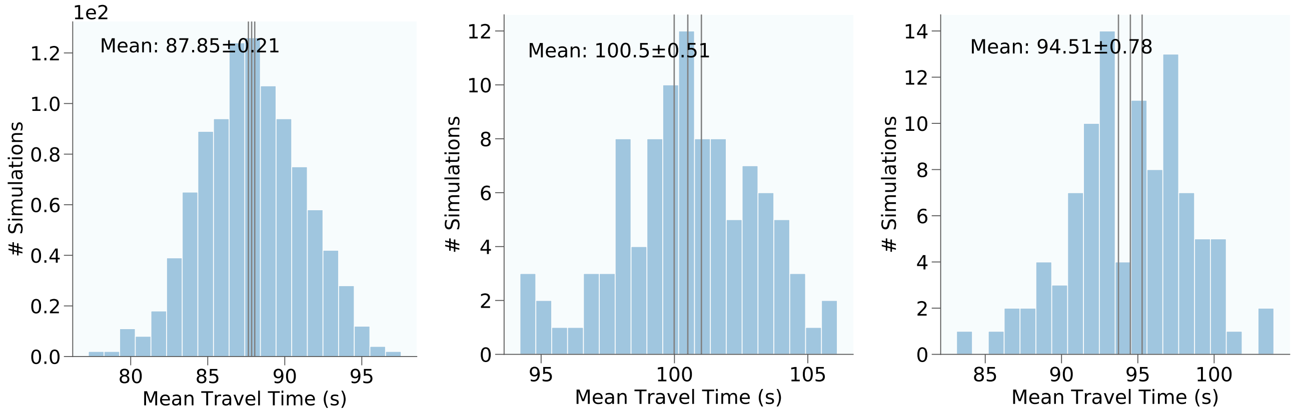
Figure 8: Travel Time distribution in a Single Simulation

simulation runs, the confidence interval for this simulator at 95% was +-0.21, showing just how close and consistent the data was.

8.2.2 Cellular Automata

Figure 8b shows the distribution of travel time for vehicles in one simulation. The peak around 10 seconds could be attributed to vehicles entering the system at the last intersection, after the traffic light. These vehicles can travel to the exit almost instantaneously due to the short distance to the exit. The 90% confidence interval of 0.5 seconds indicates that the choice of warmup and cooldown is appropriate.

Figure 9b shows the mean travel time from 100 runs of cellular automata simulation. From the graph, the simulation can be observed to provide a result which resembles the normal distribution. The slight peak at each ends of the distribution can be attributed to vehicle being spawned at 113, the closest



(a) Activity Scanning **(b) Cellular Automata** **(c) Event Driven**

Figure 9: Mean Travel Time distribution

point to the exit, and vehicle spawned at 101 just before the light turns red.

8.2.3 Event Oriented

Event driven simulation resulted in a mean travel time of 94.51 seconds with a 95% confidence interval laying well within 0.78 (Figure 9 when the simulator has been run for over a 100 executions. This means that the simulator is fairly confident about the result of its execution and the reliability of the simulator is very high. The distribution as can be seen is a very close to normal distribution implying that the results are mean centered and the expectation of the simulation result will lay close to the mean.

When we look at the distribution of travel times within a single simulation, Figure 8c, we notice that there is strong peak on the lower travel times and a spread in the higher sides. The lower times are due to the vehicles being generated in the pen-ultimate intersection and quickly passing through into the exit section. However this also mather the original distribution of the test set(Figure 2b), suggesting that our simulators generalize the model although they only used the the inter arrival time distributions from the train set.

8.2.4 Comparing the Simulators

As we can see in Figure 10 our different simulators when run for 100 times generate mean travel time fairly consistently. It falls well within the travel time as was captured from the test set. This uniformity in the travel times across simulations suggests the simulators are modeling what they want to model reasonably (hence are also validated). Their equivalent slight upwards deviation from the mean of test set suggests that their assumptions are correlated and their simplifications are logical, hence they model similar phenomenon.

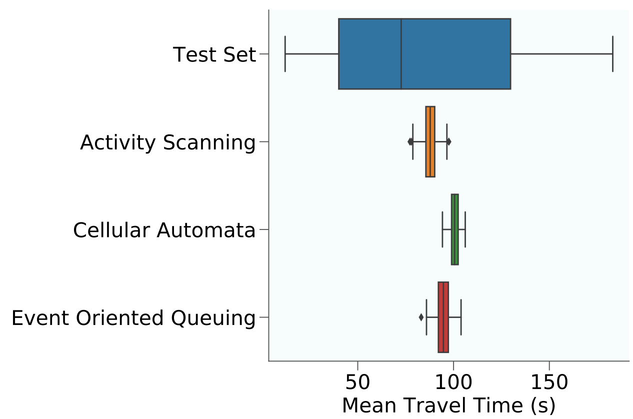


Figure 10: Comparing Models

9 SCENARIOS

In this section, we test some scenarios we thought we interesting, like a traffic flow of entirely motorcycles or trucks, how the rate of flow of traffic can affect travel time.

9.1 Simulating Different Volumes of Traffic

Figure 11 shows the effect of varying the inter-arrival time, and consequently traffic flow rate, of the incoming traffic. From the graph, the mean travel time can be observed to decrease with the increase in traffic flow. This counter-intuitive result is due to the higher traffic flow having higher "back pressure" preventing the new vehicles from entering the system before the lights turns green, whereas with lower traffic, the vehicle can enter the system and get stuck behind a red light.

The variation of λ is manifested in the form of *peak traffic times* in real life like post work hours, or morning hours etc when the number of vehicles entering at any given time is very large as opposed to say lunch time. These two time periods

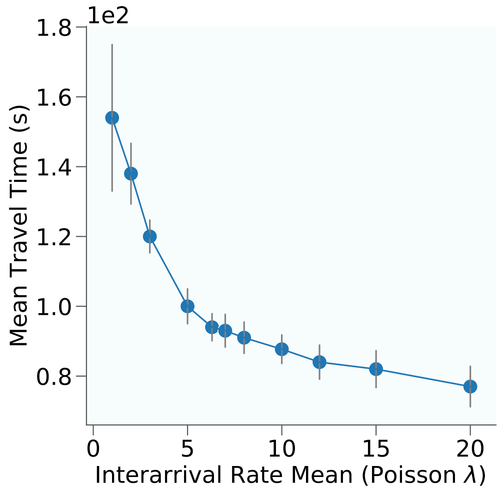


Figure 11: Inter arrival rate vs Traffic Volume

would have a different value of λ with lunch time having a larger value.

9.2 Varying the Length of Vehicles

Figure 12 shows the effect of variation in vehicle length. From the graph, the effect of vehicle length can be observed to be mostly linear with the vehicle length. this is due to longer vehicles spending more time between the point its first part enters the system and its last part leaving the system.

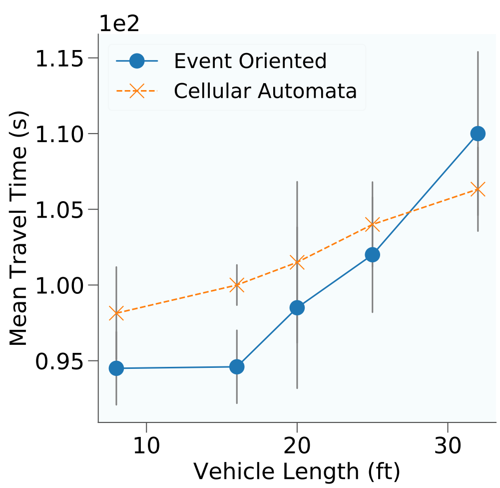


Figure 12: Size of Vehicles vs Traffic Volume

This is similar to real systems as the effect of trucks entering the roads. It is common intuition that big vehicles like trucks take longer to navigate through traffic while shorter vehicles like cycles can quickly slip through it.

10 CONCLUSION

We built models to simulate vehicles driving along the Peachtree corridor. Although we took into account traffic in one direction we analyzed the average travel time for a vehicle to go from the origination zone to the destination zone. We made some simplifications and assumptions. However our simulations were fairly equivalent and were consistent with actual distributions. Our simulations also could be used to simulate other phenomenon (traffic flow and different sizes of vehicles) that weren't present in the NGSIM data but were still able to provide results that had intuitive explanations there by providing credibility to our models.