2019

Simulation of travel time on Peachtree Street

TEAM 41

DONGMIN HAN

SHAN XIONG

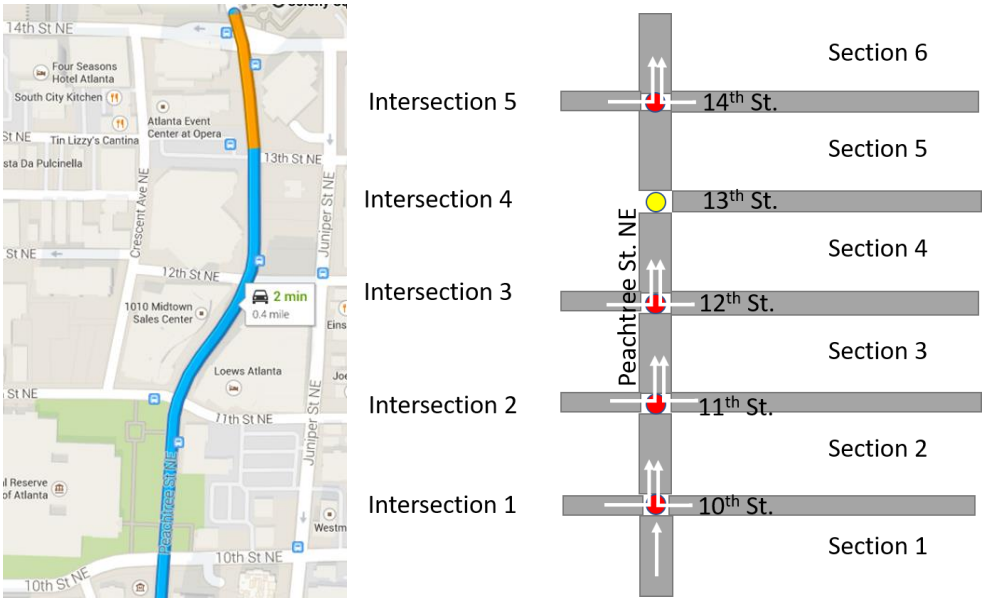
CHONG YE

GitHub repo link: https://github.gatech.edu/dhan64/Traffic-Simulation

1. Problem Statement

The goal of this project is to simulate and study the traffic flow on a portion of Peachtree Street (the corridor from 10th to 14th street) in midtown Atlanta. Peachtree St. is a major street in midtown and serious traffic congestion is a common problem during rush hours. To understand potential factors behind traffic congestions, traffic simulators based various discrete event simulation approaches will be developed. These simulators will be validated with actual traffic records. Based on the simulation results, we will have a better understanding of the main factors that affect traffic conditions, such as traffic light timings, car density, and gap distances.

In our simulation, the system under investigation (SUI) is the traffic flow on Peachtree Street from 10th street to 14th street, as shown in **Figure 1.1a** (from Google Map). A schematic representation of the corresponding road segments is shown in **Figure 1.1b**, which includes five intersections, as represented in red and yellow circles. Here the yellow circle represents a stop sign, while red circles represent traffic lights.



(a) (b)

Figure 1.1. (a) Map of Peachtree St. NE from 10th street – 14th street. (b) Schematic of the road under investigation

In Peachtree St., both north bound and south bound have two lanes. We will mainly focus on traffic flow travelling from south to north, using the actual lane setting in the cellular automata (CA) model. Simplification with neglecting lane-settings is adopted for some other models, such as event-oriented and process-oriented models, based on the nature of the conceptual model. With the two-lane setting in CA, the governing rules of car behavior will include examination of neighboring lane environment and randomized lane-changing decision in addition to basic car-following models. Lane-changing is expected to affect surrounding cars and therefore the overall travel flows.

Our models will also include the traffic lights. There are four traffic lights in Northbound Peachtree St. At the intersections of 10th and 14th St., the traffic lights include left-turn-only signals for left-turn vehicles, whereas at the intersections of 11th and 12th, there are no left-turn signals. We would implement the actual light timing patterns in the simulation.

In order to reflect as real traffic condition as possible, actual traffic records of cars entering the road of interest will be used as input, i.e., the distribution of interarrival time. The output of the simulation, i.e., travel time on Peachtree Street from 10^{th} to 14^{th} street, will be compared with actual statistics to validate and improve the models. Simplification and assumptions are made to ease the implementation.

One specific goal of this simulation study is to assess the average travel time for vehicles to traverse northbound Peachtree Street from 10^{th} to 14^{th} St. under various car densities. In addition, as we worked on this project, we found a secondary goal that we are interested in, which is simulating the synchronization of traffic lights and investigating its impact on travel time. To achieve these goals, we have developed simulators based on three different world views, namely cellular automata, event-oriented, and process-oriented queueing models. In each simulator, we will use the same sets of input to ensure the consistent quality of simulation. Actual traffic records will be used to validate these simulators. We expect to provide a good estimation of travel time for vehicles during various periods, such as rush hours and off-peak hours. More importantly, we will have a better understanding of the modeling and simulation life cycle with hands-on experiments on an actual problem.

2. Conceptual Model

2.1 Cellular automata (CA)

In a CA program, time and space can be divided into discrete steps and cells, where the traffic flow can be represented by the location exchanges among cells within each time step. In our CA model, the sections of interest on Peachtree St. will be divided into discrete grids uniformly. Each grid represents one specific position, which may or may not be filled with vehicles, as shown in **Figure 2.1.1**.

The conceptual model is illustrated in **Figure 2.1.1**. The vehicles will enter at one intersection. The positions of all the cars will be updated every timeframe. The car behaviors over time will be determined by the car in front (leading car) and the traffic lights. For instance, assuming there are L positions in the section of interest, all the possible locations of cars will include $x_i = 0, 1, 2, \ldots L\text{-}1$, where i represent the car ID. The speed range is set to be $[0, v_{max}]$. Based on common driving habits, the philosophy of individual car behavior will be driving as fast as the maximum speed allows and as the car in front allows and decelerate to avoid possible collision (when the distance between current car and the leading car in front, or gap, is smaller than a safe threshold). Specifically, the mathematical rules governing individual car are as follows (for each time step Δt):

i. If change lane, keep v_i
ii. If $Random_i < p$, then $v_i \leftarrow v_i - a_c/2$
iii. Else if $Random_i > p$, then $v_i \leftarrow v_i + a_c$
iv. If $v_i > v_{max}$, $v_i \leftarrow v_{max}$
v. If $v_i < 0$, $v_i \leftarrow 0$.

If $v_i > g_i$, $v_i \leftarrow g_i$

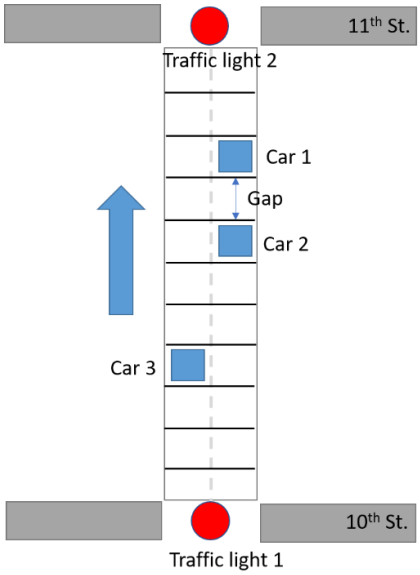


Figure 2.1.1. Conceptual Model of Cellular Automata Simulator

Based on the above rules, the speed and the acceleration of a car are only related to the gap in the front and are independent of speed of other vehicles. Therefore, the state of each car can be updated in parallel. However, these rules are assuming identical drivers' behavior and infinite deceleration rate, which is not accurate for practical cases. Additional rules may be added to illustrate various driving habits (aggressive, conservation, moderate) and real deceleration limitations.

In addition, we are also working on lane-changing behavior of vehicles, which will be more accurate compared with the actual situation. Two-lane model is developed with the rules of lane-changing behavior mainly derived from the work of Nagel and Schreckenberg [1]. Specifically, when all the following rules are satisfied, the ith car will change to the other lane:

vi. If $g_i < l$
vii. If $g_i^{other} > l^{other}$
viii. If $g_b^{other} > l_b^{other}$
ix. $Rand_c < p_c$

Here, l, l^{other}, and l_b^{other} are the parameters to decide how far a car will look ahead on its lane, ahead on the other lane, and back on the other lane, respectively. These rules ensure that the lane-changing behavior will not affect the other cars to a certain level or cause any accidents. The last rule also introduces some randomness, which is closer to the real cases and avoids certain artifacts.

In addition to the above rules, there are a few assumptions in this model as follows:

1. All vehicles enter with the same initial speed, which is the average speed based on NGSIM-data;
2. The effect of two-way stop in intersection 4 is ignored;
3. The vehicle length is set to be the same, which is the average value based on NGSIM-data
4. All the vehicles will travel from intersection 1 to 5. Entering or exiting at other intersections is neglected in this model.

5. In addition to the leading car, each car will also follow the instruction of traffic light if it is fronting the light (meaning there is no other cars between this car and the traffic light in front), just as in real cases.

2.2 Event-oriented model

In the event-oriented model, the traffic flow is driven by sequence of events. In the simulation engine, we have a priority queue data structure, i.e., future event list (FEL) to hold all unprocessed events. In each simulation loop, the event with the smallest time stamp would be removed and handled. Each event can change the state variables and schedule new events.

In this simulation, a few assumptions are made.

6. All vehicles travel at the same speed;
7. The effect of two-way stop control in intersection 4 is ignored;
8. No lane change is considered in this event-oriented simulation;
9. The vehicle length is not considered, therefore the volume of each segment is infinity.
10. At each intersection, vehicles are possible to travel through heading north, turn left or turn right exiting the Peachtree street;
11. If the traffic lights do not have left turn signals, then left turn signals are the same as travelling through signals.
12. When a vehicle is waiting in front of a signal light with other vehicles in front of it, it has to wait a certain time after the front vehicle started to move across the intersection (like in reality).

There are *TurnGreen* and *TurnRed* events for traffic lights and *Arrival*, *Departure* and *Exit* events for vehicles. Boolean variable *isGreen* indicates if the traffic light is green. All traffic light events during the whole simulation time are added to FEL at the initialization since the timing for traffic light events are predetermined. Event *TurnGreen* sets the Boolean variable *isGreen* true and schedule *Departure* for the first vehicle in the queue. Event *TurnRed* just sets the Boolean variable *isGreen* false.

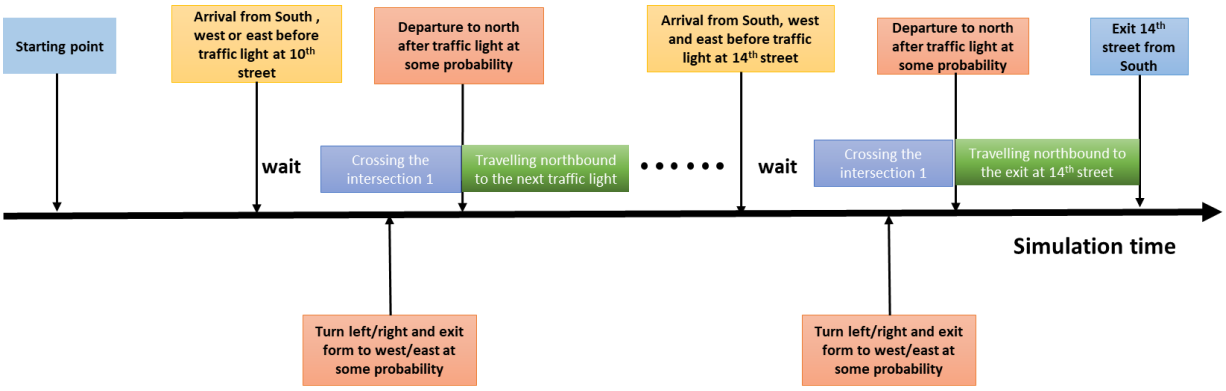
Event *Arrival* handles the vehicles arriving intersections. Vehicles would be added to the queue. At each intersection, vehicles are possible to travel through heading north or turn left/right exiting the PeachTree street based on possibility. In the *Arrival* event, it would also schedule *Departure* or *Exit* event for the first vehicle in the queue.

In the event *Departure*, if the traffic light is green and if the intersection is not the last one (i.e. 14^{th} street), vehicle would be scheduled *Arrival* event to the next intersection after some time. If vehicles arrive the last intersection (i.e., 14^{th} street), vehicle would be scheduled *exit* event.

Event *Exit* handles vehicles exiting the Peachtree street.

The conceptual model is shown in **Figure 2.2.1**.

(a)


Starting point
Arrival from South , west or east before traffic light at 10th street
Departure to north after traffic light at some probability
Arrival from South, west and east before traffic light at 14th street
Departure to north after traffic light at some probability
Exit 14th street from South
wait
Crossing the intersection 1
Travelling northbound to the next traffic light
wait
Crossing the intersection 1
Travelling northbound to the exit at 14th street
Simulation time
Turn left/right and exit form to west/east at some probability
Turn left/right and exit form to west/east at some probability

(b)


10th St & Peachtree
Southbound
Northbound
Westbound
Eastbound
7
3.6
2.2
34.7
3.6
49.3
55
5
4.2
28
3.8
8
1.8
30
0
20
40
60
80
100
GLT YLT RLT GTR YTR R TR/LT
R TR/LT G LT Y LT R LT G TR Y TR


11th St & Peachtree
Southbound
Northbound
Westbound
Eastbound
41.5
3.2
55.4
76.2
20.3
3.6
76.1
20.2
0
20
40
60
80
100
GLT YLT RLT GTR YTR R TR/LT
R TR/LT G LT Y LT R LT G TR Y TR

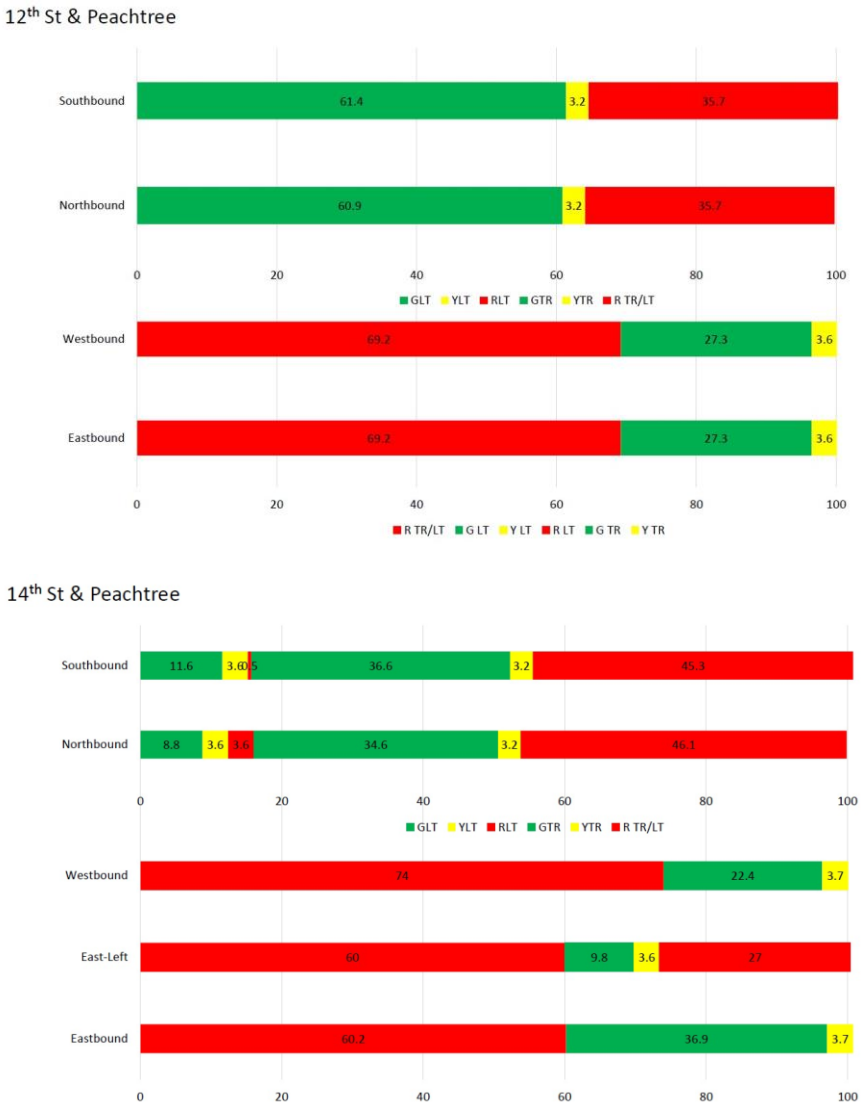


Figure 2.2.1 Conceptual model of event-oriented simulator (a) *Arrival*, *Departure* and *Exit* events

(b) Traffic light events timing

The queueing model for event-oriented simulator is shown in **Figure 2.2.2**.

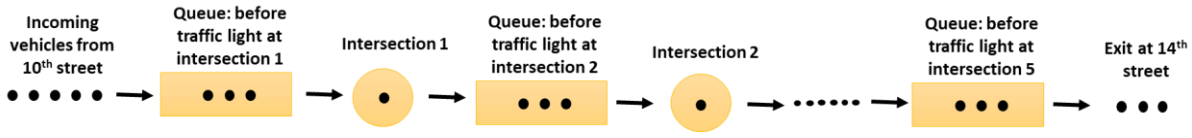


Figure 2.2.2 Event-oriented queueing model

2.3 Process-oriented model

The conceptual model of process-oriented model is shown in **Figure 2.3.1**. In process-oriented model, the traffic flow is driven by pausing and resuming each vehicle thread. In the thread of the scheduler, there are

two list. One is the event list (FEL) implemented in a priority queue, the other is the waiting vehicle list which is implemented by a LinkedList. In each loop of the scheduler, the first event with the minimum time stamp is handled, then, the entire waiting vehicle list will be checked to see if any of the vehicles thread can be resume. In this model, even though it is implemented by multi-thread programing, there is only one thread running at a time, either a vehicle thread or the scheduler thread.

This model is based on a few assumptions:

1. All vehicles travel at the same speed;
2. The effect of two-way stop control in intersection 4 is ignored;
3. No lane change is considered;
4. The vehicle length is not considered, therefore the volume of each segment is infinity.
5. At each intersection, vehicles are possible to travel through heading north, turn left or turn right exiting the Peachtree street;
6. If the traffic lights do not have left turn signals, then left turn signals are the same as travelling through signals.
7. When a vehicle is waiting in front of a signal light with other vehicles in front of it, it has to wait a certain time after the front vehicle started to move across the intersection (like in reality).

There are mainly 3 types of event in this model: vehicle thread event, traffic light event and scheduler event. Vehicle thread event consists of the status changing events, including *Enter*, *Exit* and *Resume*. Traffic light events are basically turning signal lights red or green. Scheduler events include *WaitUntil* and *CheckWait*. A *WaitUntil* event will add corresponding vehicle into the waiting vehicle list with the intersection and deirection of the signal light the vehicle is waiting for. *CheckWait* will be handled by forcing the scheduler to scan the waiting vehicle list.

The vehicle thread is the essential part of this model. In each thread, the program must handle all different cases with the vehicle entering from and exiting to different directions and intersections. And at each intersection the vehicle has to across, it will calculate the exit direction based on the empirical probability distribution extracted from the given data, then, the thread will resume the scheduler, sent the *WaitUntil* event to the scheduler thread, and pause itself.

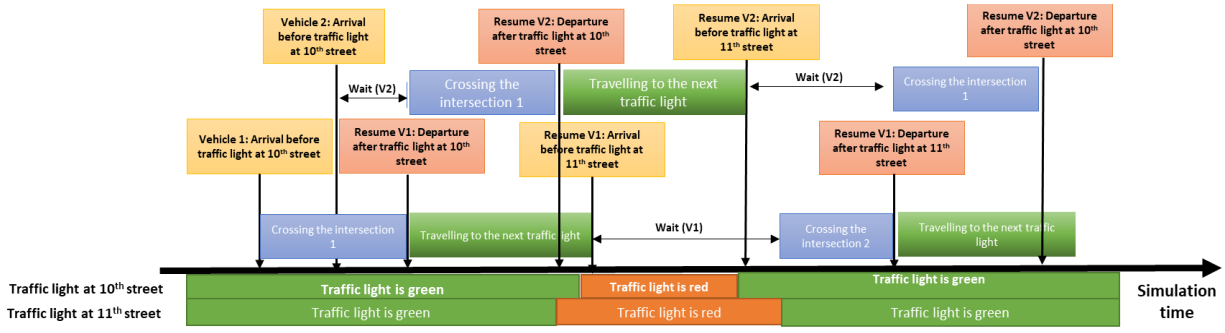


Figure 2.3.1. Conceptual model of process-oriented simulator

3. Input Analysis

To generate car flow for initial tests, we analyzed the real traffic records based on the NGSIM-Data provided by the class and generated the input car flow following the same pattern, or the same distribution of inter-arrival time periods. The detailed process for NGSIM-Data analysis is as follows.

First, to filter out the noises of the collected data, all the cars with total frame less than 20 (total time in the corridor < 2 seconds) are excluded. Then, only the cars with directions equal to 2 (only the cars driving northbound) are investigated, which gives us 290 cars in total during 4:00 – 4:15 pm. To further visualize the behavior of individual cars, we observed the evolution of sections, intersections, and movements based on the interactive plots, as shown on one example in **Figure 3.1**. The x axis indicates the time flow (unit: 0.1 seconds) in the form of frame ID. The top two figures show that the car entered the corridor at intersection 1 (Peachtree & 10th St.) and traveled through section 2, 3, 4, and 5, and exited at intersection 5 (Peachtree & 14th St.). During this process, the movement of this car kept being 1, meaning it drove straight without any turns. According to these plots, we can calculate the distribution of the entering and exiting behavior of all the cars of interest by examining the intersection and movement parameters at the beginning and the end of each car, respectively. The results are shown in **Table 3.1** and **Table 3.2**.

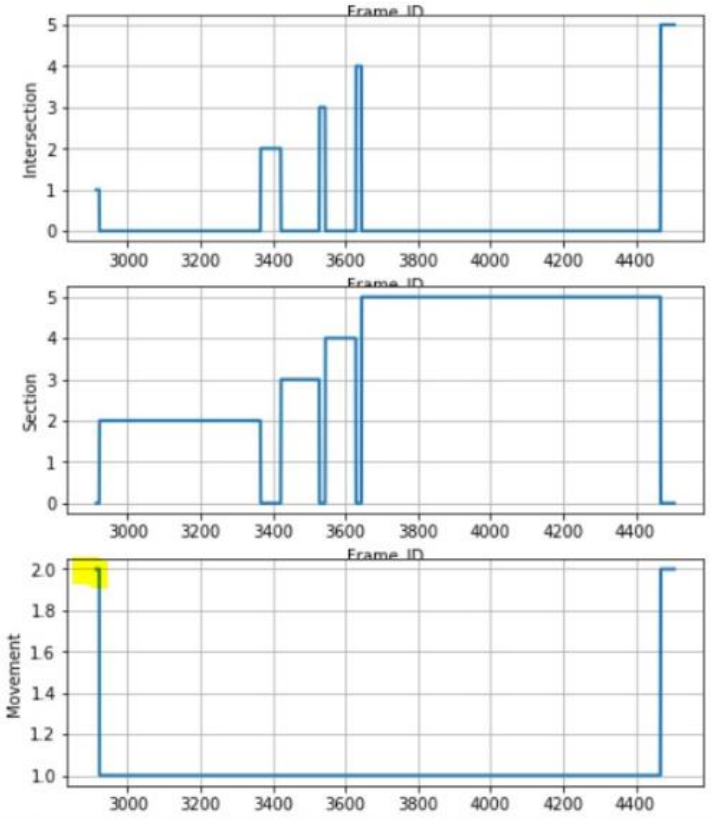


Figure 3.1 Illustration of one car behavior across the corridor

Table 3.1 Statistics of entering behaviors of cars

ENTERING INTERSECTION	MOVEMENT	VEHICLE COUNTS
1	1 (travel through)	91
	2 (left turn)	61
	3 (right turn)	31
2	1	138 (NA)
	2	8
	3	4
3	1	128 (NA)
	2	8
	3	10
4	1	141 (NA)
	3	6
5	1	125 (NA)
	2	70
	3	12

Table 3.2 Statistics of exiting behaviors of cars

EXITING INTERSECTION	MOVEMENT	VEHICLE COUNTS	PROBABILITY
1	1 (travel through)	NA	0.737
	2 (left turn)	36	0.197
	3 (right turn)	12	0.066
2	1	NA	0.94
	2	7	0.047
	3	2	0.013
3	1	NA	0.918
	2	7	0.048
	3	5	0.034
4	1	NA	0.993
	2	0	0
	3	1	0.007
5	1	125	0.604
	2	70	0.338
	3	12	0.058

As shown in **Table 3.1**, other than the artifacts for intersection 5, most cars entered the corridor from intersection 1 (Peachtree & 10th St.). Half of them entered straightly from the south, others from 10th by turning left or right. There are fewer cars entering from the other intersections by turning left/right. Therefore, for some of our models, such as CA, we only considered cars entering at intersection 1 from three directions and neglected the cars entering from other intersections.

For exiting the corridor, as shown in **Table 3.2** in red fonts, for intersection 1 through 4, majority of the cars traveled through the intersection. Only a small portion of cars exited at intersection 1 ~ 4. Most of them exited the tracking system at intersection 5 in 3 directions. Therefore, for some models (CA), we mainly focused on car exiting at intersection 5.

To generate input car flow for our models, we investigated the car entering flow at each intersection by calculated the distribution of inter-arrival time periods. The statistics of car-entering inter-arrival times is shown in **Figure 3.2**. Based on these results, we can generate the probability for car entering intervals, an example is shown in **Table 3.3**. Basically, whenever a car enters the intersection, the next car entering at the same position in the same direction will be generated after certain period of time (within 50 second in this example) with a certain probability. The heading line indicates the intersection 1 (1), travel through (1), and total numbers of the following bins/lines (14) for car entering, shown as #1 1 14. The input file generated this way follows the same pattern as the empirical data, and therefore can be a good representation of the actual system under investigation (SUI). Similarly, all the car flows entering various intersections with various entering directions can be shown in the files, including #11, #12, #13, #22, #23, …, #52, #53 (from intersection 1 to intersection 5, all entrances).

As a goal of our investigation, we will need to analysis how the travel times vary with different car flows with 15 minutes. Based on the distribution generated from real data, we kept the general car flow distribution at each entrance and changed the inter-arrival periods by multiplying a certain coefficient. For example, based on #1 1 14 for intersection 1 travel-through cars, as show in Table 3.3, we doubled the values of first column to extend the inter-arrival time periods for entering cars, which makes the total number of car entering the corridor less, leading to less car density. If we use 0.5 time the periods, it will lead to smaller intervals and larger car densities. We can easily generate various levels of car entering flows that are close or similar to actual distributions.

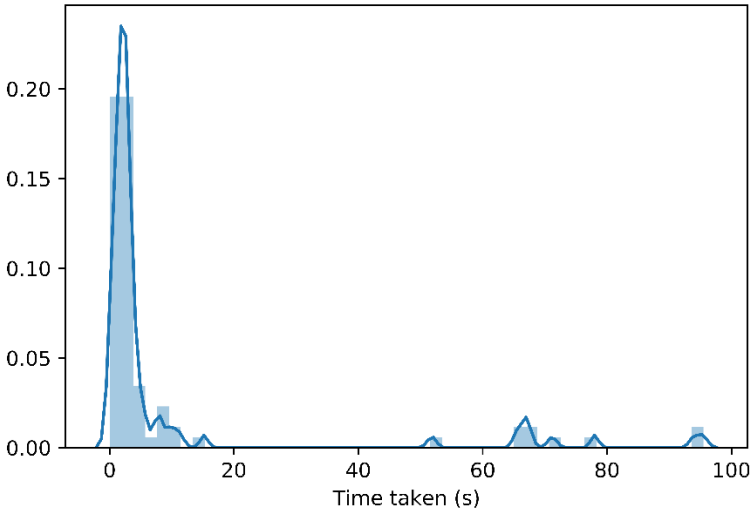


Figure 3.2 Probability of inter-arrival time periods of cars entering intersection 1 from south

Table 3.3 one example of input file used in our models (car entering from south at intersection 1)

#1 1 14	
2	0.366667
4	0.388889
6	0.066667
8	0.033333
10	0.011111
12	0.022222
14	0
16	0.011111
30	0
40	0
60	0.011111
70	0.044444
80	0.022222
100	0.022222

In addition to input files, we also need some other parameters for the simulations. For event-oriented and process-oriented models, whenever a car leaves one intersection, it will travel a certain period of time before reaching the next intersection and waiting for the lights. The average traveling time periods based on real data are calculated and listed in **Table 3.4**. It indicates how long a car normally takes to travel through a particular section before entering the queue for events of traffic lights.

As shown in the Table, each section requires different travel times, since they have different lengths. These values do not include the waiting times for traffic lights, which will be taken care of by other event handlers. Also, it is worth noticing that we didn't separate the time periods before and after intersection 4, which is a stop sign. This is because we didn't simulate the event for cars traveling through stop signs, which follows a different behavior from traffic lights. As a simplification, we only used the total travel time from intersection 3 to 5 in our models to eliminate the influences of stop signs.

Table 3.4 Travel times between the intersections

Intersection	*Time (sec)*
<1	14.14
1-2	26.12
2-3	36.07
3-5	55.63
>5	6.14

For Cellular Automata (CA) model, we need the average entering speed for the cars as initial velocity, which is calculated to be 22.59 ft/s.

4. Simulation Model

4.1 Cellular automata (CA)

The code framework of CA model is show in **Figure 4.1**.

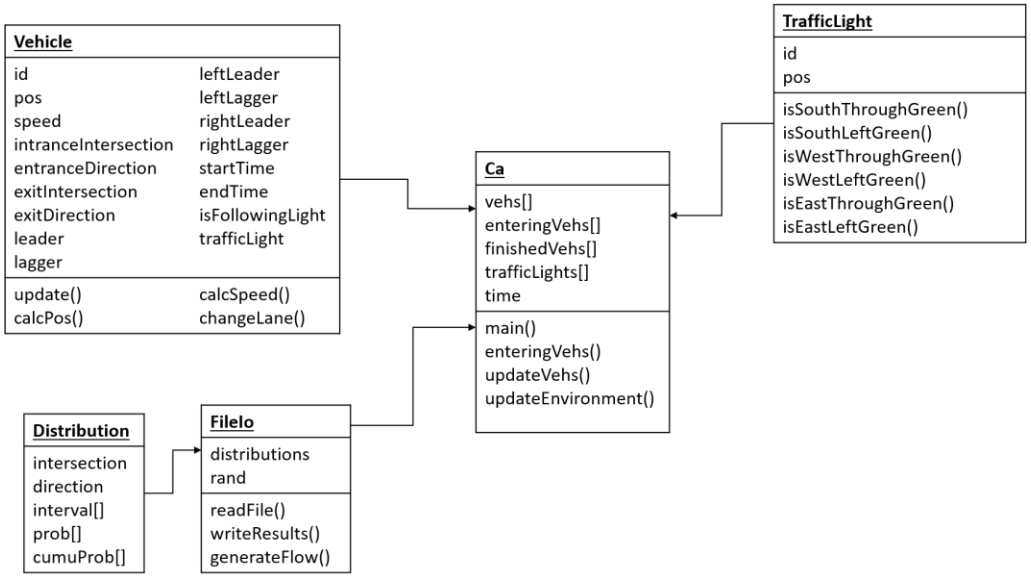


Figure 4.1 Code framework of Cellular Automata

4.2 Event-oriented model

The code framework of event-oriented model is show in **Figure 4.2**.

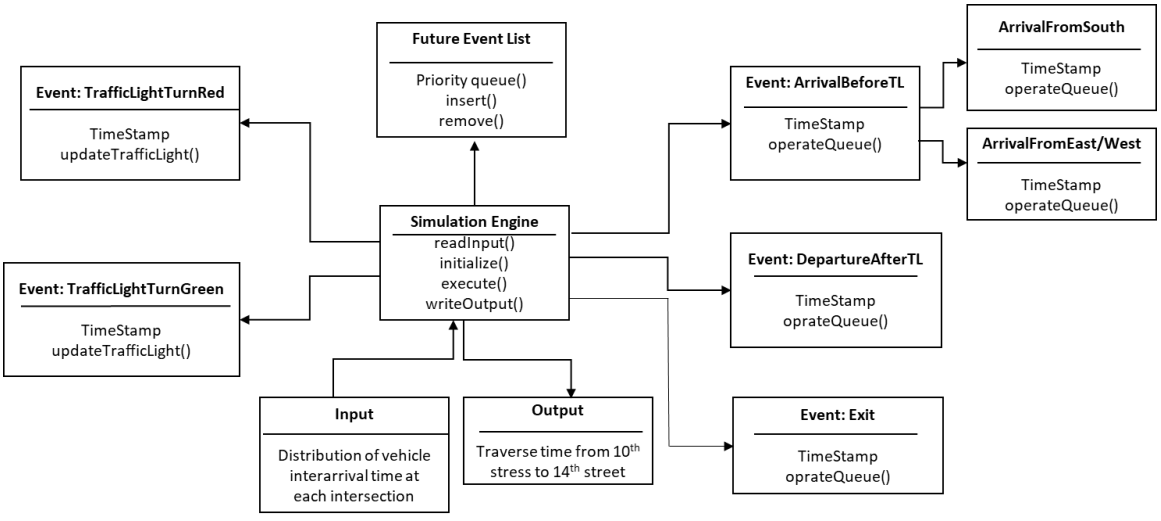


Figure 4.2 Code framework of event-oriented simulator

4.3 Process-oriented model

The code framework of event-oriented model is show in **Figure 4.3**.

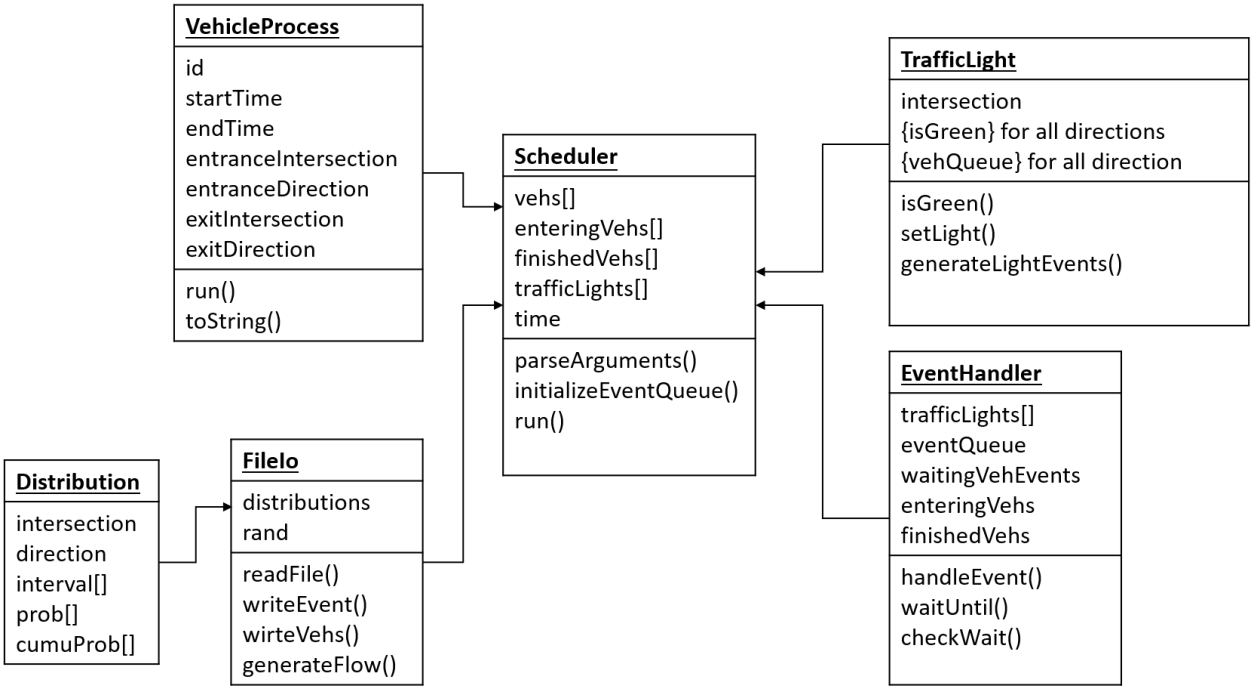


Figure 4.3 Code framework of process-oriented simulator

5. Verification & Validation

5.1 Verification

Verification is to ensure the implementation of the computerized models are correct and can represent the conceptual models we developed. To do this, we mainly did static verification. All the team members went through the code line by line structurally to check the implementation of the algorithm. The code can run smoothly without errors. The log book of individual car behavior shows that the implementation is correct and the travel routes of all the cars make sense without obvious anomaly.

Specifically, there are two kinds of output files as our simulation results. One is event files with names of "event_*", which serve as log book to keep records of all the updates in each step. For event- and process-oriented models, it recorded the information of every event updates. In CA, it recorded the driving parameters and local environment of each car at each moment. Another type of output files is the whole path of each vehicle, with the names of "vehs_*". There are 7 or 8 columns in this file, which include car ID, lane ID (if applicable), entering time, exiting time, entering intersection, entering direction, exiting intersection, and exiting direction. We mainly analyzed the output based on the "vehs" output files.

5.2 Validation

Validation is to confirm that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model. Strictly speaking, we cannot prove our model is absolutely valid, but we can fail to prove it is invalid, thus enhance the confidence to apply our simulation models.

To evaluate how well our models represent the actual system, we collected the travel times for the cars using various models and compared them with the statistics of actual data. We warmed-up our system and only kept the results after the first car reaching intersection 5, which is typically 200 seconds. Then for all the cars that travelled from the southmost entrance to the northmost exit (from intersection 1 to 5), we analyzed their total travel times and compared with actual data. As shown in **Figure 5.2.1**, most cars traveled 150s to 270s in the corridor, which is a bit higher than the average value based on actual data (100s ~ 150s) but is acceptably close. The process-oriented model produced similar results, as shown in **Figure 5.2.2**. The output of event-oriented and process-oriented models are very close, which is reasonable. These results confirmed that our models are valid to the best of our knowledge.

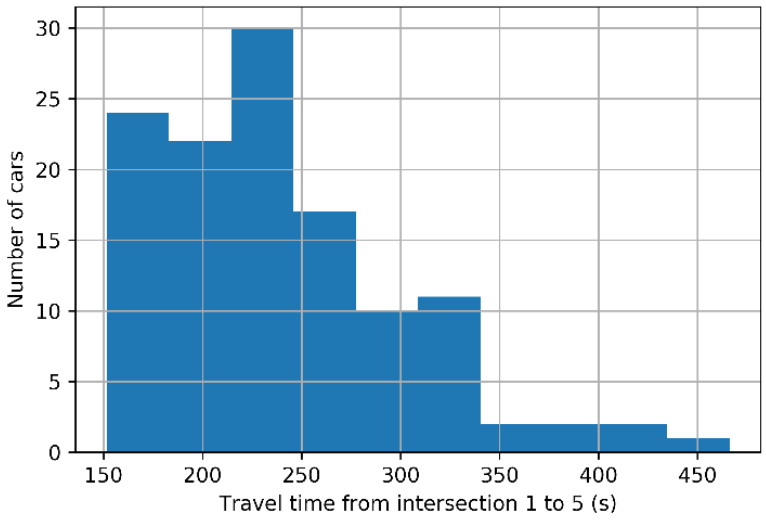


Figure 5.2.1 Histogram of cars travelling time (sec) from 10th St to 14th St though Peachtree St NE using event-oriented model. (Total car number: 121)

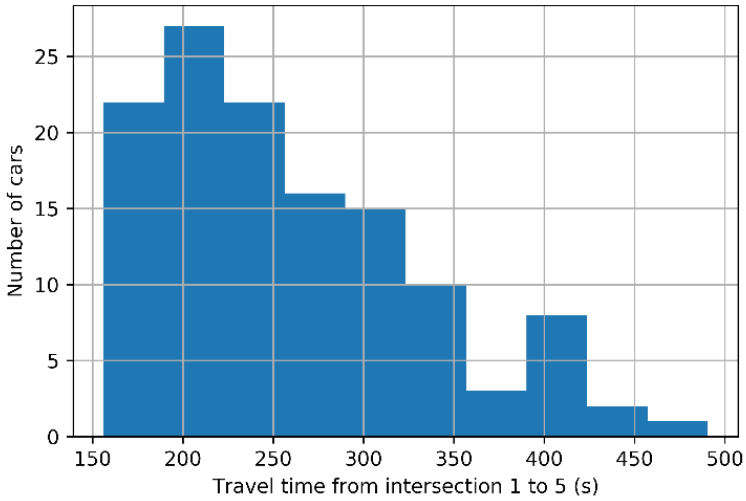


Figure 5.2.2 Histogram of cars travelling time (sec) from 10th St to 14th St though Peachtree St NE using process-oriented model. (Total car number: 126).

The simulation results of CA model are shown in **Figure 5.2.3**. This model is built up with a very different conceptual model compared with event- or process- oriented models as stated in Section 2. As shown in **Figure 5.3**, the travel time from intersection 1 to 5 using CA tends to be shorter and is closer to the actual data.

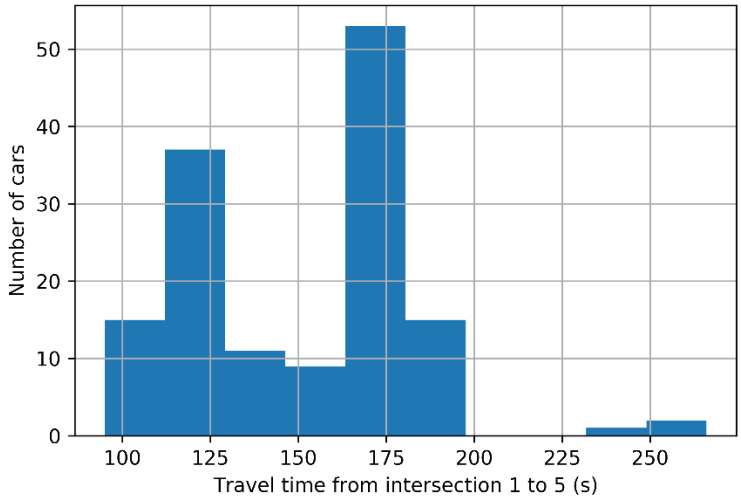


Figure 5.2.3 Histogram of cars travelling time (sec) from 10th St to 14th St though Peachtree St NE using cellular automata (CA) model. (Total car number: 143).

To further demonstrate the validation of our models, we introduced randomness into our input files for the event- and process-oriented models. With various seeds generated randomly, the simulation will take different input each time based on the same distribution. For event-oriented model, we can see from **Figure 5.2.4** that the average travelling times are very stable (ranging from 270s to 310s) throughout the 10 cases.

Similar results are also obtained for the process-oriented model, as shown in **Figure 5.2.5**. In contrast, simulations using CA model show shorter average travelling times (**Figure 5.2.6**), which is closer to real cases. The distributions of CA results are also very stable. With these results, we can be more confident with our models on further studies. In the following part, we will use event-oriented and process-oriented models to do multiple simulations, and the results will be demonstrated and discussed in a similar way as here.

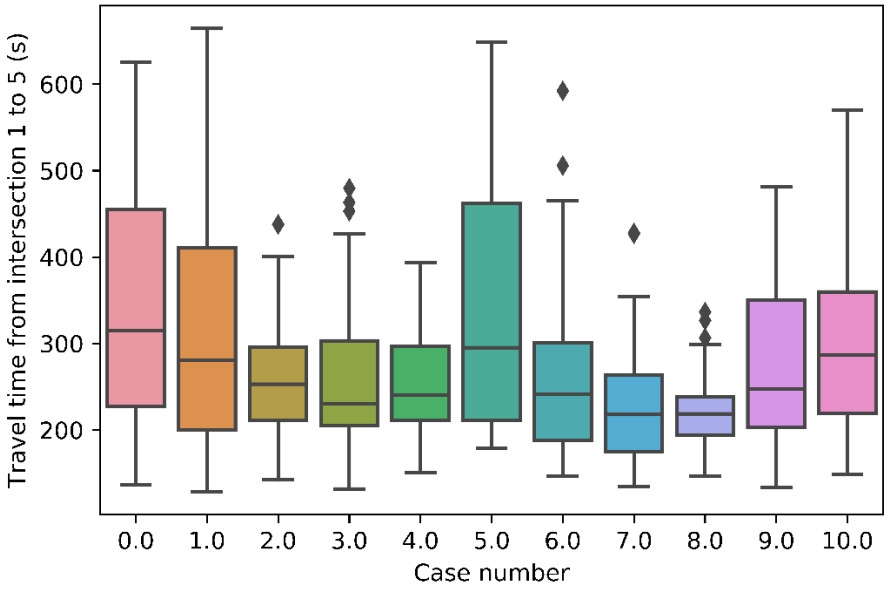


Figure 5.2.4 Statistics of 10 simulation trials of travelling time (sec) from 10th St to 14th St using event-oriented model.

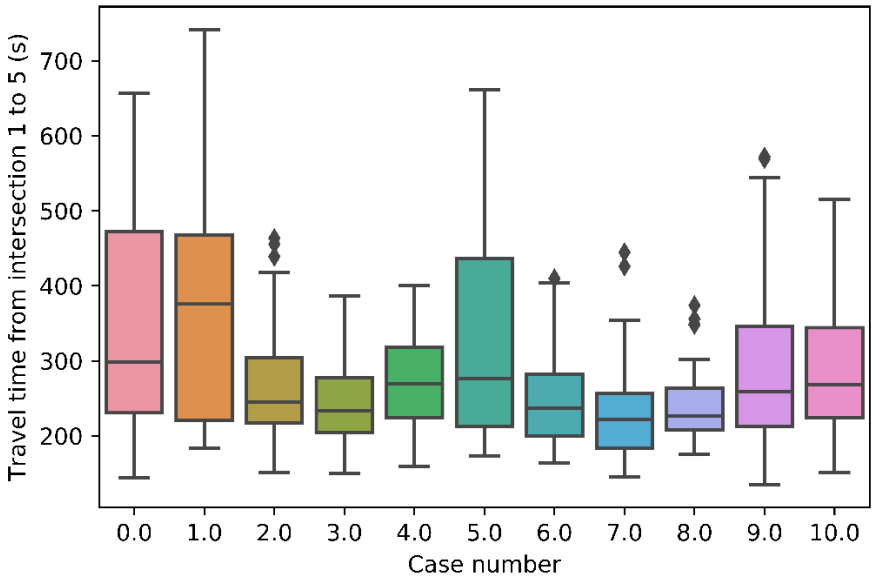


Figure 5.2.5 Statistics of 10 simulation trials of travelling time (sec) from 10th St to 14th St using process-oriented model.

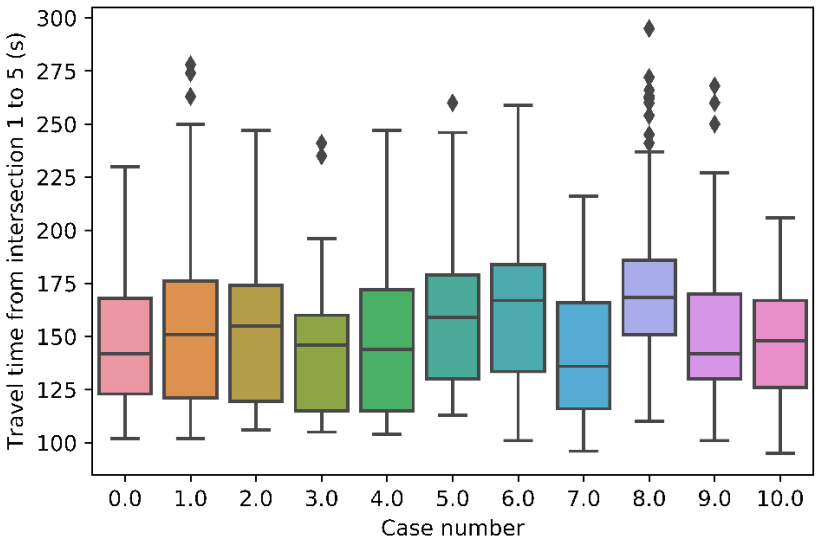


Figure 5.2.6 Statistics of 10 simulation trials of travelling time (sec) from 10th St to 14th St using CA model.

6. Experiment design and output analysis

6.1 Investigate the impact of car density on travel time

In this section, we will mainly use event- and process- oriented models to investigate how the average travel time varies under various car densities or car flow rates. As mentioned in section 3, to generate input files with various car densities, we kept the general car flow distribution at each entrance based on the distribution from real data and changed the inter-arrival periods by multiplying a certain coefficient (>1 for longer interval and lower car density, <1 for shorter interval and longer car density). For example, if we use 0.5 time the periods, it will lead to smaller intervals and larger car densities. We can easily generate various levels of car entering flows that are similar to actual distributions. The car densities will be indicated by level 1 to level 4, with level 1 being the lowest car density and level 2 being the highest. Based on our results, level 4 is corresponding to about 350 cars entering intersection 1, and level 1 is about 35 cars. For each parameter or each level, we ran 100 simulations using randomly generated seeds. This will ensure our results are representative of the system and exclude possible artifacts.

To warmed-up our simulations, we used the same method as mentioned in section 5.2. Specifically, we only kept the results after the first car reaching intersection 5, which is typically 200 seconds. This indicates the whole corridor has been filled with cars. Then we filtered the results by keeping the results from the cars that travelled from the southmost entrance to the northmost exit (from intersection 1 to 5). The results simulated by event- and process-oriented models with the same parameter (density level) are put together and compared to ensure the validity, as plotted in the following figures.

As shown in **Figure 6.1.1** through **6.1.4**, the input file produced car densities from low to high, corresponding to level 1 through level 4. There are about 30, 60, 160, and 320 cars entering intersection 1 for level 1, 2, 3, and 4, respectively. In each figure, the left panel represents simulation results by event-oriented model, and the right represents process-oriented model. For each panel, it includes the statistics of

all the 100 trails with 100 random seeds using certain model. As shown in the figures, each panel shows stable distribution of travel times, confirming the validity of our models.

By comparing the results from two models using the same parameter, we can see that event- and process-oriented models indeed generated very similar results, in terms of average travel times, deviations, maximum and minimum values, etc.

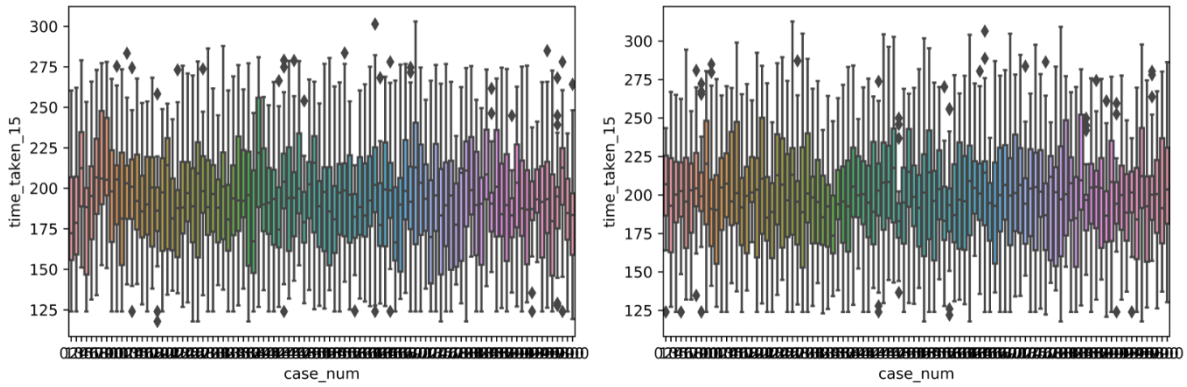


Figure 6.1.1 Statistics of 100 simulation trials of travelling time (sec) from 10th St to 14th St using (left) event-oriented model and (right) process-oriented model, respectively. *Density level =1*

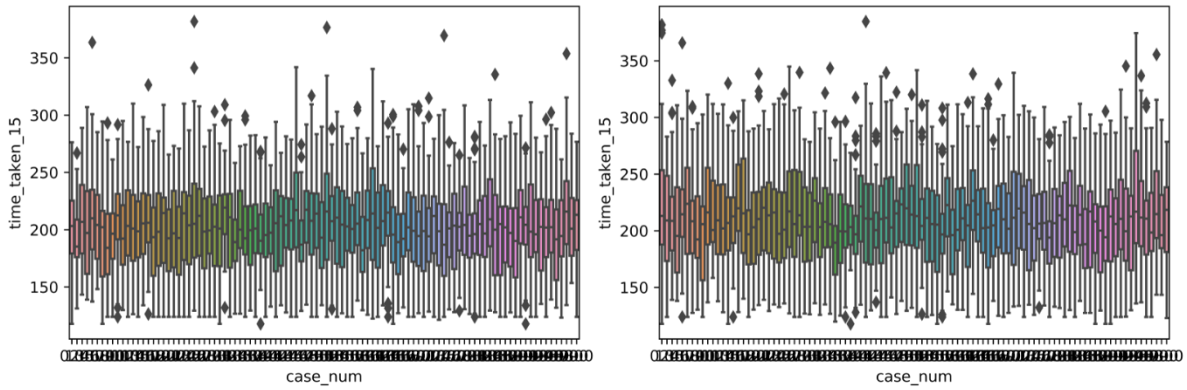


Figure 6.1.2 Statistics of 100 simulation trials of travelling time (sec) from 10th St to 14th St using (left) event-oriented model and (right) process-oriented model, respectively. *Density level =2*

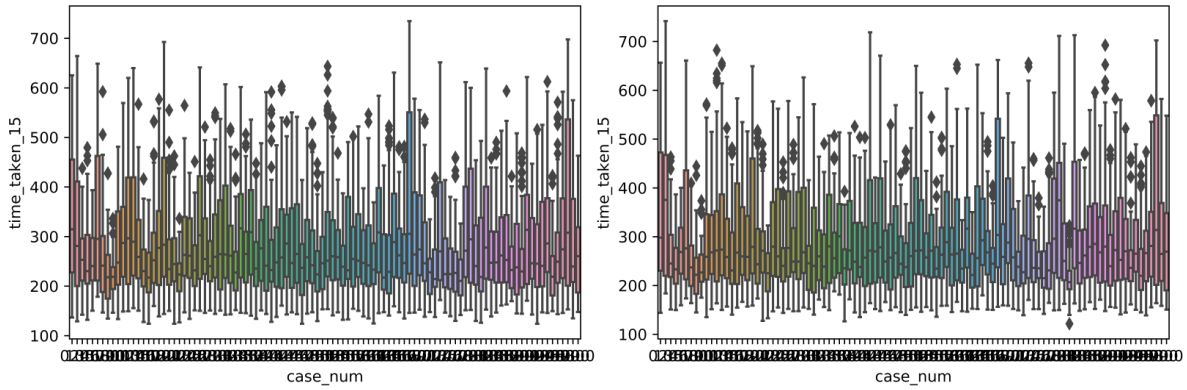


Figure 6.1.3 Statistics of 100 simulation trials of travelling time (sec) from 10th St to 14th St using (left) event-oriented model and (right) process-oriented model, respectively. *Density level =3*

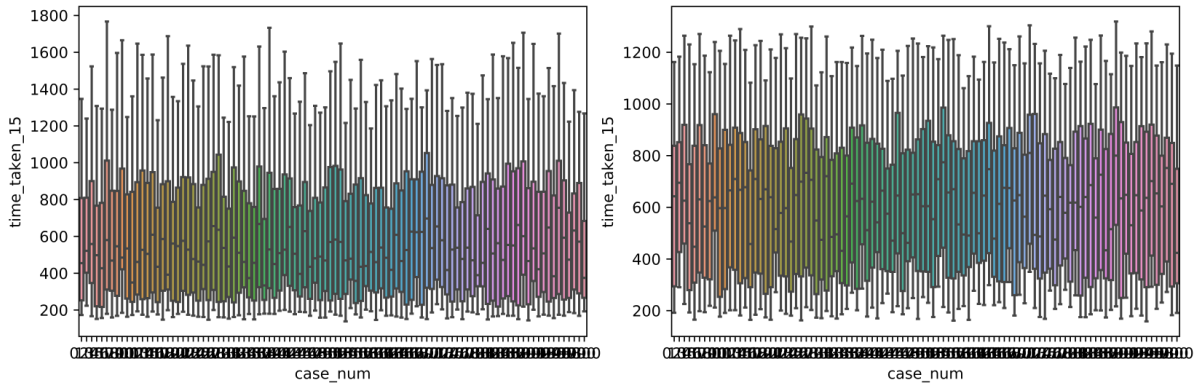


Figure 6.1.4 Statistics of 100 simulation trials of travelling time (sec) from 10th St to 14th St using (left) event-oriented model and (right) process-oriented model, respectively. *Density level =4*

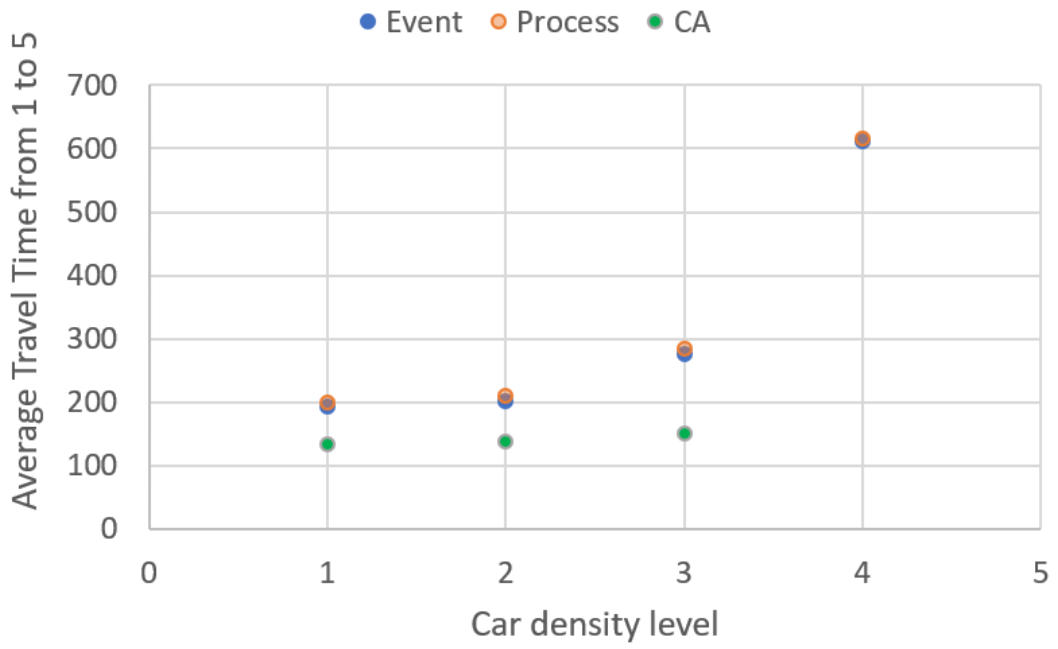


Figure 6.1.5 Average travel time vs. car density level. Based on 100 simulation trials using event-oriented model, process-oriented model, and CA model, respectively

Furthermore, if we compare the results among various car densities, we can clearly see that with higher car density levels, the travel time from intersection 1 to 5 increases (**Figure 6.1.5**). This is reasonable since higher car density means more waiting time and slower motion due to conjunction. For level 1 to level 2, CA models are also used for simulation and the results are also included in **Figure 6.1.5**. The results of CA are generally lower than the other two models and are closer to real cases. Further investigation on the discrepancy among the models are needed. So far, all of them generated acceptably similar results and

patterns. These results show clear trends and can be reasonably used for traffic prediction upon some level of optimization and refinement.

6.2 Investigate the synchronization of traffic signals

During the development and testing of our models, we found that the alignment of traffic lights play a critical role in total travel time. Therefore, we want to use our model to investigate whether the fully synchronized traffic lights and the non-synchronized lights generate different results. As shown in section 2, the northbound traffic lights at 4 intersections all have roughly the same period, which is 100 seconds. If all the traffic lights run based on the same pattern as we plotted, meaning all the northbound green lights start at the same time, we call it fully synchronized. For unsynchronized case, we shift the starting point of each sequential green light (lights at 11^{th} St to 14^{th} St) by 25 seconds or 50 seconds with respect to the one before it (to the south). Then we ran 30 simulations for each case using process-oriented model with random seeds.

The results are plotted in **Figure 6.2.1**, with top panel representing fully-synchronized case, middle delay = 25 seconds, and bottom delay = 50 seconds. The average travel time throughout all the trails for each case are 310.5, 310.2, and 292.3 seconds. We can see that the synchronization of traffic lights do have an influence on the average travel time as well as the distrubution. Although the differences between our resutls are not huge, which may be due to the car density we used here, our models and simulations indeed show that certain relationships exist between timing of traffic lights at sequential intersections and the total travel times. Further investigations are needed to clarify the detailed relationships.

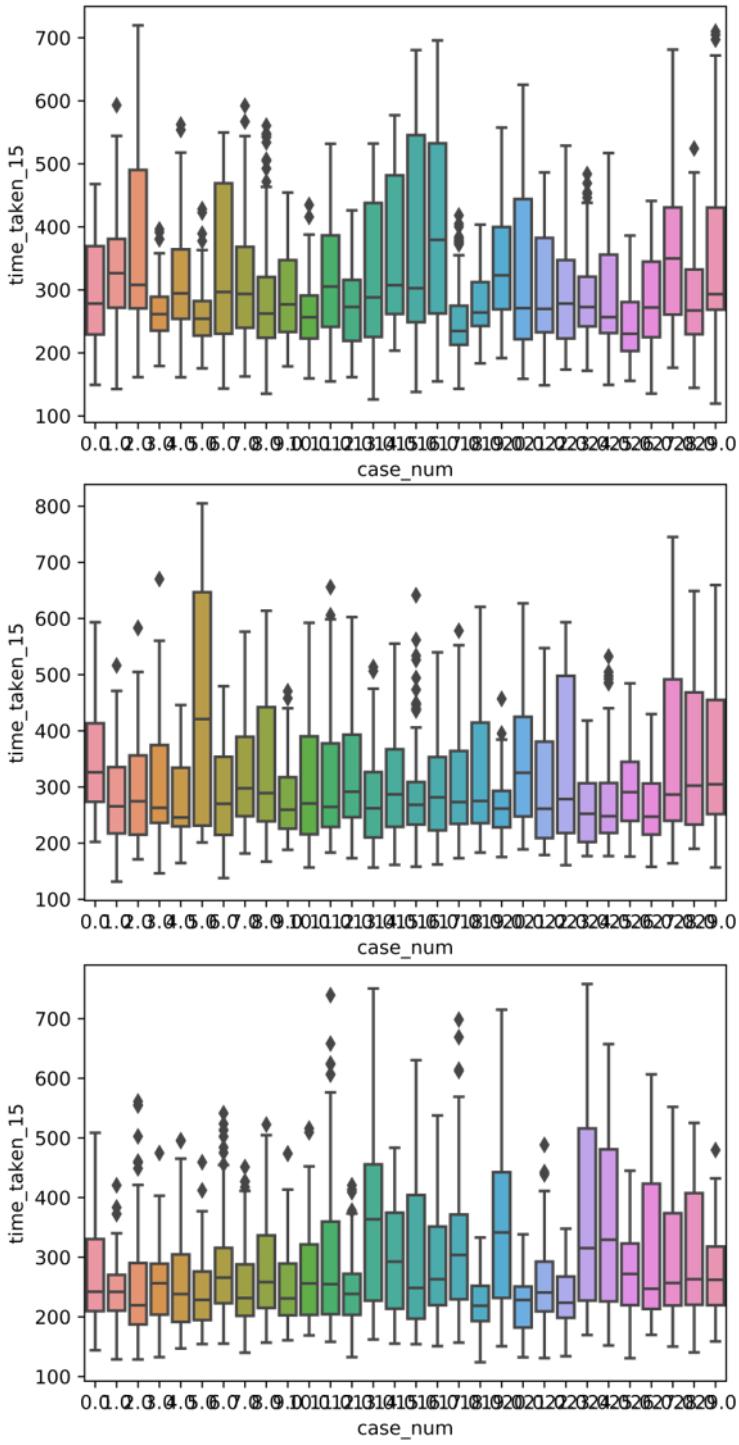


Figure 6.2.1 Statistics of 30 simulation trials of travelling time (sec) from 10th St to 14th St using process-oriented model. From top to bottom, delay = 0 (synchronized), 25 sec, and 50 sec, respectively

7. Conclusion

In this project, we have developed software packages based on three conceptual models to simulate the traffic flows on Peachtree St from 10th St to 14th St. All these 3 models are validated by comparing with the actual data collected from real traffic situations. Randomized simulations are run by introducing random seeds for the input files. Among these models, event- and process-oriented models are similar to each other by being constructed with similar sets of modular components, such as event handler, event scheduler, entity classes, etc. They generated reasonably reliable results and showed clear trends based on various car densities, which can be useful for traffic condition prediction. We have also investigated the effect of synchronization of traffic lights on overall travel times using process-oriented models. Further investigations are expected to reveal the in-depth relationships between these factors and traffic flows.

8. Member contributions, Github repo link

During the development of our models, we have held multiple group meetings to discuss all aspects of the project, which allowed everyone to stay posted with the progress and to give suggestions on possible issues. Coding work is split among group members to ensure efficiency. Specifically, Chong was in charge of event-oriented model, Dongmin in charge of process-oriented model, and Shan in charge of cellular automata model. In addition, Dongmin worked on the input file analysis and generation, while Shan and Chong mainly contributed to running the simulations and output analysis. All members have worked together on the reports.

Github repo link: https://github.gatech.edu/dhan64/Traffic-Simulation

9. References

[1] Nagel, Kai, and Michael Schreckenberg. "A cellular automaton model for freeway traffic." Journal de physique I 2.12 (1992): 2221-2229.