ECE 6730 PROJECT 2: FINAL REPORT

Caitlyn Caggia, Blaine Costello, Lakshmi Raju

Github Repository: https://github.gatech.edu/ccaggia3/ECE6730Project2

PROBLEM DESCRIPTION

This project will demonstrate how different models analyze the flow of one-directional traffic through a section of Peachtree street between 10th and 14th Street to predict travel time. Of the five intersections on this route, all contain stoplights except for the intersection of Peachtree and 13th Street. The 13th Street intersection is regulated by a one-way stop sign where vehicles coming off of 13th Street must yield. Northbound vehicles may only turn right onto 13th Street; Southbound vehicles can only turn left.

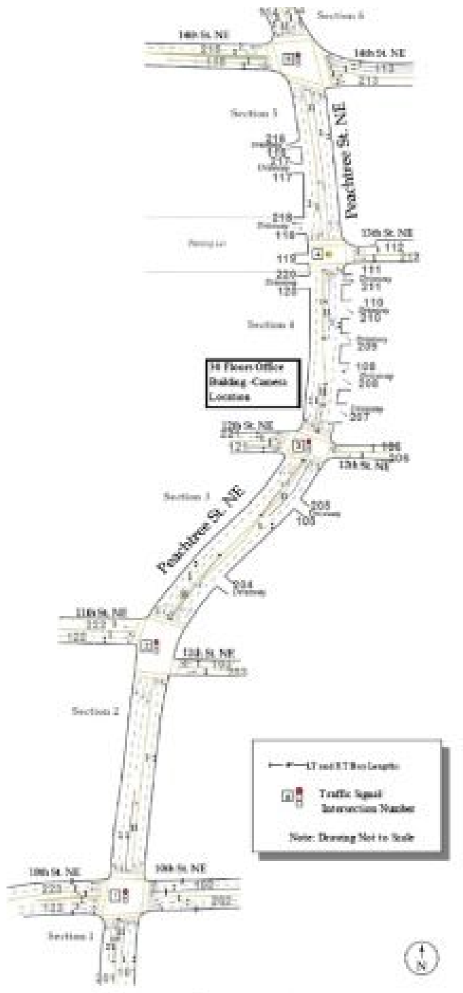

Figure 1. *Map of the section of interest of Peachtree Street NE, with small indicators on each intersection to indicate the traffic laws and driveways shown. All streets but 13th Street have a stoplight; 13th Street has a stop sign instead.*

Initial testing will consist of simplest-case scenarios to ensure proper functionality.

- Single car driving along the corridor northbound with no interruptions.
- Single car driving along the corridor northbound, interrupted by red lights.
- Multiple cars driving along corridor northbound, interrupted by red lights.

- Multiple cars driving along corridor northbound, with probability of turning left or right.

These initial tests will verify the proper functionality of all data structures for basic functionality in all three of the methods explored in this work. Additional tests will be required to obtain total functionality, and finally, several larger-scale, experimental scenarios will be addressed with further exploration in the models based on the various traffic levels measured by vehicle arrivals per minute:

- Random generation of cars entering corridor
- Stochastic behavior determination
- Multi-lane dynamics
- Cars passing cars
- Congestion of roadways via gridlock
- Congestion of intersections via “cautious left-turners”
- Blocked intersection due to cars stopped in the middle of intersection (cross-traffic congestion)
- Irregular factors, like construction or an accident (lanes are blocked/merging)

There are two NGSIM datasets for Peachtree Street NE from 12:45 pm to 1:00 pm and from 4:00 pm to 4:15 pm on November 8, 2006. The same cameras and data collection systems are used for both periods.

CONCEPTUAL MODEL

Underlying Assumptions

Some assumptions are so critical that they are common to all three models:

- Gridlock is not an issue, and that vehicles may clear the intersection as soon as they enter it.
- Cars simulated travelling down the corridor only in the northbound direction.
- No cars are passing any other cars (FIFO).
- There are no models that account for car accidents, it is assumed that all drivers are vigilant and adjust their speed accordingly
- This is only to be used with this specific section of Peachtree St NE, the models cannot be reused for other traffic corridors without modification.

Model Details

1. Event Oriented Queueing Model

The Event-Oriented Queueing Model is named as such because it describes an architecture which uses events (or significant changes in state) to initiate sequences of action. In this particular model, the traffic lights are the primary action initiators which will motivate the locomotion of vehicles through various sections of the peachtree artery. We refer to the cars in this system as consumers because each car is driven by the events dictated by the traffic light, and each car consumes a portion of the resource that is "roadway space."

It is important to first define some simplifications and assumptions made specifically for this event-oriented model. To provide the most general implementation of this model, only northbound traffic is simulated along the specified corridor. This will cause only a minor (if any) discrepancy in left-turn wait time, which will primarily be overshadowed by the turn-lights for which we are given timing information. The event oriented approach relies heavily on problem simplification and model abstraction to discretize the ongoings of a particular system into very few computationally efficient, yet accurately representative chunks called events. That stated, the following is also assumed to reduce complexity:

***Cars*:**

- Cars travel at one of two speeds to account for stop-and-go acceleration time.
- Cars will travel slower for one block after stopping at a red light
- Continuing through a green light, cars keep constant, uniform speed.
- Cars' intended path is determined prior to entering the roadway
- A car exiting the roadway before 14th will randomly choose to turn left or right
- All cars strictly obey all traffic lights, and do not stop unnecessarily.
- There are no lane-changes, and cars do not pass each other.

Intersections:

- A congested roadway is treated as a red light
- The flow of traffic into the corridor via 13th street is allowed only when no cars are in the preceding segment of road.
- It will take longer for cars to cross the intersection after waiting behind several cars stopped at a red light than if car was at front of queue.
- Green and yellow lights are both considered "Green"
- Right turns are only allowed on green lights

This model is composed of several vague abstractions of the interactive components of such a system, and is broken down into just two active components; cars, and intersections. Each intersection is defined as an object composed of a FIFO queue representing the previous segment of road, several variables denoting the light timing, and a number of functions for passing cars between intersections, checking for green lights, querying the timestamp (**TS**) of the next green light, and other supporting functions. Each car in the corridor is represented by just a series of variables denoting intended path, entry time, exit time, and relative delay (quantified by the total time spent stopped at traffic lights or blocked intersections).

To handle the events driving this system, a future event list (FEL) is implemented This list contains all events in the immediate future of each car in the corridor. Each event is defined as an object of custom type "event," which specifies a series of variables representing the actors involved (car ID and intersection ID) as well as the type of motion (stop, straight, left, right). When an event is consumed, it is determined by the event handler whether or not a new, subsequent event will be created (i.e. car passing straight through intersection to next roadway). The current time (**CT**) is also updated The FEL implemented here maintains a global priority queue of these events, and is sorted by TS to ensure no temporal discrepancies. Table 1 provides a comprehensive list of events specified within this model. Each event is handled strictly in TS order, and generates a cascade of subsequent events resulting in the car exiting the corridor. Note that no light changes are recorded here, as they are used in future event timing calculations to determine, for each car, the time at which the next event is expected to occur for a particular vehicle. This model is utilized for its high degree of abstraction, and as such, it has been determined that the events listed here accurately describe all system operations within the bounds of this model. Detail is sacrificed to ensure fast computation and a low level of complexity.

Table 1. Abstraction of Critical Events

Event	**Trigger**	**Next Event**	**Time of Next Event**
Car Stops at Red Light:	Car Enters Road & Light is Red	Car passes through intersection at	TS = nextGreen(CT)
Car Continues Straight:	Light is **Not** Red	Stop at Red Light (or) Continue at Green Light	TS = CT + (Time to Intsctn)
Car Exit Roadway:	Light is **Not** Red	No Subsequent Event is Triggered	N/A
Car Enters Corridor:	Stochastically Generated Prior to Simulation Body	Stop at Red Light (or) Continue at Green Light	TS = CT + (Time to Intsctn)

This abstraction described above provides sufficient data to very quickly simulate the system under investigation (SUI), while adhering to the most essential system-level behaviors for roadway transportation systems. Figure 2 depicts an example of this behavior, i.e. the stopping and passing of cars at intersections. In this model, it is important to note that the timestep is irregular, and because the details of the time between events are ignored, computation time can be dramatically reduced, but this comes at the cost of precision.

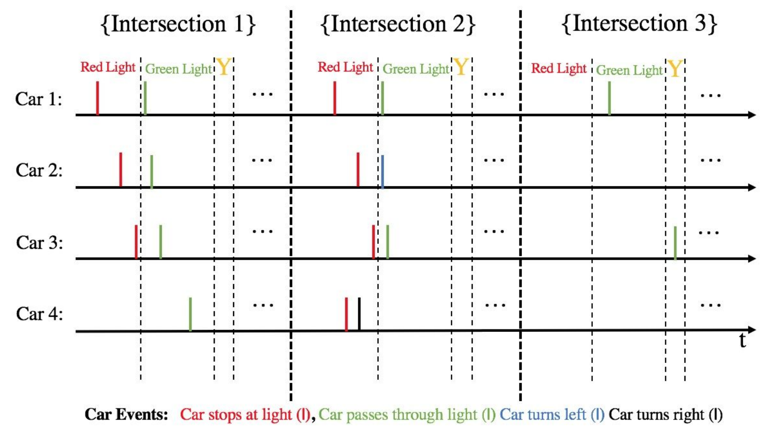
{Intersection 1}
{Intersection 2}
{Intersection 3}
Red Light
Green Light
Y
Car 1:
Car 2:
Car 3:
Car 4:
t
Car Events: Car stops at light (l), Car passes through light (l) Car turns left (l) Car turns right (l)

Figure 2. *Event-oriented abstraction of cars traversing three intersections in the simulated Peachtree corridor at various points in time.*

2. Cellular Automata Model

The cellular automata model is based in dividing the scenario, in this case Peachtree, into cells. These cells can be occupied, or unoccupied, by a single vehicle. The basic rules for vehicular traffic on a single and two lane street are taken from "Two lane traffic simulations using cellular automata" by Rickert et.al. Since Peachtree is a two-lane road, and we are investigating the Northbound traffic, cells were created by splitting the road into 16ft cells (Average found from NGSIM data).

A class was created for each section of road between intersections, this was to allow for the different rules due to the varying traffic signal times and lane numbers. Most of the sections had three lanes, so this was added to the rules of movement.

The first subset of rules is to check if cars want to change lanes. This can be prompted by a fast car being behind a slow car. There are 4 steps in the literature that need to be met to begin a transverse movement into the next lane:

T1) Is there space in front of you?

T2) Is there enough forward space in the next lane?

T3) Is there a car coming from behind in the next lane?

T4) A little element of randomness to slow down, adding stochasticity to the model.

After this subset is completed, it is imperative that the single lane rules apply, to ensure traffic flow:

S1) If maximum speed is not reached, then speed up.

S2) If going faster than the car in front of you, slow down to maintain space.

S3) An element of randomness to introduce deceleration to account for erratic behavior

S4) Take traffic lights/stops at intersections into account - yellow lights and red lights trigger slowing down or stopping of speed. These are triggered 4 cells before the end of the intersection

All of this is written as a Python 3.6 class, and the lanes are modeled as numpy arrays, where the indices of positive values are cars, and the positive values are the speeds of the cars. The simulation is looped for *t* timesteps to simulate the flow of traffic through Peachtree Street. Each timestep is 0.5 seconds. The intersection lengths were also variables of the classes, to allow for the increased or decreased number of cars possible on each path.

For the 14th street intersection, the left turn traffic signal was implemented when the straight through green signal was lit. All the traffic signal times were rounded to .5 to accommodate the timesteps, and modular arithmetic was used to calculate what color was displayed. The speed was variable for each car and the maximum speed was set to 50 mph.

3. Activity Scanning Queueing Model

The underlying code structure of the activity scanning model has a time loop that iterates through activities. Activities can be triggered by reaching a certain simulation time or by various other simulation conditions. A list of current activities is updated as the simulation progresses, which is added to dynamically as new activities are created. More specifically in the context of this project, the activities used in the activity scanning model are summarized in Table 2 and 3 below.

Table 2. B-type (Bounded) Activities

Event	**Trigger**	**Start Behavior**	**Duration**	**End Behavior**
Green Light	Vehicle is at minimum stopping distance from intersection beginning and light is green	Move vehicle into intersection, accelerate	Calculated by simulation based on vehicle's speed/acceleration and timed light changes	If the vehicle is turning, remove it from the active vehicle list. Otherwise proceed to next section and accelerate.
Red or Yellow Light	Vehicle is at minimum stopping distance from intersection beginning and light is red	Vehicles decelerate	“”	Accelerate

Car Entering	At a simulation time interval determined based on desired traffic level	New vehicles are added at an insertion point (start of road or intersection)	1 frame	Add vehicle to active vehicles list

Table 3. C-type (Conditional) Activities				
Event	**Start Condition**	**Start Behavior**	**End Condition**	**End Behavior**
Accelerate	Vehicle is not in any intersection	Vehicle has positive acceleration	Light changes, vehicle starts to tailgate, or vehicle exits	Vehicle slows down or stops, or vehicle exits
Decelerate	Vehicle is tailgating preceding vehicle	Vehicle has negative acceleration	Acceptable spacing headway has been reached or vehicle exits	Vehicle accelerates or exits

A few additional assumptions are specific to the Activity Scanning Model:

- All vehicles are cars. Trucks/motorcycles/etc. are not considered.
- All cars enter from the southmost entry point. All vehicles have a randomly selected exit point from one of the main street exits (parking lots, driveways, etc. are ignored). The probability of selection of each exit is even.
- Thirteenth Street is treated as a constant green light, since only Northbound traffic is considered. A future improvement of this model would be to build in an appropriate delay or queue to accomodate a stop sign rather than a stoplight.

Simplifications

Many simplifications will be made for each of the models as listed below. The model's fidelity and robustness could be improved by reducing these simplifications.

- Vehicle locations will be measured from the center of the front bumper of each car.
- Only Northbound traffic on Peachtree will be considered. Thus, vehicles waiting to make left turns will only wait on the light to turn green, and will not check for oncoming Southbound traffic.
- Vehicles trying to make a right on red will be ignored.
- Pedestrian traffic will be ignored.
- Vehicle width will be ignored, and we assume vehicles only occupy space in the lane where the center of the front bumper occupies.

Stochasticity

The Event Oriented Model stochastically generated cars to traverse the roadway, and stochastically determines the starting point and destination of all cars in the simulated corridor. The Cellular Automata Model has stochasticity in the velocity change behavior in each lane, as well as the random ability of a car to exit the corridor by turning. The Activity Scanning Model has all cars enter from the southmost data point, and randomly chooses any of the available exit points on main roads. Each exit is equally likely to be chosen.

INPUT DATA COLLECTION AND ANALYSIS

Cellular Automata: The interarrival time (about 23 timesteps or 11.5 seconds) was found through an analysis of the ½ the NGSIM data of cars entering 10th street through the 101, 102, and 123 lanes. To account for changing flow rates, a random number generator was used to select the time at which a car would enter the corridor at 10th street, with a random speed between 0 and 50.

Activity Scanning: Analysis of the NGSIM data, University of Idaho study, and other provided analyses files provided many important parameters to set up the simulation as outlined below. In the model, these values are set in the randomParams() function, which would be modified with different data sets. The simulation itself is created and run by generateSim(), which can be reused with different data sets.

- Average stopping distances (which compensate for driver reaction time) are listed below. If a vehicle is less than the stopping distance away from the edge of an intersection when a light turns yellow, the car will not stop in time, and instead it proceeds through the intersection.
    - 30 mph = 75 feet
    - 20 mph = 40 feet
    - 10 mph = 30 feet
- Cars are assumed to not drive more than 5 mph above the speed limit (no more than 56 ft/sec), and acceleration is limited to no more than 12.2 ft/s^2.
- Stoplight durations were extracted from the provided signal timing spreadsheet.
- Intersection and section lengths are outlined in Table 4 and 5 below.

Table 4. Intersection Lengths

Intersection	Length (feet)
1	99.645

2	129.795
3	73.815
4	66.979
5	117.319

Table 5. Section Lengths	
Section	Length (feet)
1	127.391
2	441.437
3	412.070
4	353.727
5	343.922
6	11.730

SIMULATION MODELS

Code for all three models can be found in their respective folders in the team's Github. Division of work was as follows:

1. Event Oriented: Blaine Costello
2. Cellular Automata: Lakshmi Raju
3. Activity Scanning: Caitlyn Caggia

with all team members contributing to the written report.

VERIFICATION AND VALIDATION

Event Oriented Model

Initial validation was provided by first examining the input datastream to get a vague idea of the system behavior, then running a single car through the corridor. To ensure that each model is capable of running larger datasets, the number of cars passing through the corridor (N) within the specified time window (T) was increased while the resulting data was manually verified and tuned to be "within reason." Qualitatively, each of the models behaves properly, and at this stage, is compared to measured data for verification.

The first 100 seconds, of the runs with this model were excluded from consideration, as this was designated to be the 'warmup' or 'loading' period. The remaining 900 seconds of was compared to the NGSIM data by computing average cumulative time stopped among all cars (i.e. how long in total red lights delayed travel for each car). It is noted that for extremely large numbers of cars in the roadway, this delay grows dramatically. The number of cars and the interarrival time of these cars was determined empirically using the provided data. A small number of cars (15 cars) was used to provide verification and a stochastically generated interarrival time of uniformly distributed between 7 and 13 seconds. Though there was significant error (up to 34%) between the model and the measured data, the predicted delay was within one order of magnitude, and was within what we considered to be a reasonable window given the high level of abstraction and fast execution time associated with this methodology.

Cellular Automata Model

A warm up period of 200 timesteps was used, and data during that time was not collected. When the input cars was varied from the NGSIM data to test the model, appropriate reactions were shown. When the model was inundated with cars with very short interarrival times (between .5 and 2 seconds), the cars backed up and caused errors as not enough cells were available. This served to validate the model, along with the case of very few cars (4) interspersed with enough space (interarrival of 30 seconds) to just stop at lights and go. All runs were completed for 2000 time steps (for high traffic and NGSIM) and 400 timesteps (4 car model).

Activity Scanning Model

Initially, the simulation was tested for simple cases, including:

- One car driving through all green lights and exiting at the northbound Peachtree exit
- One car driving through all green lights and exiting randomly
- One car driving through varied timed lights and exiting randomly
- 10 cars driving through all green lights without collisions and exiting randomly

To improve numerical issues with using floating point numbers, rather than using time, frames were used to progress the simulation. One frame matched the frame length of the data collection of the NGSIM data set (0.1 seconds). A warm up period of 10 frames was used.

As with the other models, a main metric of success was the average amount of time spent stopping for each vehicle. As this would be a driving motivation for actual traffic analysis and simulation, the team considered this our most significant metric.

Some errors occur in very high traffic situations, as one of the critical underlying assumptions was that cars do not change lanes during the simulation.

DESIGN OF EXPERIMENTS AND OUTPUT ANALYSIS

The purpose of this simulator is to predict travel time along the specified corridor. One major component which leads to an increase in travel time is time spent stopped at traffic lights or blocked intersections. As such, this parameter is used not only for verification and validation, but also is analyzed for varied levels of traffic loading (N).

Additionally, time complexity was examined among all models explored in this work. This was to compare the efficiency of the three models, as each model has a few separate assumptions and designs. This was decided to be the best parameter for comparison of all three methodologies as implemented here. For a standardized run of 1000 cars in a timeframe of 2000 seconds, each code was executed on the same computer to ensure scientific precision, with the results listed in Table 6.

As expected, the event oriented model performs fastest, but is less accurate than the other two. This further validates the efficacy of implementation of these models using Python.

Table 6. Efficiency Between Models

Model	**Run Time**
Event Oriented	0.3731474876403809
Cellular Automata	3.8632397651672363
Activity Scanning	0.5160350799560547