Automatic Intersection Management and Control

Ismayil Hasanov, Yubo Tsai, Han Wang
CSCI 6341, Continuous Algorithms
Spring 2017

Abstract

In a world with cars transforming to self-driving ones, would our current intersection management systems with timed traffic lights be an efficient method of traffic management? This project studies the feasibility of the automatic intersection management and control by simulating an intersection of two perpendicular roads, each consisting of two number of lanes, where there are no general traffic lights but individual control signals sent to the cars (whether they can move next or not).

The project involves a simulation, queue control and collision detection mechanisms. The cars will be populating the lanes based on some probability distribution, and the signal sent to the cars will be based on the result of the received signal from the cars with regards to the turning direction and the collision detection control at the intersection. The analysis of the results might suggest a real-world application to traffic control centers around the world.

1. Introduction

Models from queueing theory are now widely recognized as useful aids toward understanding and controlling congestion while maintaining throughput in many systems. These include computer systems, voice and data telecommunications, vehicular traffic flow, emergency public service, and industrial customer shops, production lines, and flexible manufacturing systems. The past work has been done to describe traffic behavior at intersections based on queueing theory [1] [2] [3].

The project takes the idea of a single-server queue and applies it to the context of intersection of roads and arriving cars. Based on the arrival time of cars, the system will decide who should depart next. We are treating 4 lanes as a single arrival queue where the service time can be considered as the time interval from the moment car enters the intersection box to the time unit when the same car leaves the intersection box. We assumed that the cars are not allowed to U-turn. It means that the choice of direction is limited to left, straight or right.

2. Model Design and Strategy

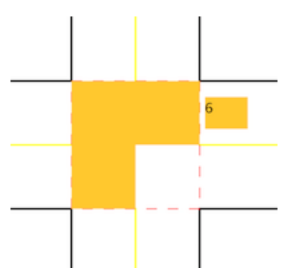
The system lets the cars depart on a basis of first-come-first-serve strategy. For the arrival times and directions, we have utilized the exponential distribution. Similarly, the service times are using the exponential distribution. However, the arrival and service rates could be modified, as fit.

The main difference from the queue control simulation we have seen in the class, is the idea of simultaneous "green lights" for the cars waiting in the front of lanes for their turn.

If we divide the intersection box into quadrants and label them from 0 to 3 starting from the bottom left and going in the clockwise order, we will get the below table which characterizes the quadrants occupied when the car from a lane (columns) goes in a direction (rows):

direction\lane	0	1	2	3
Left	0 2 3	1 0 3	2 1 0	3 2 1
Straight	0 3	1 0	2 1	3 2
Right	0	1	2	3

For example, "the car (id:6) at lane 2 turns left" can be seen in the following figure as occupying quadrants 2,1 and 0:



Furthermore, to make the traffic control realistic, we defined the service time as time interval the car spent inside the intersection box.

3. Project and Challenges

The project consisted of two parts: the model implementation (AIMC.java) and GUI simulation (AIMCGUI.java). In the model implementation, a single event is handled in nextStep() method. Based on the event type, we let the model handle arrival or departure, separately. Moreover, a departure event fires a trigger for the GUI to respond. Then, the statistics collecting method is called at the end.

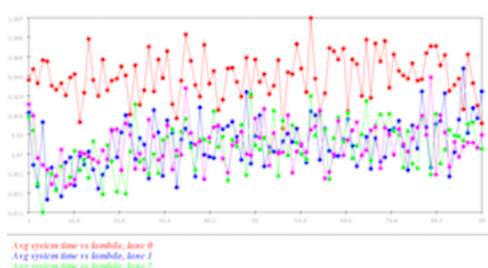
The essence of our model is allowTraffic() method, where the algorithm checks what other cars are allowed to depart at the same time as the main departure event. The subtle implementation trick is to check, if the departure time is need to be changed for those cars which are allowed to depart simultaneously.

The biggest challenge we encountered was modifying the event list priority queue when we had more than one car simultaneously departing. Therefore, we added car id to the event class to be able to remove these cars.

Another challenge we faced was making sure the GUI and the model are communicating with each other. We have decided to use observer pattern for this task.

4. Results

For the statistics purposes, we have gathered the total system time for each lane. For 10000 departures, we have gathered these values and found the average time spent on each lane. The following plot shows the average time spent per lane for arrival rates in the range of [1,100]:



We have to be careful when interpreting the results due to sources of error in next-event simulations: 1) errors and approximations in the model itself. The cars arrive at the intersection in groups (if you treat all lanes together) but are modeled as arriving individually according to a Poisson process; 2) initialization bias, where we have no cars in the beginning of the simulation; 3) model implementation errors.

5. Future Work

This model could be extended in many ways. One way is to improve the car departures by switching to real-time simulation for those events. Since the real-time collision prevention is more important in practice, this simulation is for academic purposes only and should not be taken as a face value and applied to real roads.

References

[1] D. Heidemann, "Queue length and waiting time distributions at priority intersections," *Transportation Research-B*, vol. 25, 1991, pp. 163–174.

[2] D. Heidemann, "Queueing at unsignalized intersections," *Transportation Research-B*, vol. 31, 1997, pp. 239–263.

[3] A. C. Soh, M. H. Marhaban, M. Khalid and R. Yusof, "Modelling and Optimisation of a Traffic Intersection Based on Queue Theory and Markov Decision Control Methods," *First Asia International Conference on Modelling & Simulation (AMS'07)*, Phuket, 2007, pp. 478-483.