Transportation System Simulation

Anita Zakrzewska
Borja Zarco
Marat Dukhan
Yi Wei Cheng

Project Plan

Introduction

Queuing models have been applied extensively to traffic flow simulations to improve efficiency of many aspects of current traffic infrastructures. *Cheah and Smith* (*1994*) employed a state dependent M/G/C/C queuing model to model the behavior of pedestrian traffic flows. Uninterrupted traffic flow has been modeled with M/M/1 and M/G/1 queuing models by *Heidemann* (*1996*). To include for the effects of congestion, *Jain and Smith* (*1997*) modified the model of *Cheah and Smith* (*1994*) and modeled the vehicular traveling speed as a decreasing function of number of cars in the system. In the recent decade, behavior of non-stationary traffic flow has been modeled as the transient dynamics of M/M/1 queue model by *Heidemann* (*2001*). Many of these early queue models simulate traffic flows with the assumption zero incidents. Recently, *Baykal-Gürsoy et al* (*2009*) explored the impacts of incidents on traffic flows with a steady state M/M/C queuing model.

Priority queue is an abstract data structure that contains a set of elements each with an associated value or priority. The two basic operations on a priority queue are enqueue and dequeue. Enqueue inserts an element into the priority queue while dequeue removes the lowest (or highest) priority element in the queue. In the recent decades, many priority queue implementations have been developed. The simple linear list implementation has complexity of O(n) while the two list method reduces the complexity to O($n^{0.5}$). The splay tree (*Sleator and Tarjan*, *1983*; *Tarjan and Sleator*, 1985), pagoda (*Francon et a*l, *1978*), skew heaps (*Sleator and Tarjan*, *1983*; *Sleator and Tarjan*, *1985*), and binomial (*Vuillemin*, *1978*) all have complexity of O(log n). *Brown* (*1988*) developed a calendar queue for the simulation event set problem that is experimentally justified to have complexity of O(1). Recently, *Tang et al* (*2005*) developed a ladder queue that is theoretically justified to have O(1) amortized access time complexity.

Priority queue models also rely extensively on random number generation for simulation of events. Many random number generators have been proposed (Brent 1994, Matsumoto and Nishimura 1998, Marsaglia 2003) with provably good statistical properties. However, the use of Mersenne Twister (Matsumoto and Nishimura 1998) has become almost standard in numerical simulations: this random number generator is used by default in MATLAB, R, Intel MKL, AMD

ACML because it combines good speed with excellent statistical properties. A variety of methods exists for transforming uniform variates into variates from other distributions. One of the simplest methods is inversion of cumulative distribution function, but there calculating inverse CDF may be prohibitively expensive for many distributions. For example, the CDF of Student's t distribution is given by incomplete beta function which is typically calculated by numerical intergration, and multiple evaluations of this function are required to calculate its inverse. However, more efficient sampling techniques were proposed for normal and gamma variates, of which the methods of Marsaglia and Tsang (2000) provide the best combination of speed and statistical accuracy.

In this project, we will develop a priority queue model to investigate traffic routing plans that will efficiently route vehicular traffics from parking lots near the Georgia Dome, Atlanta, GA, to the major interstates (I-20, I75/85). The plan should minimize the average delay for a football fan to reach the interstate after the game.

Assumptions

Our model relies on the following assumptions:

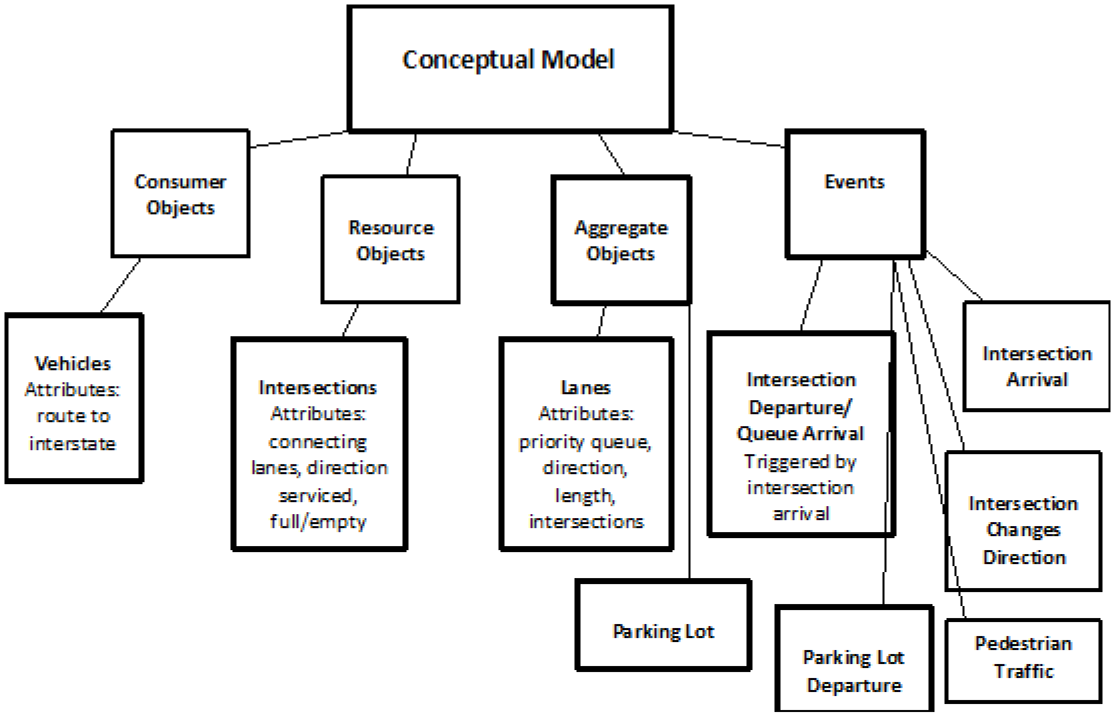
1. Once individuals reach the interstate they can travel freely to the final destination. This assumption exists in order to only model the traffic flow from parking lots to interstates. Thus, we do not model any traffic jams that may occur on the interstate, which could stop vehicles from entering an interstate.
2. All drivers initially try to take the shortest route from the parking lot to the interstate. This is a simplification because not all drivers will know the shortest route. However, since drivers will tend to take short routes to their destination, this is a reasonable way of approximating the routes that cars will take.
3. No road accidents.
4. The occupancy of a parking lot is a random variable. Although the parking lots tend to be quite full during game days, they may not be completely filled up. Thus, we model their occupancy with a random variable.
5. External traffic follows a random process. We have no knowledge of the destinations of external traffic. Thus, the directions they proceed at intersection follows a random process. This simplification allows us to avoid creating a specific path for each vehicle in external traffic.
6. All turns are allowed and possible for external traffic on intersections as long as the directions of lanes connecting to an intersection allow for such a turn. This will allow us to more naturally model the flow of external traffic, which may be traveling in any direction (following a random process).
7. Pedestrian traffic is only allowed on pedestrian crossings, which occur at intersections. Pedestrian traffic on controlled pedestrian crossings (at an intersection) is modelled implicitly. Pedestrians are assumed to cross at the same time as vehicles traveling in the same direction are entering and departing from the intersection. Since these events occur concurrently, such pedestrian crossings are modelled implicitly.
8. Each intersection has a person directing the flow of traffic, overriding any stop signs or

traffic lights. This assumption allows us to not model stop signs and to have full control over which direction of traffic an intersection is servicing.

9. Vehicles are travelling at a uniform speed.

Conceptual Model

1. The consumer objects in our model are the vehicles passing through the road network. Vehicles contain an attribute that states whether it is a vehicle traveling from a parking lot to the highway or whether it is external traffic. Vehicles traveling from a parking lot to a highway will follow the road network to the specified highway by the shortest path (or shortest path with minimum number of intersections). Through traffic vehicles will follow a random path.
2. The resource objects in our model are the intersections. A finite number of vehicles may occupy an intersection at a time. This number of vehicles depends on the number of lanes in the roads meeting the intersection. For example, if both roads crossing the intersection are two lane roads (one lane per direction), then at most two cars may occupy the intersection at a time.
3. The aggregate objects in our model are the Roads. Each Road has Lane objects and each Lane object contains a list of Vehicles in it.
4. We are modeling only major roads for routing cars from parking lots to highways, but external traffic may be modeled on minor roads.
5. The following events can occur and change the state of the system:
    a. A vehicle emerges from a parking lot (modeled using the random number generator). This is the event that feeds most of the vehicles into the road network being modeled.
    b. An intersection changes the direction it is servicing.
    c. A vehicle leaves a road and enters an intersection. The preconditions for this event are that the intersection must be servicing the direction that the vehicle is traveling and the intersection and road to be entered must not be full.
    d. A vehicle departs from an intersection and enters a new road.
6. Pedestrian crossings at intersections are modeled implicitly. Pedestrians who wish to cross North/South are assumed to cross when the intersection is servicing traffic from its North/South lanes (going straight, not turning).



Routing Strategies

We will investigate the efficiency of two traffic routing plans. In the first plan, the shortest paths between a specific parking lot to the respective major interstate entrances are delineated. Cars leaving this specific parking lot will be directed along the shortest path depending on their destination interstate. The process is repeated for all the other parking lots. The second routing plan is an extension of the first routing plan. In addition to directing each car along the shortest path, we impose the second condition that this shortest path must also contain the least number of intersections. Similarly, these shortest paths will be delineated first for all parking lots. The better plan of the two should minimize the average delay for a football fan to reach the interstate after the game. For each plan, the average delay can be evaluated as the total time for all the cars from the parking lots to reach the interstates normalized by the total number of cars.

The traffic routing plans will be enforced by police officers stationed at every intersection. As such each intersection and the assigned police officer represents a server in our model. The service time of the server is determined by the strategy adopted by the police officer at the intersection. We suggest three strategies for police action at the intersections. (1) The police will direct the traffic such that the first vehicle to arrive at the intersection will be cleared first, a.k.a first in-first out protocol. (2) The police will give each lane a specific amount of time to clear the traffic, a.k.a constant time protocol. (3) In this last strategy, the police will always seek to clear the lane that has the highest vehicular density, a.k.a high density protocol.

The Data Set

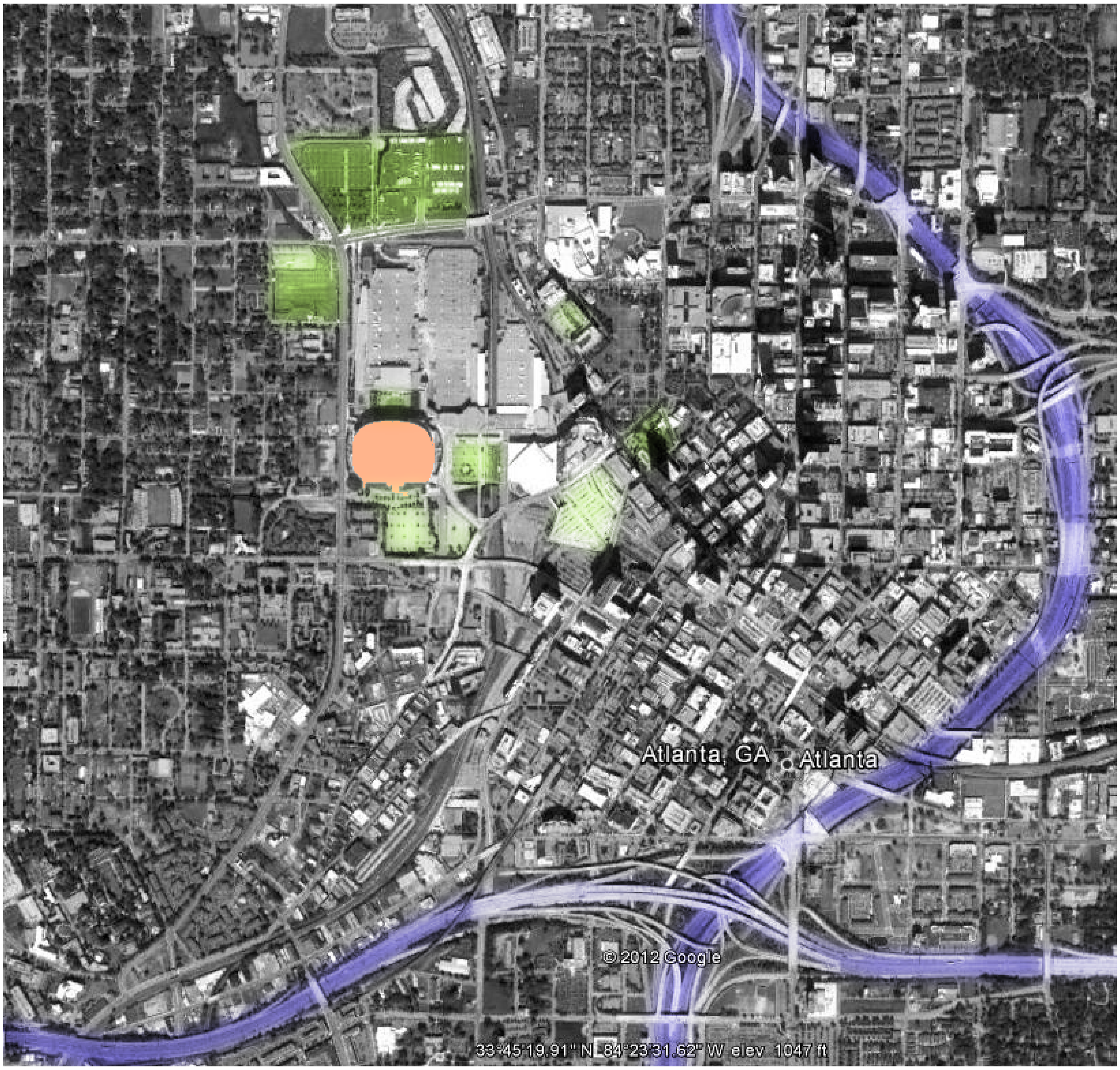
Our transportation model relies on information about road network topology and positions of parking lots. The following information is to be collected:

1. Road segments
    a. Traffic direction
    b. Number of lanes in each direction
    c. Length of segment
    d. The two bounding roads
    e. The two intersections on either end
2. Intersections
    a. The latitude
    b. the longitude
    c. Whether the intersection is a destination
3. Parking lots
    a. Capacity
    b. The intersection at the exit
4. The shortest paths from each parking lot to each destination

We collected the above information mostly from online maps[1], and also use information on parking lot from Georgia Dome website[2].

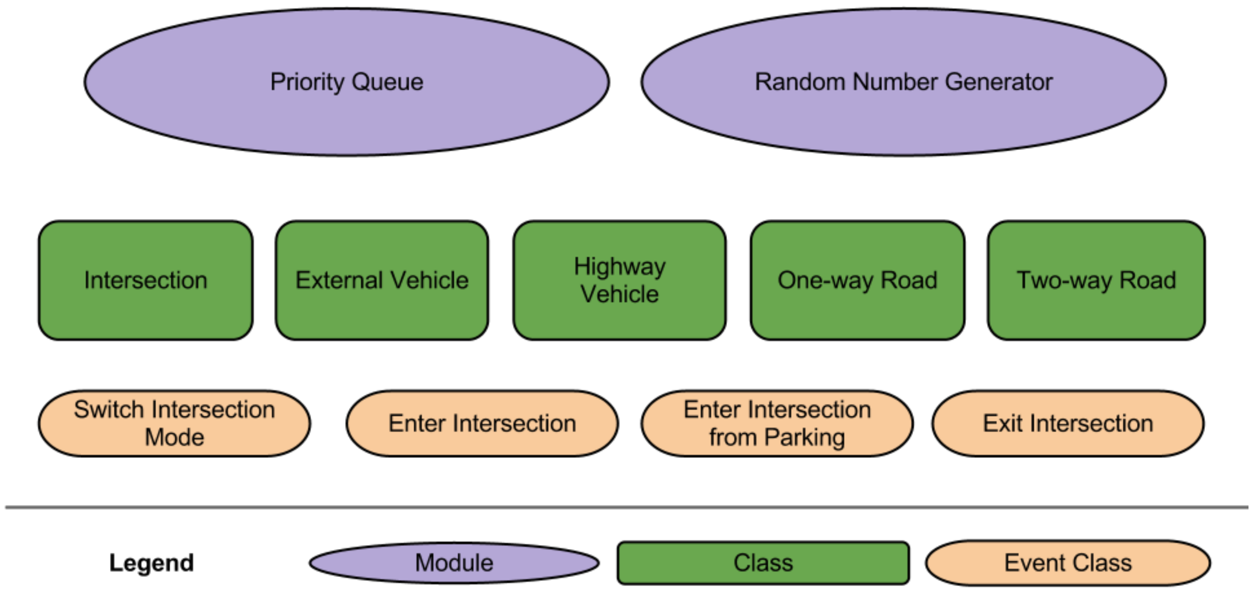
[1] http://toolserver.org/~geohack/geohack.php?pagename=Georgia_Dome&params=33_45_27_N_84_24_3_W_type:landmark_scale:2000

[2] http://www.gadome.com/guest/directions/Parking.aspx



Software Modules

Software Modules



1. Priority queue (List, Splay, Calendar, and Ladder implementations)
2. Random number generator based on MT19937 (Matsumoto and Nishimura, 1998) and Marsaglia-Tsang (2000) algorithms for normal and gamma variates.
3. Modeled objects
   a. Intersection. A shared resource which is used for driving between the connected road segments. Pedestrian traffic at intersection is modelled implicitly. There is a public intersection interface that is accessed by all other classes. There is an intersection class extending this interface.
   b. Road. Input/output in an intersection. The road class has an internal lane class. Each road has a list of lanes. Each lane has a priority queue associated with it. A road stores information about its length and direction. Have finite capacity. There is an abstract Road class and two non abstract road classes: for one way roads and for two way roads.
   c. Parking lot. A source of cars which leave it. Each parking lot is connected to an intersection on its exit. Parking lots have finite capacity and random occupancy.
   d. Path. This is a list of roads from a parking lot to a destination. Each vehicle has a path.
   e. Vehicle. Vehicles are removed from Parking Lots into the adjacent Intersection. Each non-external Vehicle has a destination intersection and a path of roads leading from the Parking Lot to the destination. Vehicles move along the Roads (through Intersections) in the path.
4. Modeled events
   a. Intersection arrival. New Vehicle arrives at Intersection from Road. Remove the vehicle from its current road and create and enqueue an Intersection departure

event. If the Intersection is a destination, then this is also the Interstate arrival event.

b. Intersection departure. Vehicle leaves Intersection and is added to a lane in the next road of its path. Notifies the exiting road of the departure, which may cause an intersection arrival event to be enqueued (if there are available vehicles in that road and the intersection mode matches).
c. Intersection direction change. Switch the serving direction of the Intersection. After the direction of the Intersection is changed, the roads of the intersection are notified. This causes each road to check if it has any vehicles available at that time that can enter the Intersection in its current mode (if the Vehicle's current and next road are allowed in the current mode of the Intersection). If the Intersection is not idle, another Intersection direction change event is created and enqueued.
d. Parking lot departure. Vehicle leaves Parking Lot and enters the connecting Intersection and then the next Road in its path. Dequeue Parking Lot and determine Vehicle's destination interstate (randomly) and path and set the Vehicle's starting time. Once the Vehicle is added to its first road, if the mode of the intersection still allows for Vehicles to leave the Parking Lot, another Parking Lot departure event is created and enqueued.
e. Pedestrian traffic. Used at Intersection, Vehicles wait while pedestrians cross the road. Pedestrians cross when the Intersection mode matches (Vehicles are travelling in the same direction as pedestrians), so this is modelled implicitly by Intersection direction changes.

Bibliography

Baykal-Gürsoy, M. and Xiao, W. and Ozbay, K. (2009) “Modeling traffic flow interrupted by incidents”, European journal of operational research, 195:127–138.

Brent, R.P. (1994) “On the periods of generalized Fibonacci recurrences”, Mathematics of Computation, 63:389–402.

BROWN, R. 1988. Calendar queues: A fast *O*(1) priority queue implementation for the simulation event set problem. *Commun. ACM 31*, 10 (Oct.), 1220–1227.

Cheah, J.Y. and Smith J.M., Generalized M/G/C/C state dependent queuing models and pedestrian traffic flows, *Queueing Systems*, 15, 365-385 (1994).

Francon. J., Viennot, G., and Vuillemin, J. Description and analysis of an efficient priority queue representation. In Proceedings of the 19th Annual Symposium on Foundations of Computer Science (Ann Arbor, Mich.. Oct. 16-18). IEEE, Piscataway, N.J.. 1978, pp. Z-7.

Heidemann, D., A queueing theory approach to speed-flow-density relationships, *Proc. Of the 13th International Symposium on Transportation and Traffic Theory,*France, (July 1996).

Heidemann, D., A queueing theory model of nonstationary traffic flow. *Transportation Science*, Vol. 35, No.4, 405-412 (2001).

Jain, R. and Smith, J. M., Modeling Vehicular traffic flow using M/G/C/C state dependent queueing models, *Transportation Science*, 31, No. 4, 324-336 (1997).

Marsaglia, G., and W. W. Tsang (2000) “The ziggurat method for generating random variables”, Journal of Statistical Software 5:1–7.
Marsaglia, G., and W.W. Tsang (2000) “A simple method for generating gamma variables“, ACM Transactions on Mathematical Software 26:363–372.
Marsaglia, G. (2003) “Xorshift RNGs”, Journal of Statistical Software, 8:1–6.
Matsumoto, M., and T. Nishimura (1998) “Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator”, ACM Transactions on Modeling and Computer Simulation 8:3–30.
Sleator, D.D.. and Tarjan, R.E. Self-adjusting binary trees. In Proceedings of the ACM SIGACT Symposium on Theory of Computing (Boston, Mass., Apr. 25-27). ACM, New York, 1983, pp. 235-245.
Sleator, D.D., and Tarjan. R.E. Self adjusting heaps. SIAM]. Comput.
Tarjan, R.E., and Sleator, D.D. Self-adjusting binary search trees. J. ACM 32, 3 (July 1985). 652-886.
Tang, W. T and Goh, R . S. M and Thng, I. L (2005). Ladder queue: An o(1) priority queue structure for large-scale discrete event simulation. ACM Transactions on Modeling and Computer Simulation. Vol 15, No. 3, 175–204.
Vuillemin, J. A data structure for manipulating priority queues. Commun. ACM 21,4 (Apr. 1978), 309-315.
Zhang, H. and Shi, D. (2010) “Explicit solution for M/M/1 preemptive priority queue”, International Journal of Information and Management Sciences, 21:197–208.