Proceedings of the 1994 Summer Computer Simulation Conference
ISBN: 1-56555-029-3

A TRAFFIC SIMULATION FOR MID-MANHATTAN WITH MODEL-FREE ADAPTIVE SIGNAL CONTROL

Daniel C. Chin and Richard H. Smith
Johns Hopkins University, Applied Physics Laboratory
Laurel, MD 20723-6099

ABSTRACT

A computer simulation of a moderate-sized traffic network (nine intersections) in high density sectors was developed for a study to demonstrate the feasibility of a real time traffic-adaptive control algorithm. The simulation is based on a real scenario of part of the mid-Manhattan Central Business District in New York. In this paper, the traffic flow simulation provides the system wait time used in the cost function that is optimized for training the weights of a neural network (NN) controller for the light timings. The neural network controls the split times of all signal lights at the intersections involved in the simulation study. We are controlling the cycle-by-cycle split times, based on averaged queue sizes, that give the entire traffic system a minimum total wait time. The evaluation of traffic simulation software results is also reported.

INTRODUCTION

A versatile and interactive computer simulation of a moderate-sized traffic network in a high-density metropolitan sector was developed to demonstrate the feasibility of a real-time traffic-adaptive signal control algorithm. The purpose of the simulation was to represent realistic traffic behavior under moderate to heavy demand conditions for the mid-Manhattan Central Business District. A primary goal was to develop a traffic model for the network that embodied complex link and intersection interactions and delivered realistic cycle-to-cycle traffic queue sizes and flow levels (volumes) without resorting to the extreme detail of a per-vehicle simulation. Such a microscopic level of modeling is incorporated currently in popular traffic simulation packages such as TRAF-NETSIM [5]. Although a microscopic degree of simulation will have the greatest opportunity to model specific traffic anomalies, such as the effect of parked cars, its extreme generality renders it very cumbersome for use in an interactive mode with a real-time controller. The simulation developed in this paper is much easier to implement in such an interactive mode and delivers traffic performance that closely replicates the general traffic volume patterns actually observed through on-the-site traffic monitoring.

The area simulated in this paper represents the major portion of the mid-Manhattan sector treated in Rathi [2], in which a TRAF-NETSIM [5] simulation was employed to test a control scheme. The area simulated runs between 55th and 57th Streets (north and south) and from Avenue of Americas (6th Ave.) to Madison Avenue (east and west) and therefore includes 5th Avenue as the central artery.

The simulation was used conveniently in an interactive mode with a general adaptive control algorithm capable of operating with a model-free assumption. The algorithm is described in Spall [3] and Spall and Cristion [4] and was able to achieve significantly improved traffic signal split times for the nine intersections in the above simulated network.

THE TRAFFIC SIMULATION SOFTWARE

The traffic simulation software provides realistic values of the average queue sizes and the total wait time in order for a traffic control algorithm to perform a simulated control. The simulation software is able to model several characteristics of the stochastic traffic flow and human psychological behavior while, at the same time, being simple to use. It models both nonlinear traffic dynamics and general fluctuations of traffic parameters. Although the simulation outputs are changing discretely from one time point to the next, the underlying average traffic volumes settle at relatively stable values that reflect the average values of the real traffic system we are attempting to simulate (see next section).

This software is a queuing network simulation which implements individual rules and conditions at each intersection within the traffic system. There are also general rules and conditions for the entire traffic system, there is a built-in database to direct the traffic at each queue, and there is a controlling algorithm to monitor the traffic condition. We will subdivide this section into subsections along these lines, as follows, to describe the traffic simulation software:

General Rules and Conditions

As typically applied in transportation texts (e.g., Morlok[1]), the arrival rate for an input queue satisfies a Poisson distribution. Furthermore, we assume that the cars arrive at each input queue at a constant average rate. Normally, the arrival and departure of cars in a traffic system are in equilibrium, except perhaps for cars parked for office work. We additionally simulate the egress of such cars from a parking garage between, say, 4:00 and 5:00 p.m. These cars are also modeled to enter internal queues with a Poisson distribution and constant average arrival rate.

Another important item to be modeled here is the vehicular turning rate at intersections. This is one of the most complicated items to model due to its dependency on human reactions to the traffic conditions. Modeling of this item introduces nonlinear effects into the system by making the assumption that the probability of a turn from one artery onto another depends on the current state of the system. Referring to Figure 1, we will assume that congested conditions on a downstream artery will tend to increase turning rates onto side streets and will also decrease turning rates from side streets onto the main artery. Let C_a be the capacity of the downstream artery, A be the current traffic in a queue, and μ be the turning rate. If $A*2>C_a$ then the turning rate from the main artery onto the side street will be adjusted linearly as

$$\mu=\mu+(A*2-C_a)/C_a.$$

At the same time, the turning rate from the side street onto the main artery will be adjusted negatively, that is, decreased linearly toward zero from its original value. If a car can turn in two directions (left or right) at an intersection, then the turning rate will be half as much as that used for a single-direction turn, and the adjustment will be less, too. The right-turn-on-red-after-stop will be allowed also; the software will assume that 5% of the cars in a queue will make this type of turn.

Rules for the number of cars passing through an intersection are central to the simulation. This number is limited by the length of the green light, by the size of intersection, by the capacities of the downstream arteries, and by the sizes of the downstream queues. To determine this number, the software will first determine the maximum number of cars that could possibly pass through the intersection. Then according to the turning rate, it will divide this maximum into the number of cars that will turn and the number of cars that will go straight. Second, the software will check the capacity of the downstream queues. If the downstream queues cannot hold the progressing cars, then excess cars will remain in their original queues and represent a traffic jam condition. The right-turn-on-red-after-stop is also limited by the downstream queue size.

Finally, we must implement rules for the depletion of a queue. Let l be the depth of cars in the queue at a traffic light and let t_d be the depletion time. As in McShane [6], the standard equation for time of depletion is $t_d = 3.7 + 2.1 * (l-1)$, where the units are in seconds. The value 3.7 is the average lost time per green phase, and 2.1 is the average reaction time starting at the driver's perception of the movement of the preceding car. When the software applies this equation, it also assumed that cars are evenly distributed on all lanes. If the depletion time required for the queue is greater than the length of time of the green light, then not all of the cars in a queue will be depleted. In this case, the number of cars passing through the intersection during the green phase will be just the number of cars depleted using the available green time. The incoming traffic is also subject to this depletion restriction. The actual time for traffic passing through the intersection is the length of the green phase minus the depletion time.

There are other miscellaneous assumptions made in the traffic simulation software. The signal lights for all intersections within the controlled traffic system have identical total cycle timing and the N-S artery green lights are turned on simultaneously. As mentioned in Rathi [2], the prior traffic control system sets the proportion of the green phase to total cycle time on the N-S arterial to be in the range between 0.55 and 0.6 and all the intersections have 90-second cycles. Therefore, we used the 0.55 setting as our initial condition and 90 second cycles for the Manhattan simulation study.

Built-in Database

The traffic simulation software organizes the traffic system based on a numbering of the intersections. As in the Manhattan simulation in Figure 1, the controlled signal lights are numbered from 1 to 9, the feeding intersections are numbered from 10 to 16, and the output intersections have no numbers assigned to them because they are irrelevant to the traffic control algorithm. The streets between the numbered intersections contain the relevant queues, and they are assigned indices based on the intersection numbers at each end and the direction of traffic flow. For example, 0102 refers to the queue that develops from the traffic flowing from intersection #01 to intersection #02.

The information contained in the built-in database is as follows:

1. A queue sequence that pertains to the sequence of traffic flow.

   Queues are divided into two sets, and one set follows the other. The first set of queues should consist of all those queues facing the green light at the beginning of the light cycle. The second set of queues should be all of those queues facing red lights at this same time. The preferred order of the queues is according to the traffic flow sequence, i.e., from upstream to downstream.
2. The capacity of queues, i.e., the average number of cars the street can hold between intersections.
3. The downstream links, which include
   a. access queues
   b. right turn queues,
   c. left turn queues.

   The exit queues are assigned large numbers and the no-turn queues are assigned -1.
4. The number of the intersection that the queue faces.
5. Throughput factors, i.e., the average total number of cars from the facing queue that can go through an intersection during a light cycle.
6. The number of lanes in various queues; a bus lane is counted as a half-lane.

Traffic Simulation Control Sequence

The simulation controller starts from the first set of queues given in the database and sequences through them consecutively. It assumes that the first set

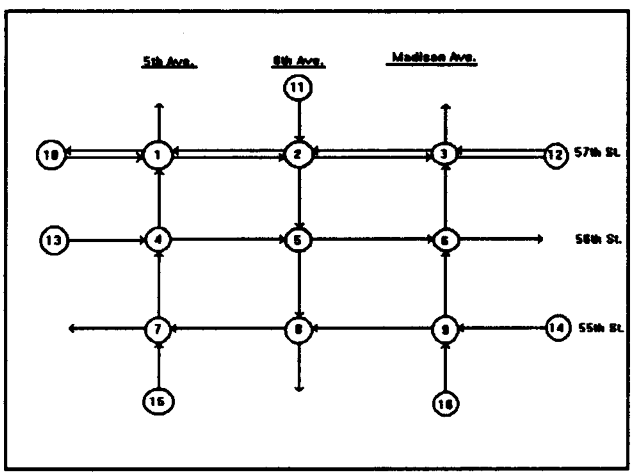

Figure 1: Traffic Simulation Control Area (Mid-Manhattan)

of queues are facing green and the second set of queues are facing red at the beginning of the light cycle. Then, the software iterates again through the queues using the reverse assumptions.

The algorithm simulating the action for the queues facing green is as follows:

Step 1: Estimate the number of cars arriving and add this to its arrival queue.
Step 2: Compute the depletion time and the delay cost for the cars in the queue facing the green light.
Step 3: Implement the turning rates as mentioned in the last subsection.
Step 4: Compute the number of cars passing through the green light, the number of turning cars, and the number of straight-through cars.
Step 5: Check the downstream queues and only allow the cars to proceed to the downstream queues if the downstream queues can hold them.
Step 6: Compute the wait times for the cars left in each of the queues, and add these to the total accumulated wait time.

The items to be computed for the queues facing red are the number of cars arriving within the red phase, the wait time, and the number of cars making a right-turn-after-stop. The sequence is not important for the red light phase. Again, the number of cars making the right-turn-after-stop is limited as in the other turn logic.

The Input/output of The Traffic Simulation Software

The inputs of the simulation software are the time of day and the timings for all the lights at the first set of queues faced. In the Manhattan simulation, the timings of the green light phase are those pertaining to the N-S direction of traffic flow. The outputs of the traffic simulation software are the total wait time, in both root-sum-square and summation form, and the average queue sizes for all of the queues.

EVALUATION OF THE TRAFFIC SIMULATION SOFTWARE

The previous section mentioned that the output of the traffic simulation software will vary from time to time to reflect the stochastic nature of the traffic flow. To verify the simulation software, we ran the software for many sample cases. Each time, we used the same arrival rates and the same setup, but we used a different seed to initialize the Poisson pseudo-random generator. Then, we compared our results against the real data of the traffic system we are simulating as recorded in Rathi[2], Table 6. Two sample results are shown in Tables 1 and 2. As mentioned in the introduction, the traffic system in which we are interested is the high traffic density sector in mid-Manhattan, i.e., the nine intersections in the area bounded on the north by 57th Street, on the south by 55th Street, on the west by 6th Ave., and on the east by Madison Ave. as show in Figure 1. It consists of a two-way and two one-way east-west side streets and three north-south main arteries.

Rathi [2] studied the 5th Ave. traffic and controlled the side street signal lights, according to the traffic conditions on the side streets, to relieve the traffic jam conditions. Rathi reported the volume counts along 5th Ave. in both rush hour and non-rush hour. Those volume counts are used for comparison to our simulation results, and both are tabulated in Tables 1 and 2. Table 1 uses the non-rush hour counts from 12:40 p.m. to 2:20 p.m.; Table 2 uses the rush hour counts from 4:40 p.m. to 6:20 p.m.

There are five sample cases shown in both Tables 1 and 2 which result from five realization of the simulation for selected fixed average Poisson arrival rates at all input queues. The entries under "QUEUES" are the streets between the intersections indicated by the first two-digit number and last two-digit number. The first two-digit intersection is the traffic starting location and the last two-digit intersection is the traffic facing intersection. The entries under each case number are the average queue sizes obtained over thirty-minute periods. The entries under "AVERAGE" are the traffic volume averages over the five sample cases. The entries under "REAL VOLUME" are the real traffic volume counts reported in Rathi [2]. The blank entries represent queues for which data were not reported in Rathi.

The variation in queue levels for the five sample cases results from separate realizations of the Poisson random process. Assuming the real traffic also follows a Poisson process, as is typically done, then we may determine the probability that the real volumes came from the Poisson process simulation. It is found that five of the seven real values fall within the 90th percentile of the Poisson process data for both the non-rush hour and rush hour cases. Additionally, <u>all</u> real values are within the 98th percentile for both cases. Thus, the simulation data and the real data are in reasonably good agreement.

SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION (SPSA) BASED SIMULATION CONTROL STUDY

The scenario we are considering in this simulation control study is one of nine intersections based on five one-way arteries and one two-way artery (analogous to part of the mid-Manhattan network). Figure 1 depicts the scenario. The time of control covers the four-hour period, from 3:30 p.m. to 7:30 p.m. In this four-hour period, the streets, identified by index numbers 1001, 1203, 1304, and 1507, have their traffic levels gradually rising and then falling. Their traffic arrival rates increase linearly from non-rush hour rates starting at 3:30 p.m.. The rates peak at 5:30 p.m. to a rush hour level and then subside linearly until 7:30 p.m. Back-up occurs during rush hour. The streets 1102, 1409, and 1609 have the same traffic trends during this total time period, but they never back up. In addition, the system assumes that there are inflow rates on streets indexed by numbers 0405, 0506, 0807, and 0908 from garage-generated egress at the end of office hours from 4:30 p.m. to 5:30 p.m. The inflow rates are 1⅓ car/min for 0405 and 0506, and 2⅔ car/min for 0807 and 0908.

For the controller, we used a two-hidden-layer, feed-forward neural net (NN) with 31 input nodes. The 31 NN inputs were the queue levels for the 21 queues, time from the start of the simulation, and the 9 outputs from the previous control solution. The output layer had 9 nodes, one for each signal light's green/red split. The two hidden layers had 16 and 6 nodes, respectively. For this NN, there were a total of 597 NN weights that must be estimated as part of the control algorithm.

Figure 2 presents the results of our simulation study of the control algorithm. In order to show true learning effects (and not just random chance as from a single realization) the curve in Figure 2 is based on an average of thirty statistically independent simulations. The values along the vertical axis are total vehicle wait time at all nine intersections over the four-hour time sector (the corresponding loss function was a sum of squared wait time for all vehicles in the network over the entire four-hour time sector). Consistent with the "simultaneous perturbation stochastic approximation (SPSA)" methodology described in Spall [3] and Spall and Cristion [4], the timings change continuously (cycle-to-cycle) as a function of the instantaneous traffic flow while the underlying NN weights that define the control function are changed on a longer-term basis (over days and weeks). The initial total vehicle wait time (at day 0) reflects the wait time when the system is

Table 1: Rush Hour Traffic Volume Simulation

QUEUES	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	AVERAGE	REAL VOLUME
0102	912	878	788	818	878	854.8	
0201	542	480	480	458	440	480.0	
0203	816	754	690	716	766	748.4	820
0205	1,578	1,598	1,462	1,580	1,504	1,544.4	1,494
0302	550	498	518	472	466	500.8	512
0401	1,920	2,020	1,948	1,892	1,960	1,948.0	
0405	552	548	598	554	514	553.2	
0506	700	682	710	692	626	682.0	624
0508	1,542	1,552	1,434	1,544	1,456	1,505.6	1,546
0603	1,346	1,338	1,356	1,310	1,338	1,337.6	
0704	2,040	2,162	2,076	2,018	2,100	2,079.2	
0807	800	732	744	716	812	760.8	
0906	1,328	1,328	1,336	1,302	1,342	1,327.2	
0908	572	548	560	530	614	564.8	492
1001	1,004	962	868	916	1,000	950.0	
1102	1,434	1,474	1,320	1,464	1,368	1,412.0	1,458
1203	596	544	574	518	514	549.2	
1304	326	324	378	348	312	337.6	
1409	322	332	316	316	368	330.8	
1507	2,072	2,242	2,134	2,064	2,164	2,135.2	
1609	1,378	1,398	1,410	1,368	1,412	1,393.2	

Table 2: Non-Rush Hour Traffic Volume Simulation

QUEUES	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	AVERAGE	REAL VOLUME
0102	416	356	382	382	380	383.2	
0201	462	394	428	374	414	414.4	
0203	418	362	388	374	376	383.6	382
0205	1,648	1,700	1,672	1,650	1,648	1,663.6	1,616
0302	434	364	402	354	390	388.8	322
0401	1,740	1,706	1,704	1,724	1,714	1,717.6	
0405	394	318	310	334	332	337.6	
0506	496	424	414	428	432	438.8	478
0508	1,580	1,588	1,574	1,556	1,536	1,566.8	1,560
0603	1,594	1,594	1,550	1,508	1,540	1,557.2	
0704	1,856	1,850	1,852	1,864	1,860	1,856.4	
0807	660	654	636	602	588	628.0	
0906	1,642	1,678	1,636	1,562	1,616	1,626.8	
0908	604	626	602	562	556	590.0	584
1001	384	340	378	370	376	369.6	
1102	1,646	1,726	1,714	1,664	1,686	1,687.2	1,656
1203	414	352	406	348	388	381.6	
1304	244	172	166	190	188	192.0	
1409	524	568	542	494	494	524.4	
1507	1,890	1,902	1,918	1,928	1,928	1,913.2	
1609	1,688	1,738	1,698	1,628	1,684	1,687.2	

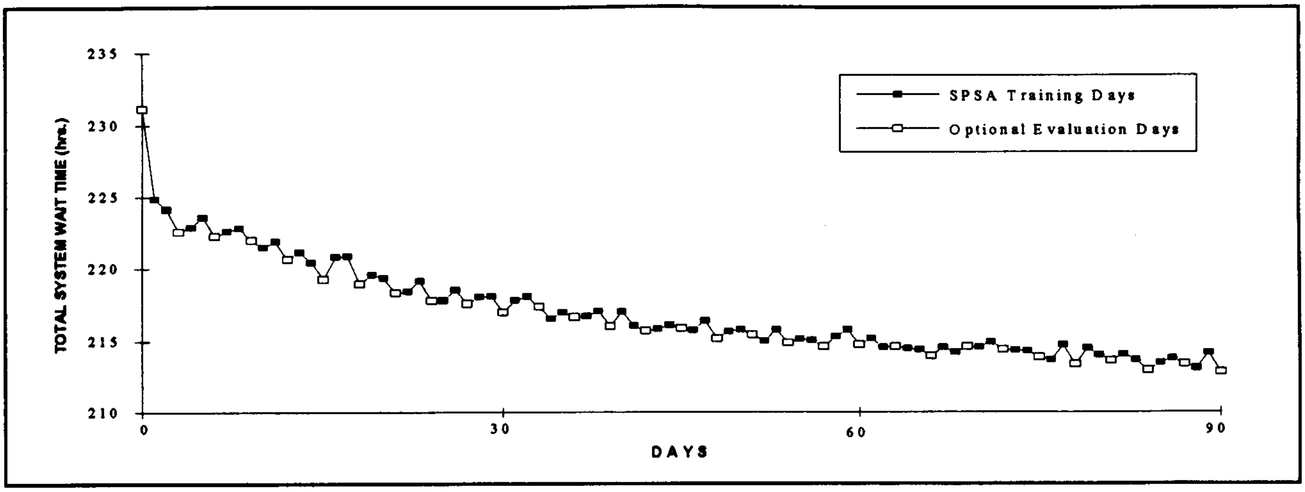

Figure 2: Reduction in Total System Wait Time Through SPSA Training for Mid-Manhattan Scenario

using prior timings consistent with a rush-hour scenario. These timings, i.e., the ratios for green-time/total-cycle-time on N-S arteries, were all equal to 0.55.

As shown in Figure 2, the total system wait time decreases from 232 hours on iteration and day 0 to 212 hours on iteration 30 (equivalent to day 60 for using only training days and day 90 for including optional evaluation days in addition to training days). Normally, the algorithm uses only training perturbations and does not evaluate the system at the current optimal solution. The reduction on total wait time of around 9%, represents a reasonably large savings with a relatively small investment for the high traffic density sectors. In comparison, major construction changes to improve traffic flow by 9% in this well-developed area would be enormously expensive. Again, the total system wait time displays a generally downward trend; given more time, the system is expected to show continuing gradual improvement.

SUMMARY

We have described in detail the mid-Manhattan high density sector traffic simulation software. Secondly, we have shown how it was evaluated against real traffic data. The simulation and the real data are in reasonably good agreement for average conditions during both non-rush hour and rush hour scenarios. Finally, we discussed the application and performance of the simulation in an interactive mode in the study of a particular adaptive traffic control algorithm. The simulation was successful in providing variations in queue sizes and total system wait time in immediate response to controller-selected signal light split times at all intersections in the network. The ease of operating the simulation for a variety of control selections over a simulated real-traffic period of 60 to 90 days allowed the controller to converge to a significantly improved selection of signal split times for all intersections in the simulated network.

REFERENCES

[1] Morlok , E. K. 1978, *Introduction to Transportation Engineering and Planning*, McGraw-Hill, New York.

[2] Rathi, A. K. 1988, "A Control Scheme for High Traffic Density Sectors," *Transportation Research B*, vol. 22B, no. 2, pp. 81-101.

[3] Spall, J. C. 1992, "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Trans. on Automatic Control*, vol. 37, no. 3, pp. 332-341.

[4] Spall, J. C. and Cristion, J. A. 1994, "Nonlinear Adaptive Control Using Neural Networks; Estimation with a Smoothed Simultaneous Perturbation Gradient Approximation," *Statistica Sinica*, vol. 4, no. 1, pp. 1-27.

[5] Federal Highway Administration 1993 *TRAF User Reference Guide Version 4.0*, Office of Safety and Traffic Operations R&D, IVHS Research Division (HSR-11), 6300 Georgetown Pike, McLean, VA 22101-2296 (April).

[6] William R. McShane 1990, *Traffic Engineering*, Prentice-Hall.