The Booth Tolls for Thee

Team 770
February 7, 2005

Abstract

In this paper, we address the problems associated with heavy demands on toll plazas such as lines, backups, and traffic jams. We consider several models in hopes of minimizing the "cost to the system," which includes the time-value of time wasted by drivers as well as the cost of daily operations of the toll plaza.

One model yields a microscopic simulation of line formation in front of the toll booths when the service rate cannot match the demand. Using hourly demand data from a major New Jersey parkway, the simulation is limited in not taking bottlenecking effects into consideration. The results, however, when subjected to threshold analysis can serve to set upper bounds on the number of booths that could potentially be suggested by any other models.

After presenting this basic model, a more general, macroscopic framework for analyzing toll plaza design is introduced. In analyzing "total cost" and allowing bottlenecking, this model is more complete than the first, and it is able to make recommendations for booth number based on data obtained from the first model. This computation melds the macro- and micro- levels, a strategy that is helpful in looking at toll booth situations.

Finally, a model for traffic flow through a plaza is formulated in the world of "cellular automata." An interesting take on microscopic ideas, the cellular automata model can serve as an independent validation of our other models.

In fact, the models mostly agree that given L lanes, a number of booths around $B = [[1.65L + 0.9]]$, where $[[x]]$ is the greatest integer less than x, will minimize the total human cost associated with the plaza.

Introduction

When Will Smith's name was called, announcing his receipt of the Best Male Performance award at a recent installment of the MTV Movie Awards, he bounded to the podium. Needling MTV for baiting him to attend their show, Smith snarkily quipped, "M.T.V.: My Time is Valuable." And of course, many other Americans would say the same for themselves, without Smith's irony. In an economy as driven by efficiency as America's, there certainly seems to be a predominant national mindset induced by the toil of the workweek. It's broader than just a desire to be busy or to accomplish; rather it extends to the notion of having control over one's own time, something we abhor to see wasted.

It is no stretch, either, to say that paying a toll to be able to travel to one's destination is largely viewed as an inconvenience. Americans have, for the most part, come to view having free, open roads as an inalienable right from our government. Toll roads are then aberrant and annoying.

But the vexing aspects of toll roads do not stop at the quarter or 35-cent fee, but rather include the time that drivers are *forced* to waste. Stopping at tolls retards the steady, quick flow of a highway, while not necessarily offering safety benefits like stopping at traffic lights (which are widely tolerated). What's worse is when heavy demand creates jams in the merging lanes exiting the booths or backs up traffic in delineated stripes of hot metal and hotter humanity entering the plaza. The time spent at a toll plaza is easily and often seen as time that could be more fruitfully spent. It is time when the drivers lose sovereignty over their personal whims and obligations.

Despite the anachronisms, imagining Sir Isaac Newton being stranded in a car at a toll plaza when the all-important apple decided to drop back at his home, or envisioning Albert Einstein sitting in a traffic jam without a pencil the moment that relativity dawned upon his own head serve to illustrate some (hyperbolic) motivation in trying to make the toll process as expedient as possible. It is certainly strange to think so, but Newton and Will Smith have something in common.

Restatement of the Problem

Drivers have places to be and people to see, but for one reason or another, tolls must at times be collected from them. It is our goal in this paper to make the process more *optimal* for everyone involved, including the owners and operators of the booths, and of course the drivers. The only mechanism for optimization at our disposal is, presumably, adjustment of the number of booths present at a certain toll plaza, given the number of lanes entering and exiting it.

During peak hours, which occur typically when suburbanites make their way to and from work in larger cities, it is common for lines to form entering the tollbooths, as demand overtakes the fastest rates that the tolls can be collected. On the other side of the booths, too, as the (often) greater number of lanes coming out of the booths converge back down to the original number, bottlenecks and jams are wont to amass in response to the harried merging.

We seek to balance these effects, along with the cost associated to offer extra booths, in order to provide reasonable recommendations for how to minimize the waste of time and money in the toll-collecting process by adjusting the number of booths offered at a given toll plaza.

Previous Work in Traffic Theory

Mark Twain famously remarked, in his disdain for arithmetic, that the answer to all mathematical problems is three. While insightful, the Twain model leaves some room for improvement in addressing the tollbooth conundrum at hand. A five-lane highway will need more than three tollbooths, but Twain wrote great novels.

There is a rather substantial literature on models for traffic flow, and most models fall into one of two categories: microscopic and macroscopic.

The microscopic models are the ones that can be said to "miss the forest for the trees." They examine the actions and decisions made by individual cars and drivers. Often these models are called car-following models since they use the spacing and speeds of cars to characterize the overall flow of traffic. Interesting models have emerged from examining cellular automata in a traffic sense (much more to come) and queuing theory.

Macroscopic models tend to view traffic flow in analogy to hydrodynamics and the flow of fluid streams: just as blood hurtles red blood cells through veins, vehicles pulse down streets toward their destinations. The "average" behavior is assessed, and commonly used variables include steady-state velocity, flux of cars per time, and density of traffic flow.

Some models bridge the gap, including the gas-kinetic model which allows for individual driving behaviors to enter into a macroscopic view of traffic, much like ideal gas theory can examine individual particles and collective gas [Tampere, et al. 2003].

The tollbooth problem is an interesting addition to the traffic literature because it involves no steady velocity, so macroscopic views may be tricky. On the other hand, specific bottlenecking events are quite complex, and microscopic ideas are certainly put to the test.

An M/M/s queue (vehicles arriving with gaps determined by an exponential random variable, to *s* tollbooths, and service at each tollbooth taking an exponential random variable amount of time [Gelenbe, 1987]) seemed appropriate at first. However, queuing assumptions did not satisfy our thirst for details about bottlenecking and about multiple lanes.

Drawing on ideas from old models, while still developing ideas more pertinent to the tollbooth problem, we were able to incorporate aspects of the situation from a small-scale into a larger-scale framework. It seems that neither micro- nor macro- will alone be adequate to capture the dynamics of a toll plaza, though our cellular automata simulation (for herky-jerky driving at lower speeds) produced some surprisingly good results (in terms of matching with other analyses).

Properties of a Successful Model

A successful toll plaza configuration should achieve the following objectives:

- Maximize efficiency of the toll plaza by reducing customer waiting time (due to bottlenecking, tollbooth lines, etc.)

- Suggest a reasonably implementable policy to toll plaza operators

- Be robust enough to efficiently handle the demands of a wide range of operating capacities

- As the number of highway lanes feeding the toll plaza is increased, the optimal number of tollbooths will not decrease.

General Assumptions and Definitions

Assumptions

- **There is only one type of driver in the system**. In navigating toll plaza traffic, all drivers act according to the same set of rules. Although the individual decisions of any given driver are probabilistic, the associated probabilities are the same for all drivers.

- **Bottlenecking downstream of the tollbooths does not hinder their operation**. Vehicles which have already passed through a tollbooth may experience a slowing down due to the merging of traffic, but this effect is not extreme enough to block the tollbooth exits.

- **The number of highway lanes does not exceed the number of tollbooths**. An obvious solution to the posed problem may especially occur to those sitting still at a traffic plaza: namely, set the number of tollbooths equal to zero. The assumption above instead ensures that the number of tollbooths must be strictly positive.

- **All tollbooths offer the same service and vehicles do not distinguish between them.** We seek to improve toll plaza efficiency by optimizing the *number* of tollbooths – not the services they provide. While several types of tollbooth exist in practice, we have not been charged with distinguishing between them and suggesting their selective use. This is a problem of a different nature. Later, we return to this assumption and list ways in which our solution might change in response to multiple booth types.

- **The amount of traffic on the highway is dictated by the number of lanes on the highway and not the number of tollbooths.** Changing the number of tollbooths for a given number of lanes does not affect the 'demand' for the roadway.

- **The number of operating plaza booths remains constant throughout the day.**

Terms and Definitions

- **Take a "highway lane" to be a lane of roadway in the original highway before and after the toll plaza**. Thus, the number of 'lanes' in a given toll plaza configuration depends not on the plaza itself but on the width of the roadway before and after the toll barrier.

- **Influx** is the rate (in cars/min) of cars entering all booths of the plaza.

- **Outflux** is the rate (in cars/min) of cars exiting all booths of the plaza. It is a function of time.

Optimization

Next, we seek a method of optimization that can be used to evaluate potential solutions to the problem. How do we decide that a given toll plaza configuration is optimal? One natural way to compare potential solutions is to compute the total time drivers spend waiting in the toll plaza. It seems logical to conclude that well-designed toll plazas will require less customer waiting time than their inefficient counterparts. Although this method might offer insight, we note some serious drawbacks. Namely, this waiting time minimization disregards the standpoint of the agency operating the toll plaza. In other words, minimizing the waiting time for customers may not present convenient policy options for toll plaza operators.

Suppose a model based on waiting time minimization suggests that forty tollbooths should be used in a plaza for a six lane stretch of highway. Should operators heed this advice? Certainly, the operators of the plaza will incur significant cost in building and maintaining such a facility. In crowded areas, it may not even be possible to construct a toll plaza of this size. Furthermore, tollbooths employ personnel to serve customers without exact change. Paying additional construction, maintenance, and labor costs may not be worth the added benefit of lowering customer wait time.

We seek a more balanced method of facility optimization. This method must consider not only the customers, but also the agency operating the toll plaza. To implement this scheme, we must somehow equate customer waiting time with toll plaza operating costs.

We elect to use cost as a yardstick of our solutions. Cost is a convenient medium due to its ubiquity in our culture and the relative ease of its translation into time. In considering the entire plaza system, we seek the facility configuration that will generate the lowest net cost. This cost will be distributed among both parties in our system – the users and the operators.

In creating a cost optimization apparatus, we invoke the following terms and definitions:

- The general cost, C [dollars], of a toll booth is the time-value of the delays incurred at a toll plaza for each individual (driver or passenger) AND the cost associated with daily operations of the booths at the plaza. The toll fees themselves and the upstart cost of building a new plaza are NOT part of this cost.

- α is the average time-value of a minute for a car occupant.

- γ is the average car occupancy.

- N is the total number of (indistinct) tolls paid over the course of the day.

- L is the number of lanes entering and leaving a plaza. B is the number of booths in the plaza.

- Q [dollars] is the average daily operating cost of a human-staffed tollbooth.

The underlying goal of this construct is to find a reasonable number of toolbooths, B that minimizes cost C, a function of B. We formulate this function $C(B)$.

First, notice that the total waiting time per car will be WN, and so the total cost incurred by waiting time will be $W\alpha N\gamma$. General human time-value is cited as \$6/hour or α = 10 cents a minute [Boronico, 1998]. The amount that must be expended to operate a booth for a day would then be QB. The average annual operation cost for a human-staffed tollbooth is \$180,000, so we set Q = 180000/365.25 [Sullivan, et al. 1994].

Reasoning that W depends on B, we now see that

$$C(B) = W\alpha N\gamma + QB$$

This is the function we'll want to minimize with respect to B (for a given L). Naturally, the knee-jerk reaction is to take its derivative and set it equal to zero, showing that the B we seek must necessarily satisfy

$$W'(B) = \frac{-Q}{\alpha N\gamma}.$$

Fourier Approximation of Toll Plaza Car Entry Rate

From a previous research paper's traffic flow data [Boronico, 1998], we find the mean demand per minute (influx) of cars for a toll plaza on a given typical day. The reason the data peaks are very high at the 6am – 7am rush hour and are not as high during the 3pm – 4pm rush hour period is that the data is collected in the direction headed toward the metropolis. Thus, the main reason for traffic on a typical weekday, the workers during a business day, will be using the "toward big city" tollbooths in the morning, and these tollbooths will be far less frequented in the evening hours.

Table 1: Fourier Approximation of Influx Data

Start Time	**End Time**	**Hour***	**Influx (cars/min)**	**Fourier Approx of Influx****
12:00 AM	1:00 AM	0.5	15.44	15.16272478
1:00 AM	2:00 AM	1.5	15.32	15.42467822
2:00 AM	3:00 AM	2.5	15.16	15.18796896
3:00 AM	4:00 AM	3.5	19.9	19.81853474
4:00 AM	5:00 AM	4.5	47.09	47.22251986
5:00 AM	6:00 AM	5.5	89.95	89.61825869
6:00 AM	7:00 AM	6.5	105.9	106.4828683
7:00 AM	8:00 AM	7.5	85.52	84.72959878
8:00 AM	9:00 AM	8.5	54.68	55.57942216
9:00 AM	10:00 AM	9.5	43.11	42.42662327
10:00 AM	11:00 AM	10.5	40.16	40.49538486
11:00 AM	12:00 PM	11.5	40.85	40.83544106
12:00 PM	1:00 PM	12.5	41.72	41.63346483
1:00 PM	2:00 PM	13.5	44.54	44.44085865
2:00 PM	3:00 PM	14.5	48.88	49.29448007
3:00 PM	4:00 PM	15.5	53.2	52.55619485
4:00 PM	5:00 PM	16.5	51.61	52.21058951
5:00 PM	6:00 PM	17.5	48.38	48.16410937
6:00 PM	7:00 PM	18.5	39.72	39.50374966
7:00 PM	8:00 PM	19.5	30.51	31.11397219
8:00 PM	9:00 PM	20.5	29.48	28.86864636
9:00 PM	10:00 PM	21.5	26.82	27.196867
10:00 PM	11:00 PM	22.5	21.21	21.2608522
11:00 PM	12:00 AM	23.5	17.22	16.91795178

* All "hour" values are averages of the start and end time, so as to accommodate the Fourier approximation.

**

$$F(t) = 41.68 - 16.38\cos(t\omega) + 12.53\sin(t\omega) - 18.59\cos(2t\omega) + 0.6307\sin(2t\omega) + 3.572\cos(3t\omega) - \ldots$$
$$-13.67\sin(3t\omega) + 7.876\cos(4t\omega) + 0.4378\sin(4t\omega) - 0.5048\cos(5t\omega) + 6.93\sin(5t\omega) - 2.97\cos(6t\omega) + \ldots$$
$$+0.4869\sin(6t\omega) + 0.2518\cos(7t\omega) - 1.554\sin(7t\omega) + 0.5785\cos(8t\omega) - 0.5871\sin(8t\omega),$$

where $\omega = \frac{2\pi}{24} = 0.2513$. This approximation fits the data points with an R^2 value of 0.9997, and a further glimpse into the coefficients can be seen in the Appendix (I).

Before settling on a Fourier Series approximation with 8 terms, we first attempt a fit with a quartic polynomial. Though we are able to find a very close fit, there is an obvious downside. The quartic polynomial will not necessarily have the same value for $t = 0$ hours and $t = 24$ hours, even though this is obligatory for a cyclic model of daily traffic influx. Therefore, we choose a Fourier Series approximation, whose main upshot is its inherent periodicity, and whose period we can define as the length of a day. Also, it is worth noting that we use an approximation, rather than the data at hand, because we need an influx rate at every minute of the day, instead of just once an hour. In order to have a value at every minute, we need an estimation with more continuity.

Model 1: Car-Tracking Without Bottlenecks

Approach

We create the car-tracking model in order to place an upper bound on the optimal number of booths in a toll plaza configuration, given a particular number of lanes. The model looks at a typical day's influx of vehicles into a toll plaza (data from [Boronico 1998] and Fourier Series approximation).

As vehicles approach the tollbooths within the toll plaza, they may or may not be held up by other vehicles being served. Each vehicle is looking to get through the toll plaza as quickly as possible, and the only factor that may cause Car A, which arrives earlier than Car B, to leave later than B is the random variable of service time at a tollbooth. In other words, cars do not make bad decisions concerning minimizing their wait times.

Assumptions

- **Customers are served at a tollbooth at a rate defined by an exponential random variable** (a common assumption in most queuing theory [Gelenbe, 1987]) with mean 12 seconds per vehicle (or 5 cars per minute).

- **Traffic influx occurs on a "per lane" basis**, meaning that influx per lane is constant over all configurations with varying number of lanes.

- **Bottlenecking occurs more frequently when there are more tollbooths, given a particular number of lanes.** This implies that omitting bottlenecking from our model will cause us to overestimate the optimal number of tollbooths for a given number of lanes, thus preserving our model as an upper bound for the optimal number of tollbooths.

- **There exists a time-saving threshold such that if the waiting time saved by adding another tollbooth is under this threshold, it is not worth the trouble and expense to add the tollbooth.** We assume that if an additional tollbooth does not reduce the maximum waiting time over all cars by the same amount as the average time that it takes to serve a car at a tollbooth (12 seconds = 0.2 minutes), then it is an unnecessary addition.

- **An incoming car within the toll plaza will choose the tollbooth that will be soonest vacated, if all are currently occupied.** If only one is vacant, the car will choose that tollbooth. If multiple tollbooths are vacant, the car will choose the one that was vacated the earliest. This last statement was created only to give the model a defined path, and it does not actually affect the waiting time in line.

- **Cars make rational decisions with the goal of minimizing their wait times.**

Expectations of the Model

- An additional booth should not increase time spent waiting in line before the booths.

- Each additional tollbooth experiences diminishing returns in terms of time saved, because each one is used at best as often as one of the already existing tollbooths.

Development of Model

As in the other models, cars arrive at the toll plaza at a rate described by the Fourier Series approximation of the data collected from [Boronico, 1998]. Cars are considered inside the toll plaza (meaning that we begin to tabulate their waiting times) when they are either being served or waiting to be served.

Service time does not count as waiting time, so if a car enters the toll plaza and there is a vacant tollbooth, its waiting time is 0. If there are no vacant tollbooths at a particular moment, cars will form a queue to wait for tollbooths, and they will enter newly formed vacancies in the order in which they entered the toll plaza. Once a car has been served, it is considered to have exited its tollbooth and the toll plaza as a whole.

Bottlenecking has not been considered in this model, because we only wished for this model to serve as an upper bound for the number of tollbooths for a given number of lanes. Inspection of waiting time in line before the tollbooths sufficiently serves this purpose, and these waiting times will be placed under threshold analysis to determine when the marginal utility (in terms of time saved) of an additional tollbooth is negligible.

It should be noted that our car tracking model does not lend itself to cost optimization, because it is a very basic model that takes into account only how long each car waits in line for tollbooths. It does not factor in the agency controlling the tollbooths, or any costs incurred therein. The code for this model is included in Appendix II.

Simulation and Results

We collected data from the program, for toll plazas with configurations of $L \in \{1,2,3,4,5,6,7,8,16\}$, and each B from L up to a point where additional booths no longer had ANY noticeable effect on waiting time, where ***L*** **is the number of lanes** and ***B*** **is the number of tollbooths** in the toll plaza configuration. For example, for $L = 1$, we collect data for $1 \le B \le 5$, because each additional tollbooth beyond 5 in a one lane configuration has an extremely small, if any, effect on waiting time. NOTE: This is not the same as our cutoff for negligible marginal utility from each added tollbooth. In fact, we clarify the cutoff for negligible marginal utility below, with the help of our work for the case of 6 lanes entering the toll plaza. We choose the 6-lane configuration in order to match those examined in the models to come.

Table 2: Waiting Time for Six Lane Simulation

Booths	Average Wait	Average Wait 2*	Maximum Wait	Marginal Utility
6	28.2683	43.0457	98.6196	N/A
7	12.1237	27.7573	55.1100	43.5096
8	5.9544	16.5089	31.6650	23.4450
9	2.4111	8.3740	15.8357	15.8293
10	0.2450	1.2188	2.7815	13.0542
11	0.0223	0.1666	0.7482	2.0333
12	0.0039	0.0653	0.3100	0.4382
13	0.0012	0.0410	0.2699	0.0401

*Average Wait 2 refers to the average wait time per car that waits for longer than 0 seconds. This is more relevant, because as more booths appear, fewer cars spend any time waiting.
** All times are in minutes.
*** These values were obtained by averaging from between 20 and 30 tests.

To get a more detailed look, we examine some data plots of the toll plaza configuration with 6 lanes and 10 tollbooths:

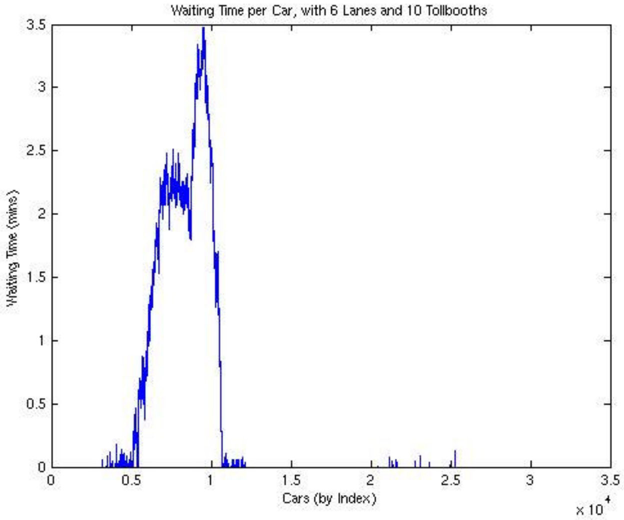


Figure 1: Waiting time per car in six lane, ten tollbooth configuration, model # 1

In choosing an optimal number of booths by threshold analysis, we seek out the first additional booth that fails to reduce the maximum waiting time for a car by at least the length of the average tollbooth service time (0.2 minutes). The column denoted "Marginal Utility" shows the amount that each additional booth reduces maximum waiting time, and we simply notice that for 13th booth, this value is 0.0401 minutes. In short, based on our assumptions, it is unnecessary to build a 13th tollbooth for a toll plaza serving 6 lanes of traffic. Thus, we optimize at $L = 6 \rightarrow B = 12$. In a similar manner, we optimize the toll plazas as follows:

Table 3: Optimized Number of Booths for *L* Lanes

Lanes	Booths
1	4
2	5
3	7
4	8
5	10
6	12
7	13
8	16
16	29

Also, we wish to explore the situation in which there is one lane per booth:

Table 4: Waiting Times for *L* Lanes with *L* Booths

Lanes	Booths	Average Wait	Average Wait 2*	Maximum Wait
1	1	28.2400	37.6928	96.1541
2	2	31.6109	43.2415	103.7251
3	3	28.7137	40.8372	99.6343
4	4	30.8517	44.5677	102.4173
5	5	29.4785	44.3510	103.0286
6	6	28.2863	43.0457	98.6186
7	7	29.7364	45.7080	103.0432
8	8	27.7961	43.9531	96.2895
16	16	31.1367	48.9584	103.7946

*Average Wait 2 refers to the average wait time per car that waits for longer than 0 seconds.
**These values were obtained by averaging from between 20 and 30 tests.

We find that with average wait times of around 30 minutes, average wait time of over 40 minutes per car that waits, and maximum wait times of around 100 minutes, the situation with one lane per booth is entirely undesirable.

Discussion

We find that our model matches our general assumptions, and our assumptions for this model, and more importantly, that our results match our expectations. Namely, worth pointing out are the facts that the number of booths is increasing with respect to the number of lanes, that additional booths have diminishing returns on reduced waiting time (true for all L tested, and visible in the $L = 6$ chart), and that each additional booth does reduce waiting time in line.

The benefits of this rather simplistic model are speed and definite upper bounds. Because we know for a fact that bottlenecking will cause a configuration with fewer booths to be optimal, we know that the optimal number of booths can be no greater than those values presented in this model. Thus, this model's greatest strength is that it serves to check the values of the other two models, which provide much more detailed and accurate simulations of the situation at hand.

The obvious weakness here is that we do ignore bottlenecking, but once again, this allows us to use our results to bound the results of the other two models. In short, the results of the next two models will show that this was in fact an appropriate model in terms of setting a boundary for our soon-to-be-realized answers. With respect to the specific situation in which there is one tollbooth per lane, the toll plaza performs quite terribly. This is mainly because the influx of cars occurs much more rapidly than the service in tollbooths, and such a situation may be addressed by allowing cars to leave tollbooth lines. This would add some practicality to the model, because while in general people may wait out a tollbooth line, they would be much more likely to leave if they knew that the wait could be up to 100 minutes.

Model 2: A Macroscopic Model for Cost Minimization

Motivation

Model 1 was an effective means of inputting and evaluating specific data sets. General trends could be inferred from the simulated motion of many individual cars at a toll plaza, especially regarding an upper bound on the number of tollbooths given a certain number of lanes. A theoretical framework for making those generalizations and conclusions, however, was missing. In this model we take a bird's eye view of the problem at hand to really address what we mean by the "optimization" of tollbooth processing. A decision-making apparatus is constructed that can be used with the results from Model 1 to refine our earlier observations.

Approach

This model operates on a “higher” level than Model 1. It is less concerned with the details of individual vehicular motion and decision-making, but rather the general aggregate effect of the motions of the cars. The variables of this model will often be averages over a day's time.

For instance, there is no need for this model to decompose analytically the situation of two cars trying to merge into the same lane. Instead, it recognizes that beyond a certain threshold of outflux from the booths, some bottlenecking will occur. The determination of this threshold and its dependencies will be one of the problems addressed in the development of this model.

Also, the concept of the “cost” of a particular plaza configuration is utilized by this model. In minimizing this cost with respect to the number B of booths given a number L of lanes, we obtain a value for B that satisfies our concept of optimality.

Defining what this nebulous notion of “cost” should mean is the real dilemma to be addressed, but also the impetus for designing this particular model. We chose not to focus on the actual values of the tolls paid by the drivers nor the initial costs to plaza owners of renovating their plazas. Those are not the costs affecting long-term efficient design: the same people will likely be using the road despite any potential increase in tolls that might be instituted to absorb the costs of renovation. The more interesting cost to the drivers is a *time-cost*. As Will Smith's quote in the Introduction hinted, human time has value, and while Isaac Newton's may be worth more than others, we'll just assume a national average of $6/hour. By calculating time wasted in waiting at toll plazas, we can get a handle on money lost from potential work unable to be completed.

The other cost to consider is that of actual daily operations of the plazas. We combine these "costs to the system" in this model and attempt to minimize their sum with respect to the number of booths. (See Optimization Section for mathematical treatment.)

Assumptions, Variables, and Terms

- We monitor traffic over the course of one day.
- W [minutes] is the average waiting time per car in the toll plaza. W accounts for time spent in lines (W_1), service time(W_2), and bottlenecking (W_3). (The numbering reflects the progression through the toll plaza.)
- F_i is a function of time giving the influx of cars per minute to the plaza from one lane. F_o gives the outflux from the booth in cars per minute (in practice, an average of at least 20 individual F_o curves since they vary probabilistically).
- r is the maximum potential service rate [cars per minute]. This is the rate each booth operates at when the influx is non-abating.
- There is an outflux barrier, K [cars per minute], above which bottlenecking takes place. It is taken to be linear in L and independent of B, and we call it the bottlenecking threshold.

Development

To discover an appropriate value for B, we'll need to know more about W.

First note that W_1 and W_3 both depend on B and that $W = \sum_{i=1}^{3} W_i$. It is easily seen from the definitions that $W_2 = 1/r$.

W_1 represents the average amount of time each car spent waiting in lines before reaching the tollbooth. Thus, it begins to accumulate when the total influx of traffic exceeds the toll plaza's capacity to handle the traffic. $LF_i(t)$ gives the total influx, and Br gives the maximal rate of service. When the difference between the former and the latter is positive (represented by areas enclosed by the curves in Figure 2), we'll want to integrate over time to calculate how many cars were forced to wait in line.

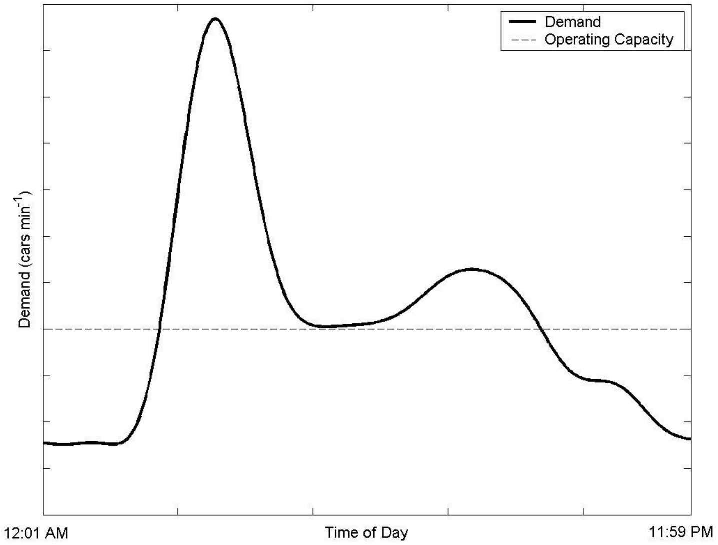


Figure 2: Enclosed area (above line, below curve) represents total number of cars in line at any time during the day.

Integrating again gives us the total amount of time waited by those cars (with 3600 being a scale factor for time units):

$$3600\int_0^{24}\int_0^{t}\max(LF_i(\tau)-Br,0)d\tau dt$$

The average over the total number of cars will be this expression divided by *N*. In summary,

$$W_1=\frac{3600}{N}\int_0^{24}\int_0^{t}\max(LF_i(\tau)-Br,0)d\tau dt$$

As it turns out, the technique to obtain *W*3 is quite analogous to that we used to find *W*1. When comparing the bottlenecking situation to the line-up situation just considered, analogies can be drawn from influx to outflux and from *Br* to *K*. As one might expect, we calculate *W*3 as follows:

$$W_3=\frac{3600}{N}\int_0^{24}\int_0^{t}\max(F_o(\tau,B)-K,0)d\tau dt$$

The problem remains of how to determine upon what variable(s) *K* depends. First, *K* is not directly dependent on *B* since bottlenecking should only be a result of general outflux from the booths into *L* lanes. *K* is instead indirectly dependent on the number of booths, because outflux, F_o, depends on *B*, and *K* in turn depends on outflux. *K* also can be considered a linear function in terms of L, because L is directly proportional to influx, which, by the law of conservation of traffic, must equal outflux in the aggregate. Now, we have all the mathematical constructions in place in order to find the *B* that minimizes total general cost reasonably. Computationally, however, there is still a long road to hoe.

Simulation and Results

As discussed, this macroscopic model will serve as a framework in which we can analyze the data obtained from Model 1 in a cost-minimization procedure. Using the same data for traffic entering the toll plaza as was used in Model 1, this new model will now be able to address the waiting time incurred by bottlenecking and eventually minimize total general cost to the system with the correct selection of B.

As a point of illustration, this particular discussion will detail the calculation of the recommended value of *B* for $L = 6$. The techniques used with other values of *L* were completely analogous.

Clearly, with *Q*, α, γ, and *N* all independent of *B*, the most challenging computational procedure will be to find *W* as a function of *B*, specifically W_1 and W_3 as individual functions of *B*.

Given the influx function used in Model 1 (see Appendix I), we can use Mathematica to numerically integrate the expression for W1 with a given *L* and $r = 5$ cars per minute (as used in the other models). (*N*, of course, comes from an integration of the influx expression.) A specific value of *B* must also be used in the evaluation, so we do the calculation for a series of *B* values ranging from *L* to $L + 7$, with $L + 7$ usually being above the upper bound from Model 1 (and step size: 0.25). A Mathematica quartic polynomial fit is then done on the resulting points (*B*,W1(*B*)). This procedure will give W_1 as a function of *B*, as desired.

The $L = 6$ example will perhaps be illustrative. We can easily integrate to find the value $N = 92355$. We compute W_1 for values of *B* ranging from 6 to 13, in steps of 0.25 (for a better fitting curve). The plot of these points along with the best-fit quartic is given in Figure 3.

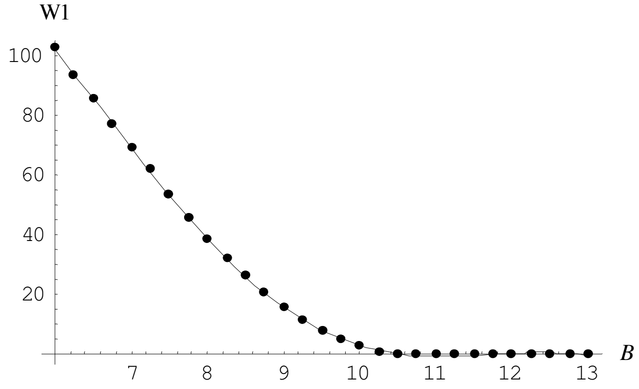


Figure 3: Plot of W1 for B ranging from 6 to 13 (actual points with quartic fit)

The recipe for $W_3(B)$ is somewhat less straight-forward since $F_o(t)$ is generated from a stochastic distribution, unlike the deterministic $F_i(t)$. $F_o(t)$ also depends on B, a significant complication. At least 20 trials of each case (L, B) were run under the first model, and the averaged outcomes of their outflux functions become the function we used for outflux in this model's simulation, henceforth referred to just as $F_o(t, B)$.

Once F_o(t, B) was determined for each salient value of L, we used surface-fitting software (Systat's TableCurve3D) to generate an expression for outflux as a function of time AND the number of booths. We use this expression in the compound integral for W_3, and used Mathematica to integrate numerically for the same values of B selected earlier (to determine the plot for $W_1(B)$). As before, we generate a scatter plot of points (B,$W_3(B)$) and do a Mathematica quartic polynomial fit to the data. This is the function of B we seek.

It is important to note that correlation coefficient values (R^2) for the surface fits all fall between 0.84 and 0.95, which are certainly acceptable values. All of the quartic fits have correlation coefficients very near 1.

Again, this process is best illustrated with the concrete $L = 6$ example. The data used for the surface fit (shown below in Figure 4) is included in Appendix IV. As in determining $W_1(B)$, the values of W_3 are shown for values of B from 6 to 13 (step size: 0.25), along with the fitted quartic (see below). Now we have $W_3(B)$, and are ready to do the cost minimization. In general, dW/dB will be a cubic (as a result of the quartic fits), and so three solutions emerge. In all cases examined, only one is an appropriate value to minimize C (when rounded). These values are summarized below in Table 5.

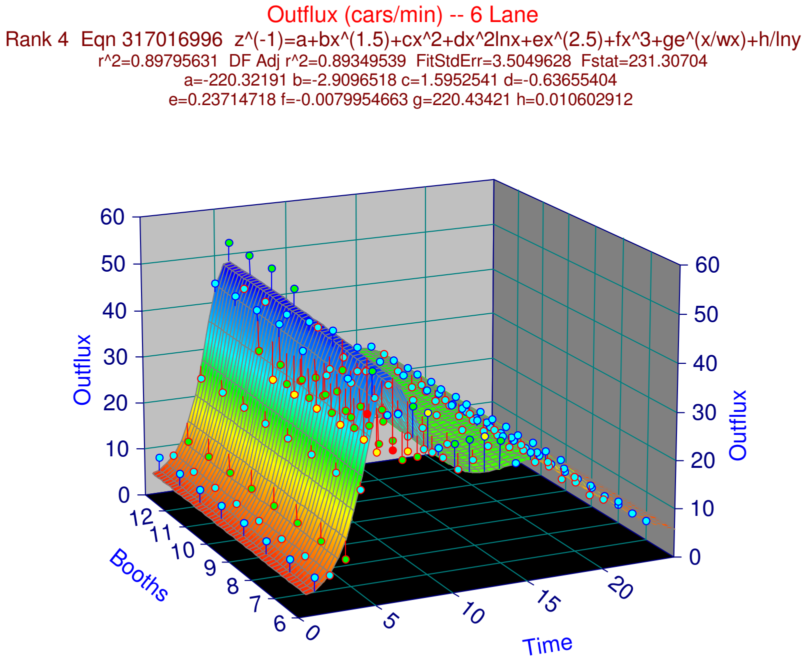


Figure 4: Surface fit for outflux function with respect to time (hrs) and number of booths for L = 6

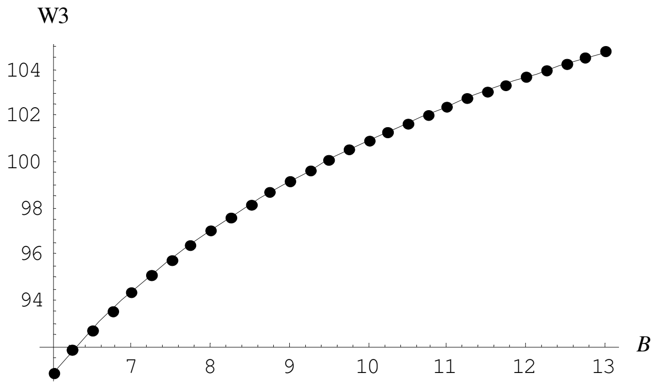


Figure 5: Plot of W3 for B ranging from 6 to 13 (actual points with quartic fit)

Table 5: Model 2 Final Recommendations

Lanes	Booths
1	3
2	5
3	6
4	7
5	9
6	11
7	12
8	14
16	27

Mathematica's numeric solver gives the B-value of the minimum in *C*(*B*) at *B* = 10.84. Thus, this model recommends 11 booths for 6 lanes of traffic at a toll plaza.

Discussion

This model really amounts to a theoretical way to calculate total waiting time for drivers based on general ideas of traffic flow. The values given above in the summary table are actually quite reasonable and satisfy many of the expectations we laid out characterizing a successful model of the tollbooth problem. The recommended B-values increase monotonically with *L*, as expected, and they also fit in under the "upper bounds" produced in Model 1, as hoped. Clear from the table above, too, is that **a 1-booth per lane setup was never chosen as optimal.** In other words, B = L never seems to be a minimizing value for B in C(B) (for given L). This is because, as we can see from the graphs above of W_1 and W_3, while bottlenecking is zero, waiting time in line before the tollbooths is much higher, thus diminishing the effect of bottlenecking.

Given the model's success, it may be disheartening to acknowledge its lack of robustness. The formulation of this model glosses over the fine details of traffic behavior, so any adjustments to fine-scale aspects of traffic, such as the addition of a potential E-ZPass lane (to be discussed later), would be impossible to implement in any specific manner. Perhaps the rate *r* of service could be adjusted higher for the given scenario, but changing lanes before the tollbooths in anticipation of such a lane would be difficult to capture with this model. Nevertheless, the model works quite well, it seems, for the given formulation of the problem.

Model 3 – Cellular Automata: Life! (on the Road)

Motivation

In approaching the problem of toll plaza optimization, we have heretofore considered traffic flow in a macroscopic sense. To avoid the complexity of accounting for individual cars, we have used continuous flow parameters averaged over the course of a day. Although this approach offers mathematical tractability, it may be overlooking important behavior. What effect does the discreteness of traffic have on the nature and solution of the problem?

At its heart, traffic movement is a discrete phenomenon. Any caffeinated weekday morning commuter would attest to this. Traffic lights, bottlenecks, and construction zones can turn a daily commute into a stop-and-go nightmare. Although steady-state traffic flow may be well-characterized by continuous equations, vehicles and tollbooths may not always lend themselves to such convenient description. Each vehicle is operated by a driver whose behavior and decisions may not necessarily be predictable. Drivers introduce the variable elements of reaction time, attentiveness, and following distance to traffic analysis models. Vehicles and tollbooths are rigid objects and occupy specific positions. While "collisions" of continuous fluid elements may be of little consequence, they can be devastating in a traffic analog.

Simply put, a continuous model of traffic may be neglecting the very factors that give rise to traffic congestion and jamming. The smoothness assumed in the continuous macroscopic approach could be inappropriate for a toll plaza. To address this possibility, we turn to cellular automata theory to develop a discrete, microscopic model. The results of this analysis have the potential either to confirm the results of our continuous approach or perhaps cause us to question the soundness of our initial assumptions.

Approach

In the cellular automata model of toll plaza dynamics, each cell is designated as a vehicle, a vacancy, or a barrier to traffic flow. The model follows individual vehicles through the plaza and computes the time that each one spends waiting. By summing the time spent waiting by all vehicles in the plaza, we arrive at an indication of the plaza's operating efficiency. Using measurements of waiting times, we may optimize the number of booths for any given number of lanes using either of the strategies discussed earlier.

In the cellular automata model, time, position, and vehicle identity are each discrete quantities. Position is specified as a cell or location in a matrix, while time is incremented using a convenient time step. Individual vehicles are created and followed through the plaza. This approach contrasts with that of the macroscopic models, which consider only the overall flow of traffic. By observing individual components of the traffic, the cellular automata model may help elucidate the mechanisms that give rise to congestion associated with tollbooths.

At its essence, the cellular automata model provides a means of creating vehicles, manipulating them through a virtual plaza, and quantifying the efficiency of the plaza configuration. Vehicles progress through the plaza as a function of time and the surrounding geometry. In any given time step, a vehicle may advance forward, change lanes, or sit still. Vehicles enter the plaza from a stretch of road containing a specific number of lanes. As a vehicle approaches the string of tollbooths, the road widens to accommodate the booths (given that there are more tollbooths than lanes). There is a specific delay associated with using a tollbooth. Once a vehicle leaves a booth, it must merge into a roadway containing the original number of lanes. During the course of a vehicle's trip, it may encounter obstacles that temporarily prevent it from moving.

Assumptions

In addition to the general assumptions about toll plazas and traffic flow listed at the beginning of this paper, the cellular automata model requires an additional set of specific assumptions (to be distinguished from governing dynamics rules):

- The plaza consists of three types of cells: occupied cells, vacant cells, and 'forbidden' cells (which represent cells that vehicles are not permitted to occupy – i.e. the boundary of the plaza).

- Cells represent a physical space that may accommodate a standard vehicle and a comfortable buffer region on either side.

- All vehicles are the same size.

We will revisit and reevaluate each of these assumptions in the subsequent discussion section.

Development of Model

In formulating a two-dimensional cellular automata model of toll plaza traffic dynamics, we give considerable attention to the feasibility of computational implementation. As analytical treatment of a two-dimensional automata model is prohibitively difficult, we seek only numerical solution to the task at hand. The following section discusses our formulation of a cellular automata model and presents a general description of the model's implementation. For this model, coding and numerical analysis are done using MatLab software.

A Virtual Toll Plaza

In devising a toll plaza, we accommodate the discrete nature of position in our model. To represent the merging or diverging of lanes, we use a simple stair-step approach like that shown in Figure 6.

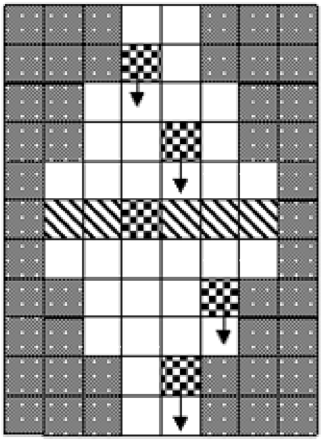
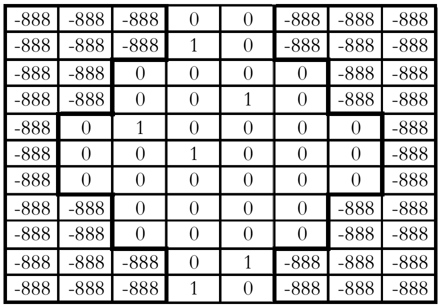


Figure 6: Schematic representation of simple plaza construction. For this case, the plaza contains two lanes and six booths. Solid (dark) cells represent positions that vehicles are not permitted to occupy, while checkered cells represent vehicles, and the striped cells represent tollbooths. The direction of motion is fixed (but arbitrary if the configuration is symmetric). White cells are vacant. Arrows indicate the direction of motion.

To implement this concept in MatLab, we create a large matrix representing the physical system. Occupied cells are labeled with ones while vacant cells are numbered with zeros. For easy visualization, 'forbidden cells' are marked with '-888.' In some cases, occupied cells are flagged and assigned numbers other than one. This will be discussed in more detail later.



-888	-888	-888	0	0	-888	-888	-888
-888	-888	-888	1	0	-888	-888	-888
-888	-888	0	0	0	0	-888	-888
-888	-888	0	0	1	0	-888	-888
-888	0	1	0	0	0	0	-888
-888	0	0	1	0	0	0	-888
-888	0	0	0	0	0	0	-888
-888	-888	0	0	0	0	-888	-888
-888	-888	0	0	0	0	-888	-888
-888	-888	-888	0	1	-888	-888	-888
-888	-888	-888	1	0	-888	-888	-888

Figure 7: Visual representation of simple plaza from Figure 6 built in MatLab. Ones represent occupied cells, zeros represent vacant cells, and '-888' denotes a forbidden

(inaccessible) cell. As a convention, cars in the model traveled from the top of the matrix to the bottom.

The plazas shown above are very simple and do not represent the actual cases simulated. Typically, a long corridor is added to the top and bottom of the system. In this way, vehicles can establish a regular, steady rate before arriving at the tollbooths and after departing from them. Additionally, the larger system minimizes the impact of the boundary conditions on the traffic dynamics near the tollbooths (for example, a line will not become long enough to reach the boundary and interfere with application of the boundary conditions).

To allow for additional flexibility, the steepness of the merging and diverging are left as variables and can be modified from trial to trial. One would expect (as will be shown later) that bottlenecking will be severe for steep merging gradients. If tollbooth lanes end rapidly, vehicles must immediately converge to the central lanes and bottlenecking effects will intensify.

A special case of the plaza configuration occurs when the following is true:

$$\operatorname{mod}(B-L,2)=1$$

where B is the number of tollbooths and L is the number of lanes in the original (and final) roadway. In short, this condition is satisfied when either B or L is even and the other is odd. When this occurs, it is no longer possible to create a symmetric plaza system. The number of stair-steps allocated for merging and diverging will be larger on one side than another. Figure 8 demonstrates such a case:

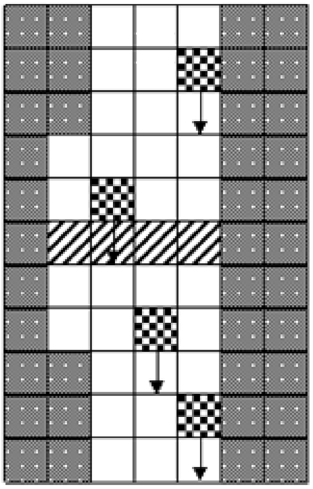


Figure 8: Schematic representation of asymmetric plaza. This example contains four booth lanes but only three normal travel lanes. We resolve issue by using one fewer stair-step on the right side.

As we will show later, the asymmetry of this design is not significant from a practical standpoint. Our results do not appear to be odd-even dependent.

The problem of plaza construction was generalized and implemented using the function 'create_plaza.' This MatLab function creates a plaza (no vehicles) of *B* tollbooth lanes and *L* normal travel lanes. For an explicit reproduction of this function, please consult Appendix III.

Governing Dynamics

With the establishment of a plaza, we next turn to the dynamics within it. As cars move through the toll plaza, we require a governing set of rules to dictate their behavior. The rules are applied to each vehicle to determine which action will be taken during the current time step. For any given toll plaza configuration or geometry, each vehicle has a specific number of options, each with an associated probability.

Just as crucial as the set of rules, however, is the order in which the rules are implemented. A fuzziness on rule application may give rise to confusing or misleading results. There are a number of decisions to consider. For instance, which car gets to act first? How does a vehicle decide to switch lanes? When two vehicles want to occupy the same cell, which is given preference? Such matters could be critical. On the other hand, one must be careful that systematically following a certain rule order does not create some obscure or unintended phenomenon. To perform any kind of meaningful simulation, these issues must be balanced.

A working cellular automata model is presented below. For each time step, the following rules are applied in sequential order:

1. Starting at the front of the traffic and moving backward (with respect to the flow), vehicles are advanced to the cell directly in front of them with probability p. If the next cell is not vacant, the vehicle does not advance and is flagged. Note: This probability is meant to simulate the stop-and-go nature of slowly moving traffic. One could think of p as a measure of driver attentiveness. $p = 1$ corresponds to the case where drivers are perfectly attentive and move forward at every opportunity. $p = 0$ would represent the extreme case where drivers have fallen asleep and they fail to move forward at all.

2. Using an influx distribution function, the appropriate number of new vehicles is randomly assigned to lanes at the initial boundary (see next section).

3. Starting at the front of traffic and moving backward, those vehicles flagged in step 1 are given the opportunity to switch lanes. For each row of traffic, the priority order for switching is determined by a random permutation of the number of lanes. Switching is attempted with probability *q*. If switching is attempted, left and right merges are given equal probability to be attempted first. If a merge in one direction (i.e. left or right) is impossible (meaning that the adjacent cell is not vacant), then the other direction is attempted. If both adjacent cells are unavailable, the vehicle is not moved.

4. Total waiting time for the current time step is computed by determining the number of cells in the system containing a vehicle.

5. The number of vehicles advancing through the far boundary (end of the simulation space) are tabulated and added to the total output. This number is later used to confirm conservation of traffic.

Other notes about the algorithm:

- In step 1, those vehicles in the tollbooth position are not necessarily advanced. Rather, the value of the cell is incremented and the vehicle is not moved. This has the effect of delaying a vehicle being serviced at a tollbooth. The cell for such a vehicle is incremented each time step until it has waited enough time steps to constitute a "service delay." The vehicle is then released. The service delay is typically three to five time steps.

- Most simulations are performed over what amounts to a 24-hr period to reflect the cyclic nature of daily traffic. This choice is examined later and its results are compared to those of a shorter simulation with a higher traffic influx (simulating *only* rush hour).

A simplified flow chart representing the above algorithm is presented in Figure 9:

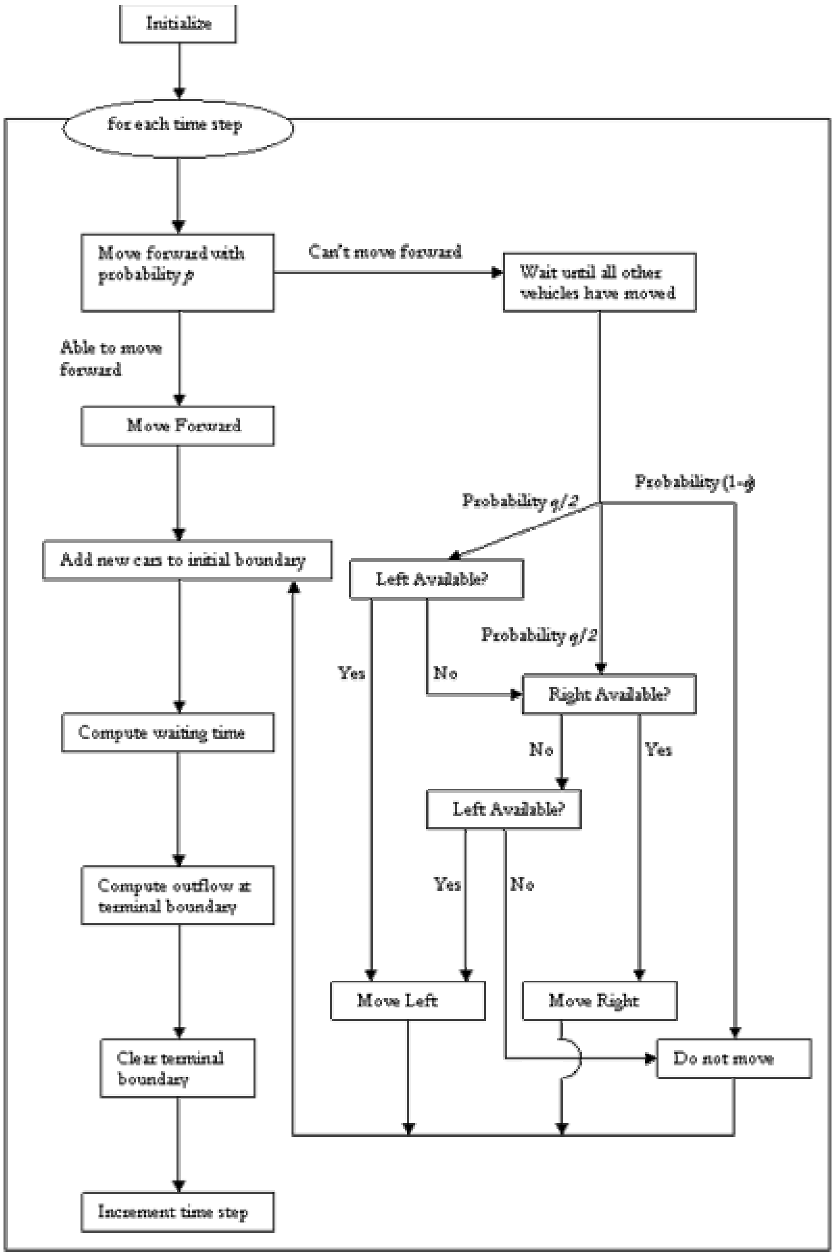


Figure 9: Flow-chart representation of the cellular automata algorithm

Population Considerations

Recall that the daily (cyclic) influx distribution of cars, $F_{in}(t)$, was previously defined using a Fourier series (See Appendix I):

$$F_{in}(t) = a_0 + \sum_{n=1}^{N} a_n \cos(n \cdot \omega \cdot t) + \sum_{n=1}^{N} b_n \sin(n \cdot \omega \cdot t)$$

This method is still valid for the automata model, but the influx values must be scaled to reflect the effective influx over a much smaller time interval (a single time step). The modified influx function, $F_{in}'(\tau)$, is computed as follows:

$$F_{in}^{'}(\tau) = \min\left[\text{round}\left(\frac{F_{in}(t)}{\eta}\right), L\right]$$

where η is a constant factor required for the conversion from units of *t* to those of τ and L is the number of initial travel lanes. A 'round( )' function is employed to round the argument to the nearest integer (recall that the influx per time step in the automata model must be an integer number of vehicles not exceeding the number of available lanes).

When this computation is carried out, we obtain the following distributions:

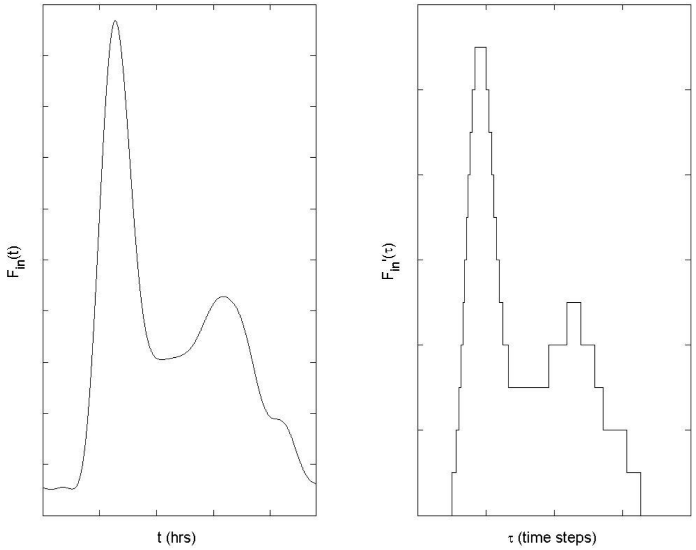


Figure 10: Plots of continuous influx ($F_{in}(t)$ – left) and normalized cellular automata model influx ($F_{in}'(\tau)$ – right).

Boundary Conditions

To simulate the automata model, we must define boundary conditions for the plaza. Although plaza model is two-dimensional, the motion is essentially one-dimensional. The barriers on the sides of the plaza prevent the cars from occupying those cells. In a sense, thus, the boundary conditions for the sides of the plaza parallel to the roadway might be thought of as analogous to either Neumann or Dirichlet (with respect to the vehicles). That is, the vehicles may neither occupy the boundary nor traverse the boundary to leave or enter the system, respectively.

The boundary conditions for the boundaries perpendicular to the direction of traffic flow are defined differently. As visited in the previous section, there is a time-dependent influx associated with the toll plaza. At the boundary where vehicles begin their trip through the plaza, new vehicles can be introduced at each time step. The number of cars introduced is given by the value of the influx function for the corresponding time step. Once the number of incoming vehicles is determined, the vehicles are distributed randomly among the available lanes. Thus, the number of incoming vehicles per time step can never exceed the number of highway lanes in the system.

At the system boundary where vehicles exit the plaza, the boundary condition acts to remove all vehicles that entered the boundary row of cells since the previous time step. In other words, the final row of cells acts as a perfect absorber – every vehicle that reaches it is immediately removed and the output count is incremented. Recall that the output count is used to confirm the conservation of traffic principle. By removing vehicles before a new time step begins, the boundary clears the way for the next row of vehicles to leave the system. This prevents a buildup of vehicles at the system's departure zone.

Computing Wait Time

Wait time is determined by looking through the entire matrix at each time step and noting the number of cells with positive values. Recall that the only cells containing positive values are those representing vehicles. Thus, by counting the number of vehicles in the plaza at any given time, we are also counting the amount of time spent by vehicles in the plaza (in units of time steps).

At time step *i*, total cumulative waiting time is computed as follows:

$$\begin{aligned}&\forall x,\ \forall y\\ &W_i = W_{i-1} + 1\left(plaza(x,y) > 0\right)\end{aligned}$$

where '1( )' denotes an indicator function and '*plaza*' denotes the matrix of cells.

Simulation and Results

To determine the optimal number of tollbooths for a given number of highway lanes, the cellular automata simulation is run for relevant combinations of the two. Recall one of our general assumptions is that the number of tollbooths in any plaza is at least equal to the number of highway lanes that are feeding it. An optimal tollbooth number is selected for a given number of highway lanes when the system cost optimization method discussed earlier is applied.

Recall the cost optimization method defines total cost as follows:

$$C_{total} = \alpha \cdot \gamma \cdot N \cdot W(B, L) + B \cdot Q$$

Using the cellular automata model, we compute waiting time as a function of both the number of lanes and the number of tollbooths. For a fixed *L*, we compare all values of C_{total} and choose the lowest one. The results of this method are presented in Table 6.

Table 6: Optimization for Cellular Automata Model

	Optimal # Booths	
Highway Lanes	**Typical Day**	**Rush Hour**
1	2	2
2	4	4
3	5	6
4	7	7
5	8	9
6	10	11
7	12	13
8	14	15
16	27	29

As indicated in Table 6, there is fairly good agreement between the recommended number of booths for a typical day and for peak hours. However, we note that the optimal booth number for a typical day never exceeds that for rush hour. Rush hour seems to require slightly more booths than a typical day in order for the plaza to operate most efficiently.

Each value in Table 6 is representative of approximately 20 trials. Through these trials, we noted a remarkable stability in our model. Despite the stochastic nature of our algorithm, each number of lanes was almost always optimized to the same number of tollbooths. There were a handful of exceptions; they occurred exclusively for small numbers of highway lanes (< 3 lanes). Integer values are presented in Table 6 only because fractional tollbooths have no physical meaning.

Example

As an example of one optimization using the cellular automata model, let us consider the instance of six highway lanes. For comparison, the analogous optimization is carried out previously in models 1 and 2.

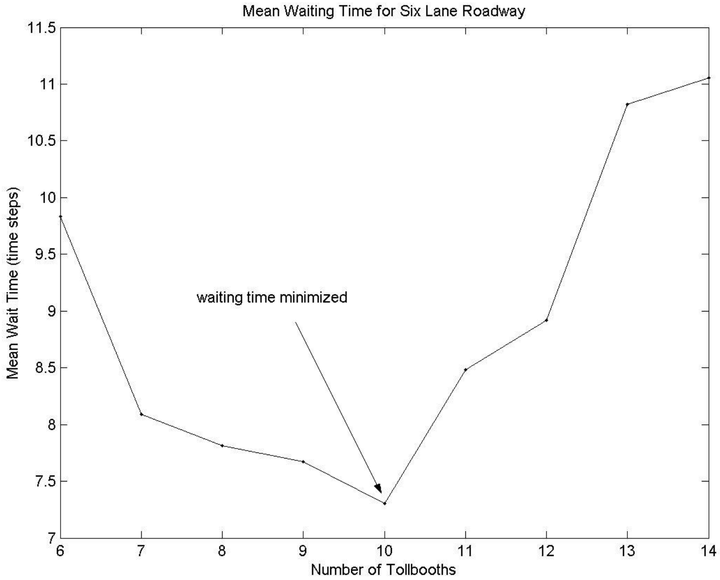


Figure 11: Minimization of mean waiting time for six lane roadway. Use of 10 tollbooths minimizes the mean wait for customers.

Figure 11 is created by running the simulation repeatedly for six lanes and varying the number of tollbooths. A choice of ten tollbooths provides the lowest mean wait time for vehicles. However, ten is not necessarily the optimal number of tollbooths for the system. To determine the optimal solution, we must refer to the cost optimization function developed previously.

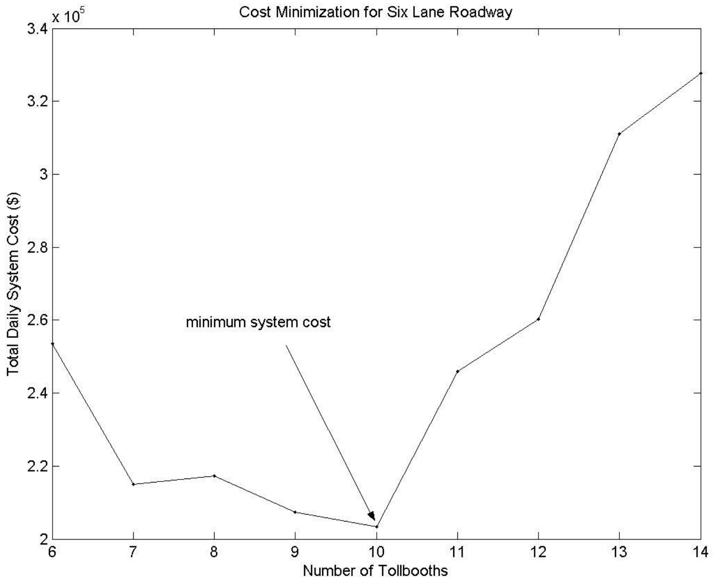


Figure 12: Minimization of total daily system cost for six lane roadway. Use of 10 tollbooths minimizes both the mean wait for customers and the system cost.

As seen in Figure 12, ten tollbooths minimizes system cost as well as mean waiting time (Figure 11). Thus, ten tollbooths is the optimal number for a toll plaza with six incoming lanes (given our selection of parameters). Although the curves in Figures 11 and 12 look very similar, they are indeed more than scalar multiples. One notes that the differences between the two curves is most pronounced at the two ends.

Discussion of Cellular Automata Traffic Model

Evaluation of Assumptions

Let us now consider the assumptions made in the development of the cellular automata traffic model. In what way have these assumptions been either confirmed or discredited? Has there been an assumption which has proven to be particularly limiting?

- We first assumed that the plaza contains only three types of cells – occupied, vacant, and forbidden. In fact, there are two other kinds of cells (flagged cells and incrementing booth cells). These arose as artifacts of the nature of the computer program and did not affect the dynamics of the system. Although treating the plaza in this simplified manner may have neglected some details, it was necessary to develop the framework for a cellular automata based simulation. The model could possibly be improved by adding detail (via additional cell types), but new features – unless dramatic – would probably not change the fundamental behavior of the system.

- Our next assumption was that cells represent a physical space that may accommodate a standard vehicle and a comfortable buffer region on either side. Again, this was a simplifying assumption designed to accommodate the use of only three cell types. However, it is not a bad assumption if when we recall that vehicles are typically required to move slowly within toll plazas (< 15 mph) and probably maintain reasonable buffer zones between adjacent vehicles. We have no clear way of evaluating whether this assumption limited our model in some important way, but we suspect not.

- Next, we assumed the presence of only one type of vehicle within the system. More specifically, we assumed that all vehicles are the same size. The use of larger vehicles that occupied multiple cells was not explored and represents one possible extension of this study. We doubt that larger vehicles interspersed in the plaza would dramatically affect traffic flow, but the possibility exists.

Sensitivity Analysis

In this section, we briefly evaluate the sensitivity of our cellular automata model. Does perturbing the value of any of our model parameters have a notable effect on the resulting solution?

To answer this question, we make changes to three key variables in our system and observe the corresponding influence on the optimized number of tollbooths. Ideally, we would hope that the optimal number of tollbooth is unchanged for even modest variations in these parameters.

The parameters we choose to modify are *p* (probability of advancement), 'delay' (number of time steps required to serve a vehicle in a tollbooth), and *q* (the probability that a flagged vehicle opts to attempt a turn). The results of this analysis are presented in Table 7. Since we have used six lanes as our standard test case, we continue with this choice here.

Table 7: Sensitivity Analysis for Cellular Automata Model (L = 6)

p	q	Delay	**Optimal # Booths**
0.9	0.95	4	10
0.8	0.95	4	10
1.0	0.95	4	10
0.9	0.90	4	10
0.9	1.00	4	10
0.9	0.95	5	11
0.9	0.95	3	10

As indicated in Table 7, our cellular automata model is relatively insensitive to both *p* and *q*. Changes of ± 11% and ± 5.2% in *p* and *q*, respectively, had no effect on the optimal number of tollbooths for a six lane highway. On the other hand, increasing the delay time by 25% shifted the optimal number of booths from 10 to 11 (10%). Decreasing the delay by 25% had no effect on the solution. Perhaps additional work could lead to an elucidation of the relation between delay and optimal booth number that could help stabilize the cellular automata model.

Rush Hour Dynamics

An important test of our model was a simulation of rush hour *only*. Previously, we have considered traffic behavior over the course of individual days. By doing this, we may risk obscuring important behavior that occurs only during peak hours. With this in mind, we use the cellular automata model to simulate a three hour period in which the influx of traffic nearly saturated the lanes.

The results of the rush hour analysis seem to confirm the robustness of our model. Despite the extreme influx of vehicles, the optimal number of tollbooths never differed by more than one from our corresponding normal case values.

Conservation of Traffic

With a computational model such as the one presented here, one must be careful that small logic errors or programming bugs do not give rise to unexpected phenomena. As one check of the cellular automata traffic algorithm, we call upon the principle of conservation of traffic. As its name implies, this rule dictates that no vehicles are gained or lost during the course of our simulation. Assuming all vehicles are allowed to traverse the plaza, we must count the same number of vehicles leaving the plaza as we permitted to enter the plaza. Indeed, for all 24-hour and rush hour simulations in this trial, it was confirmed that the corresponding input and output vehicle counts were equal. This confirmation lends credibility to our model.

Comparison of Results from the Models

Again, we found the following results for optimal number of booths given 1, 2, 3, 4, 5, 6, 7, 8, or 16 lanes approaching a toll plaza:

Table 8: Comparison of final recommendations for three models

Lanes	Car-Tracking	Macroscopic	Automata
1	4	3	2
2	5	5	4
3	7	6	5
4	8	7	7
5	10	9	8
6	12	11	10
7	13	12	12
8	16	14	14
16	29	27	27

From descriptions of our models, we first know that the Basic Car-Tracking model serves as an upper bound for the optimal number of booths, due to its omission of bottlenecking. As we can see in the data above, each value of *B* under the Car-Tracking column is greater than or equal to its counterparts in the Macroscopic Model for Cost Minimization and Cellular Automata Model. In this respect, our consideration of the simple and somewhat uninteresting Car-Tracking model is complete.

The Macroscopic Model for Cost Minimization and the Cellular Automata Model provide us with very similar results for an optimal number of tollbooths, considering a given number of lanes entering a toll plaza. Despite the overwhelming differences in methodology and theory behind these two models, the number of times they coincide or differ by only one booth is astounding. It almost goes so far as to say that precision implies accuracy here.

We conjecture that the reason the Macroscopic Model recommends a greater number of tollbooths than the Cellular Automata Model due to consideration of bottlenecking. Whereas the Macroscopic Model takes bottlenecking into account, we use it merely to say that there is an arbitrary threshold above which bottlenecking occurs. While we use reasonable factors to create the threshold of outflux over which bottlenecking occurs, this value is likely to be an overestimate, thus underestimating the amount of bottlenecking. Since the Macroscopic Model undervalues bottlenecking, it will recommend a higher number of booths as optimal.

On the other hand, the Cellular Automata Model maintains the strictest observation of the complex process of bottlenecking. Since this model details the toll plaza geometry, especially the lane merging after the tollbooths, it has the makings of a very accurate merging demonstration. Furthermore, the Cellular Automata Model uses such microscopic detail to count exactly the amount of time spent waiting in line by each car. Due to its examination of each car and each period waited, we lean more toward the Cellular Automata Model for a more accurate representation of number of booths versus lanes than toward the Macroscopic Model for Cost Minimization.

Table 9: Comparison of Linear Fit Parameters for our Models

Model	Slope	Intercept	R^2
Basic Car Tracking	1.699	1.738	0.997
Macroscopic	1.598	1.215	0.998
Cellular Automata	1.672	0.228	0.998

Table 9 shows linear fit parameters for all three models. Note that all three models are well described by a linear equation.

One Tollbooth for Each Lane

General Consideration

Now, we consider the implications of introducing no extra tollbooth lanes to the plaza. Each highway lane is leads to a single tollbooth. This scenario is represented for four lanes in Figure 12:

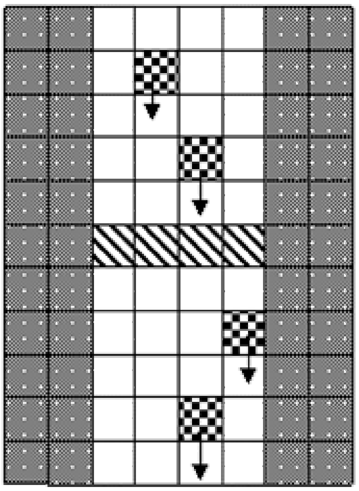


Figure 12: Schematic representation of four lane case with 1:1 ratio of tollbooths to highway lanes.

In the case of a 1:1 ratio of tollbooths to lanes, we observe that the plaza does not fan out in the center. The toll plaza has no extra tollbooth lanes to relieve traffic congestion. What are the consequences of this scenario? Certainly, it would appear that traffic congestion would increase and a line would build up behind the tollbooths. In this case, one notes that the waiting time by customers comes *exclusively* from waiting in line. That is, once vehicles pass through the toll plazas, there is no waiting time associated with bottlenecking. There can be no bottlenecking because the lanes do not merge. Thus, we see that although one component of the customers' waiting time is increased (waiting in line), another component is completely eliminated (bottlenecking).

The previous discussion focused on times in which the influx to the toll plaza was too great for the tollbooths to prevent formation of a line. What if vehicles enter less frequently? In this case, we note that line formation is not an issue (and bottlenecking remains a non-factor). Thus, it would seem that a 1:1 ratio of booths to lanes is ideal when the highway traffic is quite low.

What other factors may allow a simple 1:1 configuration to be more effective than its multi-boothed counterparts? Since bottlenecking is eliminated in the 1:1 case, any change that has the effect of reducing line length will tend to favor the 1:1 case over a case with bottlenecking.

Line length can be reduced in a number of ways:

- Increase the rate at which tollbooths are able to service vehicles. If vehicles move through a tollbooth more quickly, it will take a greater influx of vehicles to generate a line.

- Increase driver attentiveness. Recall that in the cellular automata model, drivers are each assigned a probability *p* of moving forward during each time step. If *p* is increased, drivers will move forward with greater regularity and congestion is likely to be a factor near the lines.

Changing the above parameters in the opposite ways would have the effect of favoring the 'current' case (> 1 tollbooth per incoming lane) over the basic 1:1 case.

Quantitative Estimation

For the cellular automata model, an immediate reduction in cost often is realized upon adding only a single extra tollbooth to a 1:1 plaza configuration (by 1:1, we mean one tollbooth for each incoming lane). This cost reduction is typically continued until a local minimum is reached – the number of tollbooths corresponding to this minimum is frequently the optimal number of tollbooths (which we will designate *B**). As additional tollbooths are added to the plaza configuration, the cost function typically increases at approximately the same rate as it fell before reaching *B**. For cases such as this, we may write a simple expression for the number of tollbooths at which the cost function reaches a value comparable to that for the 1:1 configuration, which we designate Ψ:

$$\Psi = 2 \cdot B^* - L$$

where L is the number of lanes feeding the toll plaza. In this expression, Ψ represents the number of tollbooths at which the benefit of added tollbooths in terms of line length is balanced by the corresponding increase in bottlenecking. In other words, if the number of tollbooths exceeds Ψ, the cost of the system increases beyond what it would have been if *no* tollbooths had been added.

Certain data from the macroscopic model indicate that the effect of bottlenecking is never sufficient to counterbalance the initial cost reduction from adding tollbooths. Such an example is provided in Figure 13.

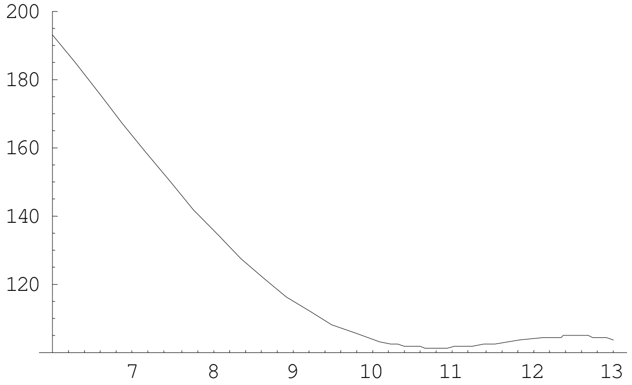


Figure 13: Total wait time vs number tollbooths. Data from macroscopic model illustrate that bottlenecking is not sufficient to incur same cost as 1:1 configuration (Note L = 6).

Briefly, both of our bottlenecking models agree that the first few tollbooths added to a plaza with a 1:1 configuration will reduce waiting time and system costs. However, much of our data suggest that these initial gains achieved by reducing line length are not ever counterbalanced by the effect of bottlenecking. There exist some counterexamples in which Ψ may be estimated using the above method.

Conclusion

We used three models – the Basic Car-Tracking Model Without Bottlenecks, the Macroscopic Model for Total Cost Minimization, and the Cellular Automata Model – in order to determine the optimal (per our definition) number B of tollbooths required in a toll plaza of L lanes.

In short, the Basic Car-Tracking Model uses a simple orderly lineup of cars approaching tollbooths and ignores bottlenecking after the tollbooths. While a quick model, it does omit bottlenecks, and provides us with a strong upper bound on B for any given L. Cost analysis on this model was not as effective as threshold analysis, and we determined an optimal B by recognizing when additional tollbooths did not decrease waiting time significantly.

The Macroscopic Model looks at the motion of traffic as a whole, rather than individual models. It tabulates waiting time in line before the tollbooths by considering times when traffic influx into the toll plaza is greater than tollbooth service time. It also finds bottlenecking time by assuming there exists a threshold of outflux, above which bottlenecks will occur, and notices when outflux is greater than said threshold. This is a much more accurate model than the Car-Tracking Model, and it provides us with reasonable solutions for *B* in terms of *L*.

The Cellular Automata Model looks at individual vehicles, and their "per lane length" motion on a toll plaza made up of cells. With a probabilistic model of how drivers advance and change lanes, this model far better details the waiting time in line and the bottlenecking after the tollbooths than the previous models. It was through this detail that we decide that this model is likely to be most accurate.

Thus we decide to recommend values closer to those provided by the automata model than the macroscopic one. In order to write *B* explicitly in terms of *L*, we invoke the linearity of the chart shown in the Comparison of Results. Also, in order to preserve integral values for *B*, we use the floor function and determine that $B = [\![1.65L + 0.9]\!]$, where $[\![x]\!]$ is the greatest integer less than *x*.

Table 10: **Final Results and Recommendations**

Lanes	Car-Tracking	Macroscopic	Automata	Recommendation
1	4	3	2	2
2	5	5	4	4
3	7	6	5	5
4	8	7	7	7
5	10	9	8	9
6	12	11	10	10
7	13	12	12	12
8	16	14	14	14
16	29	27	27	27

Potential Extension and Further Consideration

Our models assume that each booth is identical to any other. In recent years, however, systems such as E-ZPass, which allow a driver to electronically pay a toll from an in-car device without ever slowing down to stop at a booth window, have been increasingly prevalent. If all E-ZPass booths also double as regular teller-operated booths, much of our models remain the same, except the average service rate might be increased. The same effects would be similar upon introduction of electronic coin collectors for those drivers with exact change. The trouble comes when all the booths are not the same and drivers may need to change lanes upon entering the plaza. This directed lane changing was not implemented in any of the models presented here, but could easily become a part of the automata model. Exclusive E-ZPass booths also would drastically reduce the operating cost for the booth, since an operator's salary would not need to be paid (from $180,000 to $16,000 annually) [Sullivan 1994].

References

Boronico, Jess S., and Philip H. Siegel. "Capacity planning for toll roadways incorporating consumer wait time costs." *Transportation Research A* (May 1998) 32 (4): 297-310.

Daganzo, C.F., et al. 1997. "Causes and effects of phase transitions in highway traffic." *ITS Research Report* UCB-ITS-RR-97-8 (December 1997).

Gartner, Nathan, Carroll J. Messer, and Ajay K. Rathi. 1992. *Traffic Flow Theory: A State of the Art Report.* Revised monograph. Special Report 165. Oak Ridge, TN: Oak Ridge National Laboratory. `http://www.tfhrc.gov/its/tft/tft.htm`.

Gelenbe, E., and G. Pujolle. 1987. *Introduction to Queueing Networks.* New York: John Wiley & Sons.

Jost, Dominic, and Kai Nagel. 2003. "Traffic jam dynamics in traffic flow models." Swiss Transport Research Conference. `http://www.strc.ch/Paper/jost.pdf`. Accessed 4 February 2005.

Kuhne, Reinhart, and Panos Michalopoulos. 1992. "Continuum flow models." Chapter 5 in Gartner et al. [1992].

Schadschneider, A., and M. Schreckenberg. "Cellular automaton models and traffic flow." *Journal of Physics A: Mathematical and General* (1993) 26: L679-L683.

Sullivan, R. Lee. "Fast lane." *Forbes* (4 July 1994) 154: 112-115.

Tampere, Chris, Serge P. Hoogendoorn, and Bart van Arem. "Capacity funnel explained using the human-kinetic traffic flow model." `http://www.kuleuven.ac.be/traffic/dwn/res2.pdf`. Accessed 4 February 2005.

Appendix I: Fourier Series

Hour	Influx (cars/min)	Fourier Approx of Influx	% Error
0.5	15.44	15.16272478	1.796
1.5	15.32	15.42467822	0.683
2.5	15.16	15.18796896	0.184
3.5	19.9	19.81853474	0.409
4.5	47.09	47.22251986	0.281
5.5	89.95	89.61825869	0.369
6.5	105.9	106.4828683	0.550
7.5	85.52	84.72959878	0.924
8.5	54.68	55.57942216	1.645
9.5	43.11	42.42662327	1.585
10.5	40.16	40.49538486	0.835
11.5	40.85	40.83544106	0.036
12.5	41.72	41.63346483	0.207
13.5	44.54	44.44085865	0.223
14.5	48.88	49.29448007	0.848
15.5	53.2	52.55619485	1.210
16.5	51.61	52.21058951	1.164
17.5	48.38	48.16410937	0.446
18.5	39.72	39.50374966	0.544
19.5	30.51	31.11397219	1.980
20.5	29.48	28.86864636	2.074
21.5	26.82	27.196867	1.405
22.5	21.21	21.2608522	0.240
23.5	17.22	16.91795178	1.754

The equation for the Fourier Series approximation of the data table above is as follows:

$$\begin{aligned} F(t) &= 41.68 - 16.38\cos(t\omega) + 12.53\sin(t\omega) - 18.59\cos(2t\omega) + 0.6307\sin(2t\omega) + 3.572\cos(3t\omega) - \ldots \\ &- 13.67\sin(3t\omega) + 7.876\cos(4t\omega) + 0.4378\sin(4t\omega) - 0.5048\cos(5t\omega) + 6.93\sin(5t\omega) - 2.97\cos(6t\omega) + \ldots \\ &+ 0.4869\sin(6t\omega) + 0.2518\cos(7t\omega) - 1.554\sin(7t\omega) + 0.5785\cos(8t\omega) - 0.5871\sin(8t\omega), \end{aligned}$$

and it has an R^2 value of 0.9997, which represents an excellent fit to the data. We can also see that this is true because only one of our 24 percent error values is above 2%.

The following chart shows the values of the coefficients a_0 through a_8 and b_1 through b_8. In the latter two columns, we can see the lower and upper bounds for a 95% confidence level for the coefficients. The tightness of the bounds again shows how accurate this Fourier fit is.

Coefficient	Value	Lower Bound*	Upper Bound*
a_0	41.68	41.59	41.76
a_1	-16.38	-16.5	-16.26
a_2	-18.59	-18.71	-18.47
a_3	3.572	3.419	3.725
a_4	7.876	7.758	7.995
a_5	-0.5048	-0.6503	-0.3594
a_6	-2.97	-3.088	-2.852
a_7	0.2518	0.1312	0.3725
a_8	0.5785	0.4601	0.6968
b_1	12.53	12.41	12.55
b_2	0.6307	0.4814	0.7799
b_3	-13.67	-13.79	-13.55
b_4	0.4378	0.2991	0.5765
b_5	6.93	6.811	7.049
b_6	0.4869	0.3613	0.6126
b_7	-1.554	-1.673	-1.435
b_8	-0.5871	-0.7069	-0.4674
ω	0.2513	0.2513	0.2514

Appendix II: Basic Car-Tracking Model Code

```
t = 86400/5+1; %number of 5 second periods in a day
lanes = 6;    %number of lanes, assume lanes <= booths
booths = 10; %number of booths... assume booths << cars
i = 61565 * lanes/4;  %number of cars... 61565 found through integration of the influx
equation

inrate = 0;
outrate = 0;
S=0;
A=0;
L=0;
B=0;
Start=0;
N=0;
E=0;
last=0;
L2=0;
L3=0;

S(1:i) = 0;          %service duration time for car i
A(1:i) = 0;          %arrival time of car i
```

```
L(1:i) = 0;          %leaving time of car i
B(1:i) = 0;          %tollbooth used by car i
Start(1:i) = 0;      %service start time for car i
last(booths) = 0; %last time at which a car left a particular booth
inrate(1:t) = 0;     %influx at time t
L2(1:t) = 0;         %number of cars leaving tollbooths at time t
L3(1:t) = 0;         %smoothed our version of L2
outrate(1:24) = 0;  %rate of outflux
N(1:t) = 0;          %number of cars in line at time t

h = linspace(0,24,17281); %indexing vector for each 5 second time period
a0 = 41.68;
a = [-16.38, -18.59, 3.572, 7.876, -0.5048, -2.97, 0.2518, 0.5785];
b = [12.53, 0.6307, -13.67, 0.4378, 6.93, 0.4869, -1.554, -0.5871];
omega = 0.2513; %all Fourier Coefficients
inrate = a0 * ones(size(h));
for n = 1:8
        inrate = inrate + a(n).*cos(n.*h.*omega) + b(n).*sin(n.*h.*omega);
end
inrate = inrate * lanes / 4; %scaling lanes by appropriate amount
A(1) = 12/inrate(1);  %arrival time of car 1 in terms of inrate
for j = 2:i
        k = floor(A(j-1));
        if k == 0
                k = 1;
        end
        A(j) = A(j-1) + 12/inrate(k);          %arrival time of car i in terms of inrate
end

mu = 2.4;            %mean service duration time in 5-second periods
S = exprnd(mu,1,i); %service time as an exponential random variable

for j = 1:i
   for k = 1:booths
      if (last(k) == min(last))
         B(j) = k;  %find booth that was/will be emptied soonest
      end
   end
   if A(j) > last(B(j))         % if there is an empty booth, then...
      Start(j) = A(j);          % start right away
      L(j) = Start(j) + S(j);
      last(B(j)) = L(j);
   else                                      %if not...
      Start(j) = last(B(j));  %start once the soonest one becomes available
      L(j) = Start(j) + S(j);
      last(B(j)) = L(j);
   end

end

for j = 1:i
```

```
   k = ceil(L(j));
   if k > t
      k = t;
   end
   for m = ceil(A(j)):k
      N(m) = N(m) + 1;    %counts the number of people in line
   end
end

W = L - (A + S); %waiting time is line time - (arrival time + service time)

for k = 1:i
           if L(k) <= t
                      L2(ceil(L(k))) = L2(ceil(L(k))) + 1;  %creation of L2, outflux from
tollbooths
           end
end
k = length(W);

for j = 1:i
   if W(j) == 0
      k = k-1;
   end
end

totalavgwait = sum(W)/i/12;
carsavgwait = sum(W)/k/12;
maxwait = max(W)/12;
```

Appendix III: Cellular Automata Model Code

cellular.m

```
clear all;
W = 0;
for j = 0:7
   B = 6+j;  %number booths
   L = 6;  %number lanes in highway before and after plaza
   T = 1; % # hrs to simulate
   global plazalength;
   plazalength = 101;
   plaza = create_plaza(B,L);
   entry_vector = create_entry(T,L);
   waiting_time = 0;
   output = 0;
   for i = 1:T*1440
      plaza = move_forward(plaza); %move cars forward
```

```
        plaza = new_cars(B, L, plaza, entry_vector(1,i));     %allow new cars to enter
        plaza = switch_lanes(plaza); %allow lane changes
        waiting_time = waiting_time + compute_wait(plaza); %compute waiting time during timestep i
        output = output + compute_output(plaza);
        plaza = clear_boundary(plaza);
    end
    plaza;
    W(j+1) = waiting_time;
end
```

move_forward.m

```
function new = move_forward(old)
new = old; %create new plaza looking same as old
[L, W] = size(new); %get its dimensions
prob = .7;
delay = 3;
%%%DOWNSTREAM OF TOLL BOOTHS %%%
for i = (L-1):-1:((L - 1)/2 + 1)
    for j = 1:W
        if new(i,j) == 1
            if new(i+1, j) ~= 0
                new(i,j) = -2;
            end
            if new(i+1, j) == 0
                if prob >= rand
                    new(i,j) = 0;
                    new(i+1, j) = 1;
                end
            end
        end
    end
end
%%AT TOLL BOOTHS %%
for i = (L-1)/2
    for j = 1:W
        if new(i,j) > 0
           if new(i,j) == delay
                new(i,j) = 0;
                new(i+1,j) = 1;
            end
            if new(i,j) ~= delay
                if new(i,j) ~= 0
                    new(i,j) = new(i,j) + 1;
                end
            end
        end
    end
end
%% UPSTREAM OF TOLL BOOTHS %%
for i = (L-1):-1:1
    for j = 1:W
        if new(i,j) == 1
            if new(i+1, j) ~= 0
                new(i,j) = -2;
            end
            if new(i+1, j) == 0
                if prob >= rand
                    new(i,j) = 0;
                    new(i+1, j) = 1;
                end
            end
        end
    end
end
```

new_cars.m

```
function new = new_cars(B, L, old, entry)

new = old;
if entry > 0
    if entry <= L
        x = randperm(L);
        y = ceil((B-L)/2+1);
        for i = 1:entry
            new(1, (y + x(i))) = 1;
        end

    end
    if entry > L
        y = ceil((B-L)/2+1);
        for i = 1:L
            new(1,(y + i)) = 1;
        end
    end
end
```

compute_wait.m

```
function time = compute_wait(plaza)
```

```
[a,b] = size(plaza);
time = 0;
for i = 1:a
   for j = 1:b
      time = time + (plaza(i,j) > 0);
   end
end
```

create_plaza.m

```
function plaza = create_plaza(B, L)

global plazalength;
topgap = 5;
bottomgap = 1;

plaza = zeros(plazalength,B+2);

if mod(B-L,2)==0
   for row = 1:plazalength
      plaza(row,1) = -888;
      plaza(row,2+B) = -888;
   end
   for col = 2:B/2 - L/2 + 1
      for row = 1:(plazalength-1)/2 - topgap * (col-1)
         plaza(row,col) = -888;
         plaza(row,B+3-col) = -888;
      end
   end
   for col = 2:B/2 - L/2 + 1
      for row = (plazalength+3)/2 + bottomgap*(col-1):plazalength
         plaza(row,col) = -888;
         plaza(row,B+3-col) = -888;
      end
   end
else

   for row = 1:plazalength
      plaza(row,1) = -888;
      plaza(row,3+B) = -888;
   end
   for col = 2:(B+1)/2 - L/2 + 1
      for row = 1:(plazalength-1)/2 - topgap * (col-1)
         plaza(row,col) = -888;
         plaza(row,B+4-col) = -888;
      end
   end
```

```
    for col = 2:(B+1)/2 - L/2 + 1
        for row = (plazalength+3)/2 + bottomgap*(col-1):plazalength
            plaza(row,col) = -888;
            plaza(row,B+4-col) = -888;
        end
    end
    for row = 1:plazalength
        plaza(row,2+B) = -888;
    end
end
```

switch_lanes.m

```
function new = switch_lanes(old)
new = old;
prob = 0.8;
x = rand;
y = rand;
[L,W] = size(new);
for i = (L-1):-1:1
    for j = 2:(W-1)
        if new(i,j) == -2
            if x < prob %chance turn will be made
                if y > 0.5 %will attempt left
                    if new(i, j-1) == 0
                        new(i, j-1) = 1;
                        new(i, j) = 0;
                    elseif new(i, j+1) == 0
                        new(i, j+1) = 1;
                        new(i,j) = 0;
                    elseif new(i,j) == -2
                        new(i,j) = 1;
                    end
                end
                if y <= 0.5 %will attempt right
                    if new(i, j+1) == 0
                        new(i,j+1) = 1;
                        new(i,j) = 0;
                    elseif new(i, j-1) == 0
                        new(i, j-1) = 1;
                        new(i,j) = 0;
                    elseif new(i,j) == -2
                        new(i,j) = 1;
                    end
                end
            end
            if x >= prob
                new(i,j) = 1;
            end
        end
    end
end
```

compute_output.m

```
function count = compute_output(plaza)
count = 0;
[a, b] = size(plaza);
for j = 1:b
        count = count + (plaza(a,j) > 0);
end
```

clear_boundary.m

```
function plaza = clear_boundary(input)
plaza = input;
[a,b] = size(plaza);
for i = 1:b
    if plaza(a,i) > 0
        plaza(a,i) = 0;
    end
end
```

create_entry.m

```
function entry = create_entry(T,L)
k = linspace(0,T,34560);
a0 = 41.68;
entry = a0.*ones(size(k));
a = [-16.38, -18.59, 3.572, 7.876, -.5048, -2.97, 0.2518, 0.5785];
b = [12.53, 0.6307, -13.67, 0.4378, 6.93, 0.4869, -1.554, -0.5871];
omega = 0.2513;
for n = 1:8
    entry = entry + a(n).*cos(n.*k.*omega) + b(n).*sin(n.*k.*omega);
end
k = k.*1440;
entry = entry./24.*3;
entry = round(entry);
%%% FOR RUSH HOUR SIMULATION %%%
% k = linspace(0,T,T.*60.*24);
% entry = zeros(size(k));
% entry(1:2:length(k)) = L;
```

APPENDIX IV: Model 2 Outflux [cars/min] for L=6

Time	B=6	B=7	B=8	B=9	B=10	B=11	B=12	B=13
0	7.57	7.58	7.6	7.57	7.57	7.57	7.55	7.6
1	7.7	7.68	7.65	7.72	7.7	7.72	7.7	7.68
2	7.67	7.68	7.68	7.63	7.65	7.67	7.68	7.6
3	10.38	10.37	10.4	10.38	10.42	10.38	10.4	10.4
4	23.73	23.78	23.88	23.85	23.83	23.87	23.87	23.93
5	30.7	34.28	39.53	41.58	44.05	44.15	44.07	44.03
6	30.75	34.67	40.28	45.95	51.03	52.32	52.43	52.38
7	30	35.13	39.56	44.87	43.75	42.27	42.33	42.32
8	29.98	33.83	39.25	34.68	28.25	28.37	28.23	28.22
9	29.85	34.77	29.87	21.47	21.45	21.38	21.43	21.47
10	30.23	33.03	20.27	20.27	20.32	20.27	20.33	20.3
11	30.57	23.62	20.48	20.4	20.38	20.5	20.45	20.43
12	30.88	20.83	20.83	20.85	20.85	20.82	20.75	20.85
13	29.53	22.25	22.25	22.23	22.3	22.22	22.32	22.25
14	24.57	24.65	24.6	24.63	24.57	24.62	24.62	24.63
15	26.2	26.13	26.2	26.2	26.17	26.15	26.18	26.17
16	26.05	26.08	26.02	26.05	26.07	26.13	26.07	26.03
17	23.87	24.02	24.02	23.97	23.95	23.93	23.93	23.92
18	19.9	19.73	19.73	19.8	19.85	19.83	19.75	19.9
19	15.75	15.73	15.73	15.72	15.7	15.7	15.75	15.7
20	14.45	14.45	14.48	14.43	14.45	14.47	14.45	14.43
21	13.47	13.48	13.47	13.52	13.48	13.48	13.5	13.45
22	10.67	10.72	10.72	10.68	10.7	10.68	10.72	10.77
23	8.55	8.5	8.5	8.52	8.55	8.53	8.5	8.52