For office use only	Team Control Number	For office use only
T1 ____________	**71042**	F1 ____________
T2 ____________		F2 ____________
T3 ____________	Problem Chosen	F3 ____________
T4 ____________	**B**	F4 ____________

2017
Mathematical Contest in Modeling (MCM) Summary Sheet

Multi-layer framework for modeling the merging pattern

Summary

A multi-layer State-based framework has been proposed in order to provide our unique insight into the car "merging" problem in the toll plaza. With the aim of reducing the accident rate and improving the throughout rate in the merging area, we consider the problem from the views of three individual lavel, intermediate level, and top level.

Individual level mainly characterized the microcopic traffic performance in the merging area, where we mathematically model the vehicle's behaviour of accerleration and lane changing. By incorparating different factors of a vehicle, we can not only determine the shape and the size of our designed merging area, but estimate the accident rate inside this area.

Moving one layer up is our intermediate layer, where we abstracts the actual traffic situation into a set of connected merging points. This helps us to propose a tree-structured merging model, where a unique 2-2 merging pattern is proposed. Queueing theory is adopted to mathematically figure out the throughout rate of our solution. In order to simulate different scenarios in the real life, several factors are incorparated in our model, like the traffic situation(light/heavy), the proportion of different types of tollbooth in the toll plaza, and so on. We attempts to use our model to identify important factors related to the efficiency of our shceme.

The toppest layer mainly focus on the shape of the merging area, where we propose so called "Traffic Fan-in" model. Similar to the idea of water storage system, we expect our model to ease the traffic stress during extraordinary peak hour, by allowing vehicles to choose a more "free" lane.

Several numerical tests have been done to test the performance of our proposed models, and case study in New Jersey has been done to examine the feasibility of our scheme. We aim to control the total cost of our design.

Ultimately, we conclude that although our solution does improve the traffic situation to some extent, the giant turning point in accident rate and throughput may comes when more and more new technologies are adopted in the traffic system, like the traffic-improvement-aimed electronic intellectual devices.

Contents

1 Introduction

1.1 Context of the Problem

Tollbooths have become a common sight along the highway. The obvious advantage of this highway tolling system lies in the fact that it makes building long roads across cities, even countries, possible. Besides, it serves as a transportation demand management tool to control the traffic flow, unclog the crowed pathways as well as enhance the usage of transportation system[1]

The "barrier toll" on the highway is actually described as a row of tollbooths placed across the highway, perpendicular to the vehicle flow and there are usually more tollbooths than the lanes, which means there are two peculiar stages for the drivers to go through the tollbooth, which is conventionally called "fan in" and "fan out". The first stage describes that a driver drives through an area where the number of lanes increases(queueing area or fork area), while the second stage describes that a driver need to drive through a "squeezing area", or merging area, before returning the normal lanes(see figure 1).

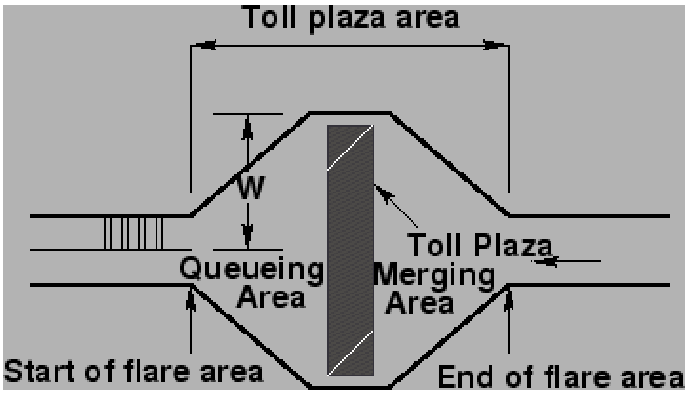


Figure 1: toll plaza

1.2 The Task at Hand

Inevitably, traffic congestion, or even accidents, may happen if too many cars try to "squeeze" back to the normal lanes, where the actual lanes are fewer than that in the merging area. Therefore, we develop a great interest in designing an efficient solution for this kind of merging problem. What mainly concerns us is how to design the size and the shape of the merging area, as well as a merging pattern, so that ur unique design not only reduce the accident rate in this area, but also ensure a large throughout rate. Meanwhile, we need to control the total cost of our design to make it financially enforceable.

1.3 Previous Work

There are several mathematical models of previous work, from which we get some idea for our models to some extent. In order to better understand this problem, we have studied those previous works thoroughly. For example, a mathematical model called microscopic traffic model, known to be the pioneer in the traffic simulation realm, has been studied for decades, and is relatively advanced. Besides, there was a lane change model developed in [3], incorporating explicit modelling of vehicle interactions using intelligent agent concepts. The lane change model was able to simulate highly congested flow condition in a realistic manner. Furthermore, we also studied a stochastic adaptive control model for traffic signal systems, based on statistical knowledge, the model contributed much to the design of our merging pattern.

2 Our Model

2.1 Objective

Aiming at reducing the accident rate and throughout rate in the merging area, we need to concentrate our attention in simulating the traffic flow in the merging area. The first thing we need to do is to model the geometry of the merging area, which naturally incorporates the shape and the size of the merging area as our considerations.

In order to study the accident rate in our model, we need to build a mathematical model to resemble the accident behaviour of the vehicles. For the sake of simplification, we assume that the accident only happens when vehicles merge into one lane. Therefore, our accident model needs to account for various factors behind the merging behaviour of vehicles, such as the types of the vehicles, since different cars have different car length and accerleration.

Furthermore,to study the throughout rate, we need to model the passing behaviour of the vehicles in the merging area, in which a merging scheme is hereby required. This model should be examined in different scenarios, for instance, under light and heavy traffic. Besides, other factors are worth our considerations, like the types of tollbooth, the types of vehicle in the merging area and so on.

Our final criterion for the model is that it is financially executable, which leads us to consider the costs of our merging plan.We wants to adopt our idea in actual situation, and discuss the feasibility of our model. Overall, we aims to provide a financially-friendly solution.

2.2 Problem Space

Honestly, the traffic pattern within toll plaza zone, in reality, is a black box and we hereby may not be able to clearly identify every aspects of it. With the aim to simplify the condition our model will work on, it is prudent to qualify a problem space simply by defining, in advance, three boundary assumptions.

Firstly, we have limited our geographically focus to multi-lane divided and limited-access highways toll plaza only in United States, one of the major country which was troubled with the highway traffic condition and also made great efforts to improve the situation. Whilst the predetermined boundary does not explicitly restrain capability of proposed model to deal with same issues in other countries, it allows us to take more specific factors into consideration, such as domestic traffic-related law and state-specific highway plight.

Secondly, we pay more attention to overall impacts on accidence prevention and turnpike throughput improvement over a long period of time (i.e. five to ten years). Actually, we notice that the failure of other models coping with the some issues roots in the fact that they, at least partially, attempted to elucidate and predict the behavior within a relatively narrow time window, (say, less than one year). However, traffic behavior can vary dramatically sometimes, behind which the real trend is hided. For instance, the list of causes, which can lead to incredibly obscured and extremely time-varying behaviors and undermine the model, include but are not limited to migration, facility construction, seasonal trend and variation of drivers awareness.

Lastly, beyond those two assumptions, we, in order to further boil down situation, we make tenable supposition about the types of tollbooths and vehicles. Our concentration are on three tollbooth types. And three lane types namely, cars, bus and trucks.

2.3 Behaviour-based and Tree-structured Multi-layer Framework

Given the broad scope of our objectives, we find it necessary to bring up a framework that beds multiple layers that respectively tackle the above objectives. The descriptions of the three layers are as follows (in figure 2).

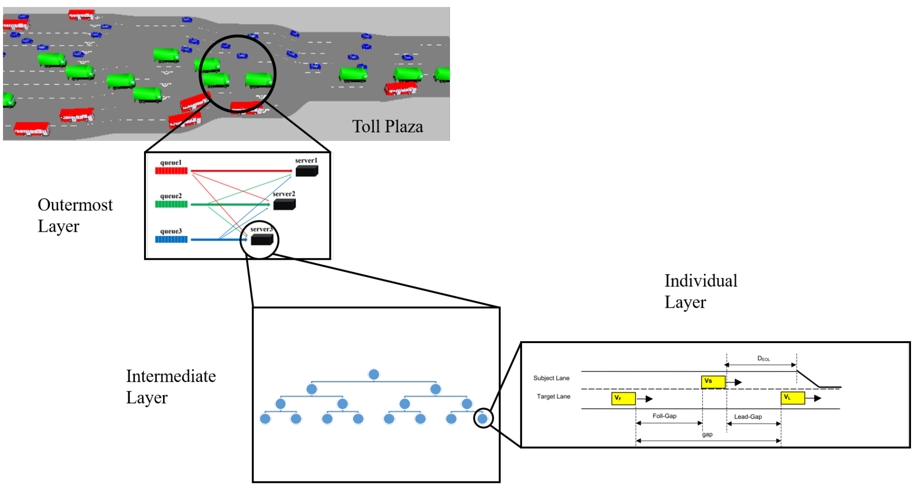


Figure 2: Behaviour-based and Tree-structured Multi-layer Framework

2.3.1 Individual layer - Behaviour-based Vehicle-Merging Model

To begin with, we model the car merging pattern from the basic layer, which is actually in an individual base. Our model involves two general part, which is Driver Merging Decision Model (DMDM) and the Subject-Car After Merging Adjustment Model (AC-MAM).

The DMDM is well presented in the decision tree model below(in figure 3). From the perspective of a driver, he will subconsciously check, before deciding to change the lane, whether desired space, which is determined by linear formula developed by Hibas[4], is satisfied to merge into the major line. Otherwise, he will slow down the speed and wait for another proper merging opportunity. After merging into the major lane, the diver will adjust vehicles speed and position accordingly so that eventually, the vehicle can maintain safety (minimum) distance and move as fast as the leading vehicle. For instance, consider a scenario that, after cutting into the major lane, the subject car falls into the leading cars dangerous region and then it will first slow down the speed and remove itself far from the dangerous region. Next, he will accelerate and make its speed as same as the followers to prevent accidence between it and the following car.

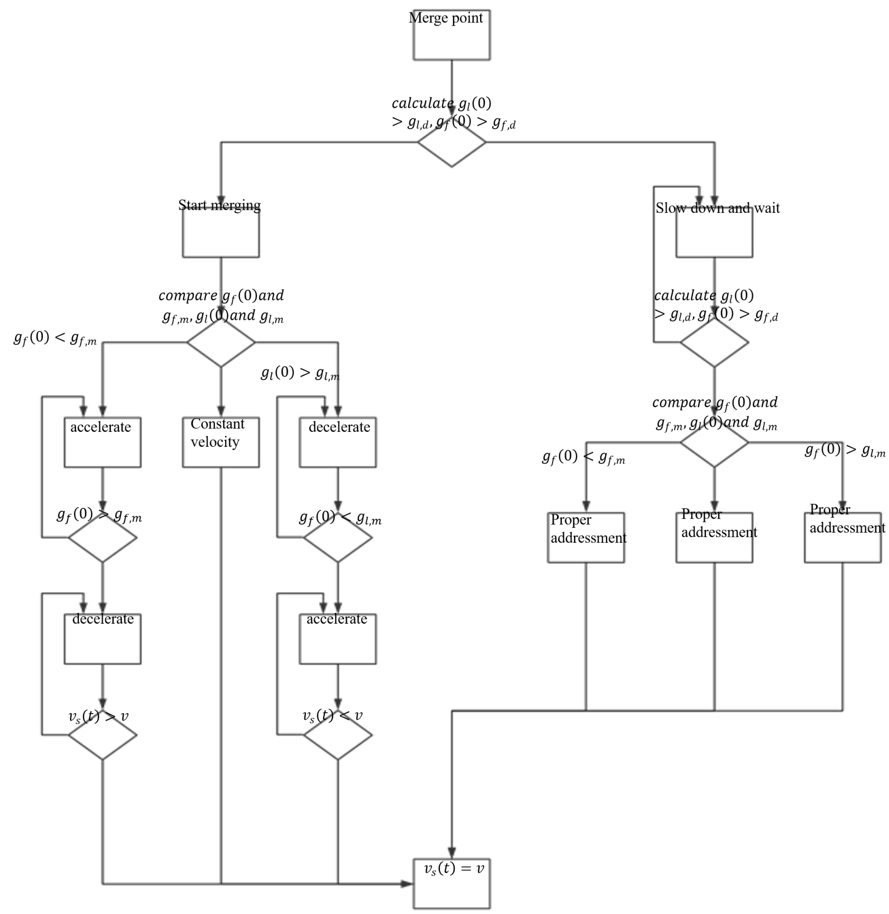


Figure 3: decision flow

As for the ACMAM, we mainly focus on the vehicles dynamics after merging. In order to simplify the calculation, we use a simple trick, taking the following car as reference coordinate (i.e. treat it as still). One can show that if the subject car merge into major line with lower speed and within the dangerous region of following car, it will take the longest time to come to desired state (i.e. keep the speed as same as the system). In other words, this case is the most dangerous one and indeed accident normally occur in such scenario. In other to prevent accident happen in these scenario, we need to make the two merge point long enough so that before subject car reach next merging point, it

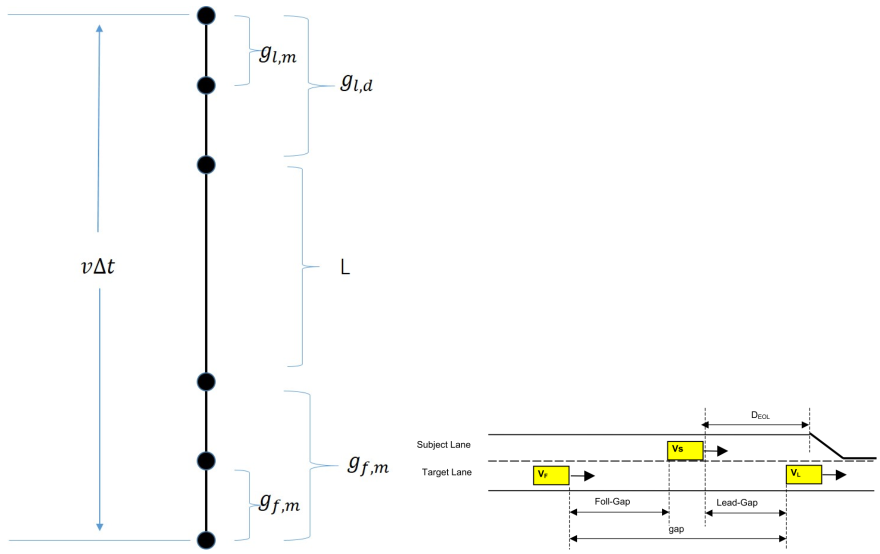


Figure 4: sketch

has already reach desired status. In figure, the dynamic system satisfies
We adopts the following symbols as our constants and variables in the model:

Table 1: Constants table

Constants	Meaning	Units
l	the distance between the leader car (tail) and the following car (head)	m
g_{ld}^1	the maximun distance between the head of the subject car 1 and the tail of the leader car within which the subject car 1 won't change its lane	m
g_{fd}^1	the maximun distance between the tail of the subject car 1 and the head of the follower car within which the subject car 1 won't change its lane	m
g_{ld}^2	the maximun distance between the head of the subject car 2 and the tail of the leader car within which the subject car 2 won't change its lane	m
g_{fd}^2	the maximun distance between the tail of the subject car 2 and the head of the follower car within which the subject car 2 won't change its lane	m

$$\begin{cases} \frac{1}{2}a^{+}t_1^2 - \frac{1}{2}a^{-}t_2^2 = d \\ -v_0 + a_{+}t_1 - a_2t_2 = 0 \end{cases} \tag{1}$$

Table 2: Variables table

Symbols	definition	
v_0	the initial (relative) speed of the subject car	(m/s)
x_0	the initial head position of the the subject car	m
a_+	the deceleration rate of the subject car	(m/s^2)
a_-	the deceleration rate of the subject car	(m/s^2)
t_1	the acceleration time	m
t_2	the deceleration time	m
d	relative distance between subject car and the follower car	m
c	ration equals $\frac{a_+}{a_-}$	

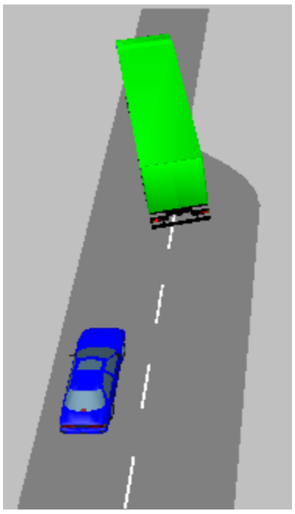
We solve the system and get the following solution:

$$D_s(v_0, d, a_+, a_-) = d + vt(v_0, d, a_+, a_-) \tag{2}$$

Where, $t(v_0, d, a_+, a_-) = t_1 + t_2 = \frac{1}{c}(\frac{v_0}{a_2} + \frac{\sqrt{2a_- cd(1-c)+v_0^2 c} - v_0}{a_2(1-c)}) + (\frac{\sqrt{2a_- cd(1-c)+v_0^2 c} - v_0}{a_2(1-c)})$

Acute readers may find that the figure shows the details of the ACMAM illustrated above.

2.3.2 Intermediate layer - Stochastic-Tree-structured Traffic-Merging Model



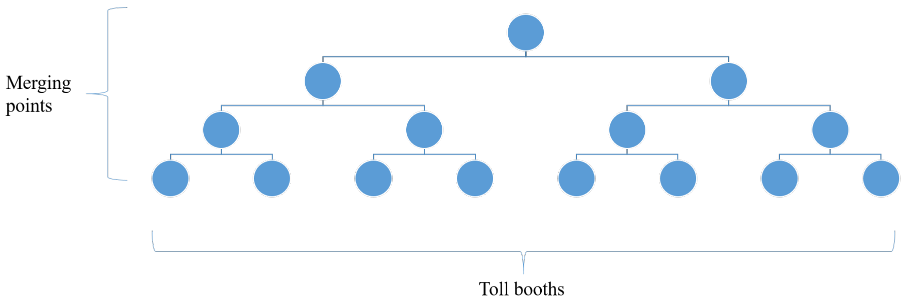


Figure 5: Merging point visulization

Moving one layer up from the merging model is the traffic flow model, where we consider the whole picture of the traffic flow in the merging area. To simplify the problem, we need to specify some assumptions on our model:

- The vehicle can be roughly classified into three types by its length: car, truck and super truck.
- The traffic flow is constant in a short period of time.
- Before arriving the tollbooth, all the vehicles have already been classified according to their types, which means one tollbooth only serve one type of vehicle.
- All the tollbooths that serve the same type of vehicle are connected.
- The drivers can only be delayed during the process of merging.

We define the merging point as a part where lanes join together. In our model, we only consider the kind of merging point where two lanes merge into one lane(see the left figure in figure 5). For further simplification, we use circles to represent merging points, and straight lines to represent roads. Thus, our model has kind of "tree" structure, with a new merging pattern: merging points are 2-2 connected. (see the right figure in figure 5)

With the above effort, we are given with a "forest" like system of circles and straight lines for each type of vehicle. We can view our model as a combination of three queueing system. We then examine different components in our model.

Merging point Merging points, as building blocks of our model, can be viewed as a server with certain arrival rate and service rate. For simplification, we assume the arrival pattern is exponential. There are two cases we need to consider in the merging process:

- Only one car appears in the merging point
- More than one car appear in the merging point.

Obviously, the merging point has different service rates in these two cases. We use μ_0 and μ_b to denote them respectively. Meanwhile, λ is used to denote the arrival rate of the merging point. For better visualization, see figure 6.

Let P_n be the probability that there are n drivers on average in the merging point (steady state). From the Queueing theory, we have

$$\begin{cases} \lambda P_0 = \mu_0 P_1 \\ \lambda P_1 + \mu_0 P_1 = \lambda P_0 + \mu_b P_2 \\ \lambda P_n + \mu_b P_n = \lambda P_{n-1} + \mu_b P_{n+1}, n \geq 2 \\ \sum_{i=0}^{\infty} P_i = 1 \end{cases} \tag{3}$$

The solution of P_n has the following forms:

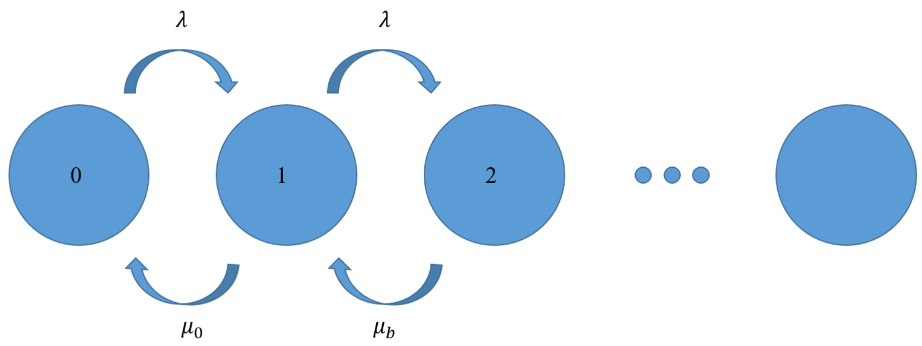


Figure 6: merging point as a queueing system

$$\begin{cases} P_0 = \frac{1}{1+\frac{\lambda}{\mu_0}+\frac{2\lambda^2\mu_b}{\mu_0(\mu_0+\lambda)(\mu_b-\lambda)}} \\ P_1 = \frac{\lambda}{\mu_0}P_0 \\ P_n = \frac{2\lambda^2}{\mu_0(\mu_0+\lambda)}(\frac{\lambda}{\mu_b})^{n-2}, n \geq 2 \end{cases} \tag{4}$$

So, the expected number of drivers in the merging point is :

$$N(\lambda) = \sum_{i=0}^{\infty} iP_i = \frac{\lambda}{\mu_b - \lambda} + \frac{\mu_b - \mu_0}{\lambda(\mu_b - \mu_0) + \mu_0\mu_b}$$

The average waiting time in the merging point can be calculated by Little's Theorem [10]

$$t_{wait}(\lambda) = \frac{N(\lambda)}{\lambda} = \frac{1}{\mu_b - \lambda} + \frac{\mu_b - \mu_0}{\lambda(\mu_b - mu_0) + \mu_0\mu_b}.$$

Furthermore, form [7], we can calculte another important indicator of our model: the average wasted time of a driver in a merging point:

$$t_{waste}(\lambda) = t_{wait}(\lambda) - \frac{1}{\mu_0} = \frac{1}{\mu_b - \lambda} + \frac{\mu_b - \mu_0}{\lambda(\mu_b - mu_0) + \mu_0\mu_b} - \frac{1}{\mu_0}$$

One queueing system A queueing system for one type of vehicle is actually a combination of tollbooths, merging points, and normal leaving lanes that flee away the toll plaza. We are given some parameters of this system:

We introduce the notion of *layer* to characterize the merging points: a merging point is in layer $k(k < M)$ if it accepts the total traffic flow of 2^k tollbooths. Then the arrival rate for a merging point at k_{th} layer should be:

$$\lambda_0 = \frac{2^k\phi}{T}$$

Furthermore, given the real life experience, we incorporate the factor of the tollbooth type in our model by introducing the parameters α, β and γ, which represents the efficiency of three types of tollbooths(conventional, automated and electronic). We assume that the parameters uniformly affects the overall arrival rate of all the merging points

Table 3: Parameters for the queueing system

Constants	Meaning
ϕ	traffic flow
T	number of tollbooths in the system
N	number of normal lanes that flee away the toll plaza
M	number of merging point in the system
A	the percentage of conventional tollbooths
B	the percentage of automated tollbooths
C	the percentage of electronic tollbooths
α	a coefficient related to conventional tollbooths
β	a coefficient related to automated tollbooths
γ	a coefficient related to electronic tollbooths

inside the system depending on the quantitative relation between A, B and C. This is reasonable because the electronic tollbooth is more efficient than the other two types of tollbooth, and the automated tollbooth is more efficient than the conventional tollbooth. So we amend our initial arrival rate in this way:

$$\lambda_k = \begin{cases} \lambda_0 * \alpha & A = max(A,B,C), A \neq B, A \neq C \\ \lambda_0 * \beta & B = max(A,B,C), A \neq B, B \neq C \\ \lambda_0 * \gamma & C = max(A,B,C), C \neq B, A \neq C \\ \lambda_0 * \frac{\alpha+\beta}{2} & A = B = max(A,B,C), A \neq C \\ \lambda_0 * \frac{\gamma+\beta}{2} & C = B = max(A,B,C), A \neq C \\ \lambda_0 * \frac{\alpha+\gamma}{2} & A = C = max(A,B,C), A \neq B \end{cases}$$

The probability that a driver reach a merging point are as follows:

Table 4: Probability

Index of the layer	Probability
1_{st}	$\frac{2^1\phi}{T}$
2_{nd}	$\frac{2^2\phi}{T}$
3_{rd}	$\frac{2^3\phi}{T}$
...	...

From the previous part, we have the average wasted time for a driver in a merging point. Thus, the total average wasted time for a driver inside this queueing system should equal *the weighted sum of all the wasted time spent in different layers of merging point*, where the corresponding rate is the probability for a driver to reach that merging point.

Therefore, the overall wasted time for a driver in the queueing system is:

$$t_{total} = \sum_{k=1}^{\log(\frac{T}{N})} \frac{2^k \phi}{T}(t_{waste}(\lambda_k))$$

where λ_k is chosen by our above scheme.
Moreover, we can calculate the expected number drivers that left the system by the following formula:

$$N_{total} = \phi - \sum_{k=1}^{\log(\frac{T}{N})} \frac{T}{2^k} N(\lambda_k)$$

Total merging process The total merging process is just a summation of the three queueing system.

2.3.3 Outermost layer- Diverted Traffic Fan-in Model

Moving one more layer up is our overall toll plaza specific traffic fan-in model. Accurately simulating traffic fan-in model within the region we tested is central to the purpose and design of our overall toll plaza model. And we notice that no one have invested interest in the improvement of toll plaza shape and size. Inspired by the capillary system and multi-layer water-conservancy storage system , we integrate the idea from these fields into our model and further develop our own multistep traffic fan-in model. Based on the data from Department of Transportation in United States, we see a traffic pattern, of which, during peak hour, the traffic delay in car lanes are extremely severe. Besides, the truck traffic volume, sometimes, may also have spike on a weekly basis. On the country, the combined truck traffic is always insignificant. And then the fan-in vehicles, via our designed traffic model, will thereafter move through the merging area stated above.

Specifically, our fan-in model, separate the originally one-step traffic fan-in into three discrete step (as is shown in figure). To begin with, car will enter the toll plaza in station, trucks will enter the B station and long truck will enter the station C. Our models uniqueness rely on the special design so that, under some certain situation(usually when the traffic is extremely heavy), cars can move to station BC. Even tracks which should enter station C can also enter station C in this scenario. As is clearly shown in the graph, the model we designed implies that our proposed toll plaza will feature a Z shape, in which different types of vehicles will pass through accordingly, and is still with controlled size, which takes both space saving and cost restriction into consideration.

Like a water storing system, this kind of multistep design is expected to ease the traffic stress during extraordinary peak hour. Consider, for instance, when the traffic in A exceed certain limit, our model can reallocate the incoming motors into the other two lane dynamically. The merits of this kind of traffic diverting is that, on the one hand, obviously when the traffic load is too heavy, the system can efficiently reallocate the traffic evenly and on the other hand, each car can pass through the toll plaza orderly, even under scenario like extreme high traffic volume. In addition to that, it is also worthwhile to mention that, under normal traffic rate, it is safe to draw the conclusion that, by separating different cars into different paths, the procedure of toll fee collecting and vehicles passing through can be faster and the divers can drive more easily, the efficacy of the many vehicles entering fork zone will increase while the accident frequency in this area will decrease.

Further in the section below, we will demonstrate a computer simulation using Python, making a performance comparison between the model we proposed (as shown in figure) and the normal case which most toll plaza in United States implemented.

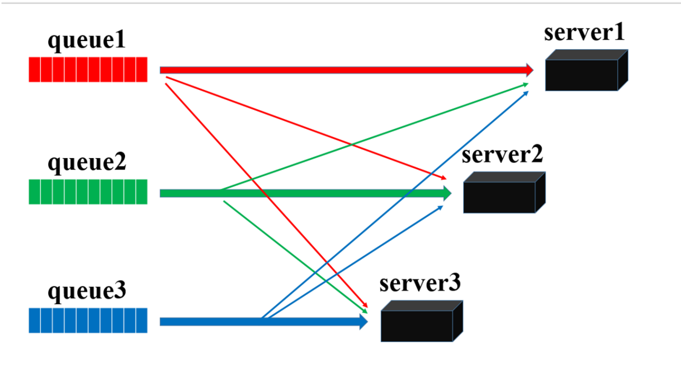


Figure 7: Diverted Traffic Fan-in Model

2.4 Additional consideration - Merging-Accident-Simulation Model

At the most basic level, the simulation models vehicles as building blocks, whose length, speed, acceleration, and the position on the lane, can be quantified and calculated. Within this layer, we mainly want to study the merging behaviour of the vehicles, which helps us to approximate the accident rate.

For the simplicity of our model, we only consider two scenarios of merging. One is the case when two lanes merge into one, the other is the case when three lanes merge into one. In the first case, only one car is going to change its lane, while in the second case, two cars on different lanes mat change to the same lane at the same time.

In our model, a vehicle is modeled as a rectangle, which is going to change its lane .We make several assumptions of our model.

- all the cars move at a constant speed initially, and moving in the same direction.(right by default)
- The rightmost part of the rectangle is defined as *head*, while the leftmost part is defined as *tail*
- The vehicles on lane B are seperated at a constant distance.
- The merging process is finished instantaneously.
- The driver in the subject car won't change his lane if he is too close to the learder car or the follower car.

Under the above assumptions, we can further assume that all the vehicles on lane B are still, and the subject car has a relative speed compare to this still system. Since the merging process is done instantaneously, the (relative) speed of the subject car won't be affected during the merging process. Furthermore, we can introduce the concept pf *positive real line* to number the position of each car on the lane. Without loss of generosity, we can assume the head position of the follower car is 0, while the tail position of the leader car is some positive constant.

The following tables are the constant and variables used in the model (apply to both scenarios).

Table 5: Constants for two scenarios

Constants	Meaning	Units
l	the distance between the leader car (tail) and the following car (head)	m
g_{ld}^1	the maximun distance between the head of the subject car 1 and the tail of the leader car within which the subject car 1 won't change its lane	m
g_{fd}^1	the maximun distance between the tail of the subject car 1 and the head of the follower car within which the subject car 1 won't change its lane	m
g_{ld}^2	the maximun distance between the head of the subject car 2 and the tail of the leader car within which the subject car 2 won't change its lane	m
g_{fd}^2	the maximun distance between the tail of the subject car 2 and the head of the follower car within which the subject car 2 won't change its lane	m

Table 6: Variable of the subject car(1 and 2)

Symbols	definition	
v_0^1	the initial speed of the subject car 1	(m/s)
v_0^2	the initial speed of the subject car 2	(m/s)
x_0^1	the initial head position of the subject car1	m
x_0^2	the initial head position of the subject car2	m
a_-^1	the deceleration rate of subject car1	(m/s^2)
a_-^2	the deceleration rate of subject car2	(m/s^2)
$len1$	the length of the subject car1	m
$len2$	the length of the subject car2	m
y	the reaction time of the driver of the subject car	m

First scenario The first scenario describes that only one car(subject car)changes its lane.

There are two types of car-crashing in this scenario: the subject car hits the leader car, or the subject car hits the follower car. In the first case, the direction of speed of the subject car is rightward, while it is leftward in the second case. In both cases, we can define *braking distance* as

$$D_{brake} = \frac{(v_0^1)^2}{2a_-^1} + |v_0^1|y$$

Two cars crash only when one of the following two situation happen:

- $v_0 > 0$ and $x_0 + D_{brake} \geq l$
- $v_0 < 0$ and $x_0 - len - D_{brake} \leq 0$

Second scenario The second scenario is different from the first one in that it considers the case when two cars on different lanes change into the same lane at the same time. Without loss of generosity, we assume that after merging, subject car 1 is placed on the left hand side of the subject car 2. More types of car crashing need to be considered: subject car(1 or 2) hit the leader car, subject car(1 or 2) hit the follower car, subject car 1 hit subject car 2. The first two types can be characterized using the criterion in the first scenario, while the third type can be characterized by the following inequalities:

- subject car 1 and subject car 2 move rightwards(relative to the still system), and $|v_0^1| > |v_0^2|$
  $x_0^1 + \frac{(v_0^1)^2}{2a_-^1} + v_0^1 y \geq x_2 - len2$
- subject car 1 and subject car 2 move leftwards(relative to the still system), and $|v_0^2| > |v_0^1|$
  $x_0^2 - len2 - \frac{(v_0^2)^2}{2a_-^2} - v_0^2 y \leq x_1$
- subject car 1 move rightwards, and subject car 2 move leftwards(relative to the still system)
  $x_0^1 + \frac{(v_0^1)^2}{2a_-^1} + v_0^1 y \geq x_2 - len2 - \frac{(v_0^2)^2}{2a_-^2} - v_0^2 y$

2.5 Implementation

Vissim, developed by PTY Group, is a microscopic, time-step and behavior-based simulation model developed to analyze a full range of traffic operations on virtually any kind of roadway. Vissim provides massive 3D models and network objects to help people build the emulated structure. The software offers flexibility in several respects: the concept of links and connectors allows users to model geometries with any level of complexity. Attributes for driver and vehicle characteristics enable individual parameterisation. Furthermore, a large number of interfaces provide seamless integration with other systems for signal controllers, traffic management or emissions models. The most attractive property of Vissim is its sketching and simulating function, which is quite advanced and powerful. We used Vissim with the support of Google Earth to locate the real case on New Jersey highway and build the 3d presentation of our mathematical model.

3 Sensitivity Analysis and Results

3.1 Factors to the distance of two merging points

We adopts two parameters in the program to mathematically calculte the distance between two merging point: the initial speed v_0 of the subject car and the initial merging position d. The result is as follows. From the graph we can see that under the same initial

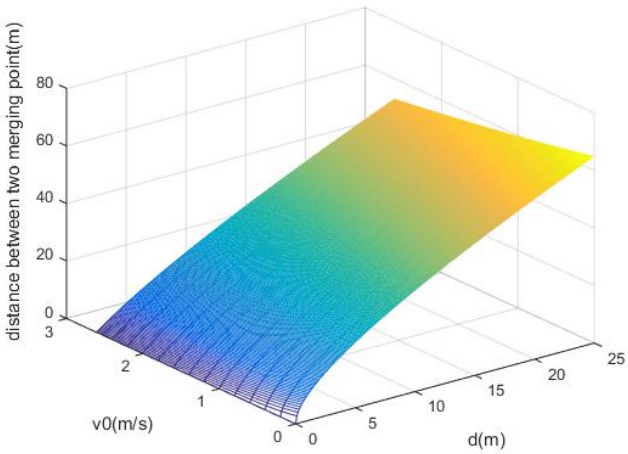


Figure 8: Test of merging point

speed, the distance increases linearly with the initial position of the merging point. The factor of speed has no influence on the distance between merging points.

3.2 Factors to Merging-Accident-Simulation Model

We test the accident rate during the merging process in this model. From [], we can roughly estimate the value of some parameters, which we are going to adopt in our numerical test.

3.2.1 The length of the car

The length of the subject car should be a factor affecting the accident rate. By common sense, we all know that a long car can be very clumsy in changing its lane, which may trigger a crashing with other incoming car. Therefore, we use MATLAB to simulate the merging process of different types of vehicle in the first scenario to see the changes in the accident rate.The result is listed in the following table.

The above result show that the length of the vehicle does affect the accident rate when merging. The longer of the car, the higher the accident rate is.

Table 7: Constant values used in testing

symbol	constant name	value
d_{car}	length of a car	$5m$
d_{truck}	length of a truck	$11m$
d_{Super}	length of a super truck	$22m$
d_{Super}	length of a super truck	$22m$
a^+_{car}	acceleration rate of a car	$3m/s^2$
a^+_{truck}	acceleration rate of a car	$1.8m/s^2$
a^+_{super}	acceleration rate of a super truck	$1.5m/s^2$
a^-_{car}	deceleration rate of a car	$3m/s^2$
a^-_{truck}	deceleration rate of a truck	$1.5m/s^2$
a^-_{super}	deceleration rate of a super truck	$1.3m/s^2$

Table 8: Simulated accident rate

number of experiment	1	2	3	4	5
car	2%	4%	0	0	0
truck	1%	6%	6%	6%	8%
supper truck	24%	26%	18%	16%	18%

3.2.2 Number of vehicle that merge at the same time

Besides the length of the vehicle,another factor that concerns us is the number of vehicle that change into the same lane in the same merging point at the same time. We did several numerical experiments to test the accident rate in three combinations: car-car, truck-truck and superTruck-superTruck.

Table 9: Simulated accident rate

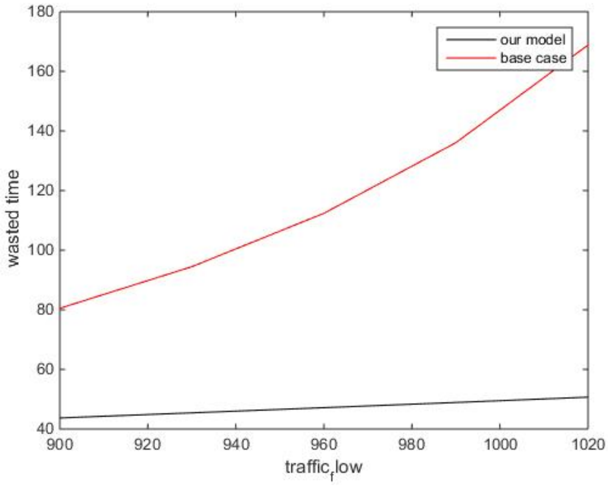
number of experiment	1	2	3	4	5
car-car	22%	24%	20%	18%	18%
truck-truck	30%	22%	26%	32%	34%
sTruck-sTruck	62%	56%	48%	42%	38%

The result show that the accident rate greatly improves. By the above two simulations, we numerically show that accident rate greatly improves when two or more car try to squeeze in the same merging point, which reflects the advantage of our designed merging pattern, which only allows one car to change its lane in the merging point.

3.3 Factors to Stochastic-Tree-structured Traffic-Merging Model

3.3.1 Total traffic flow

The volume of traffic flow may affect the throughout rate of our model. If the traffic flow is large, we expect that there are more vehicle squeezing in the merging area, which may increase the total wasted time of a driver and the throughout rate of our model. Thus, we do several numerical test to simulate the situation in light and heavy traffic. From [], we estimate the initial traffic flow is 900 $vehicle/hr$, then we increase this value by 30 $vehicle/hr$ to see the performance of our model.



3.3.2 Percentage of different tollbooth

We want to examine the performance of our model under different proportion of the three kinds of tollbooths. We expect that given the same traffic flow, the better our model performs if we increase the proportion the automated and electronic toll booth while keeping that of conventional tollbooth low.

For the sake of convenience, we denoted the conventional, automated and electronic toll booth by C , A and E respectively. We keep the traffic flow as 900 $vehicle/hr$ in our simulation test.

Increasing proportion of E The above figure indicates that increasing proportion of E shortens the wasted time of a driver, which coincides the high efficiency of the electronic tollbooth.

Increasing proportion of A The above figure also indicates a negative relation between the proportion of A and the wasted time, but it does not perform well than the first case.

Increasing proportion of C The above figure show that the low efficiency of the conventional tollbooth increase the total wasted time.

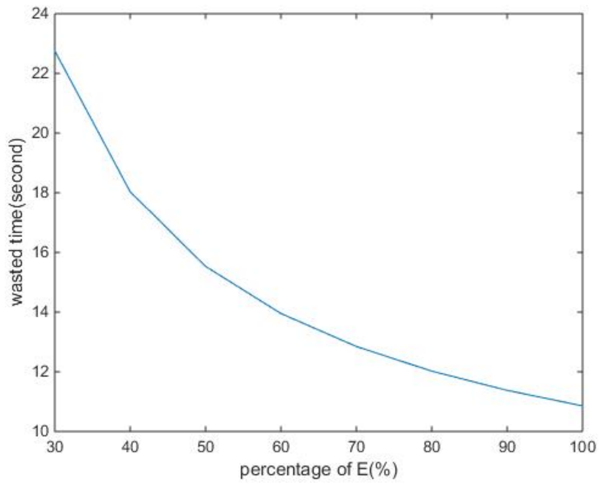


Figure 9: Simulation test result

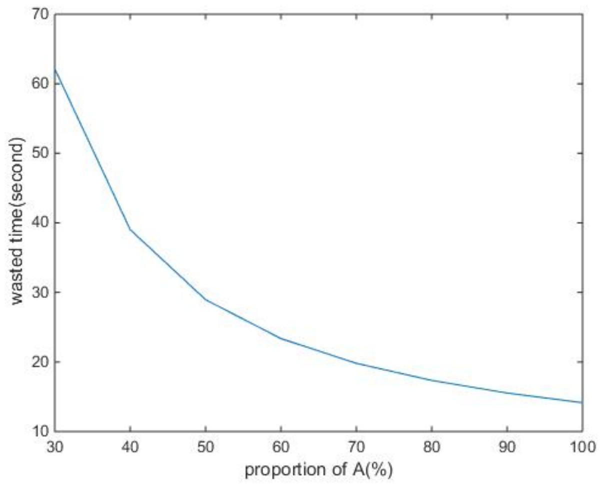


Figure 10: Simulation test result

3.3.3 Percentage of different types of vehicle

As far as the composition of the vehicles, different parameters are also used in our Tree-structured Merging model to represent the portion of different vehicles. This percentage parameters are real values between 0 and 1, by controlling which can we find the relation between the hour volume and the traffic composition.

3.4 Case Study

Our team base the study on the Gateway State Parkway, and we chose Asbury Toll Plaza for our case study. According to the statistics we got from the historical documents, there are 9 lanes pass through the Asbury Toll Plaza, and the average speed of the cars is 55 65 miles per hour. Average toll fee for car is 2 cents/hour for cars and 6 cents/hour for trunks, however, the cars toll transaction makes up 96% of the total revenue of Asbury

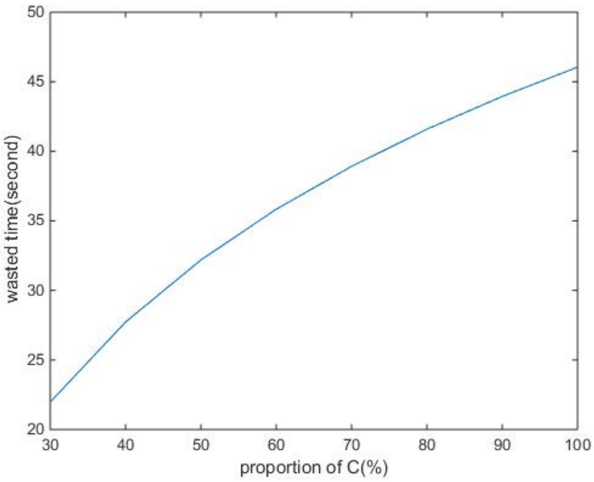


Figure 11: Simulation test result

Toll Plaza. Besides, the hour volume of Asbury Toll Plaza can be up to 8000 9000. According the above data and our calculation, we can reasonably get the following parameters:

- Lanes: 9
- Velocity of vehicles: 96.6 km/h
- Velocity when the vehicles are in the toll plaza: 5 km/h
- Composition of vehicles: 80% cars, 10% buses, 10% HGV
- Vehicles enter the Gateway State Parkway per hour: 8500

Tree-like merging pattern In the above simulation, we test our model in a growing traffic-flow situation, where x-axis stands for the value of the traffic flow $vehicle/hr$, and the y-axis stands for the expected wasted time of a driver. The red line is the base case, while the black one is the result of our model. From the figure, we can see that our model make a success to some extent in withstanding the pressure of on-growing traffic flow.

Table 10: Simulated accident rate when one car merge

number of experiment	1	2	3	4	5
	0%	2%	4%	2%	0%

Accident simulation The above results show that in the real life, the accident rate in one-car merging case is lower than that in the two-car merging case, which supports the feasibility of our model.

Figure 12: Asbury Toll Plaza

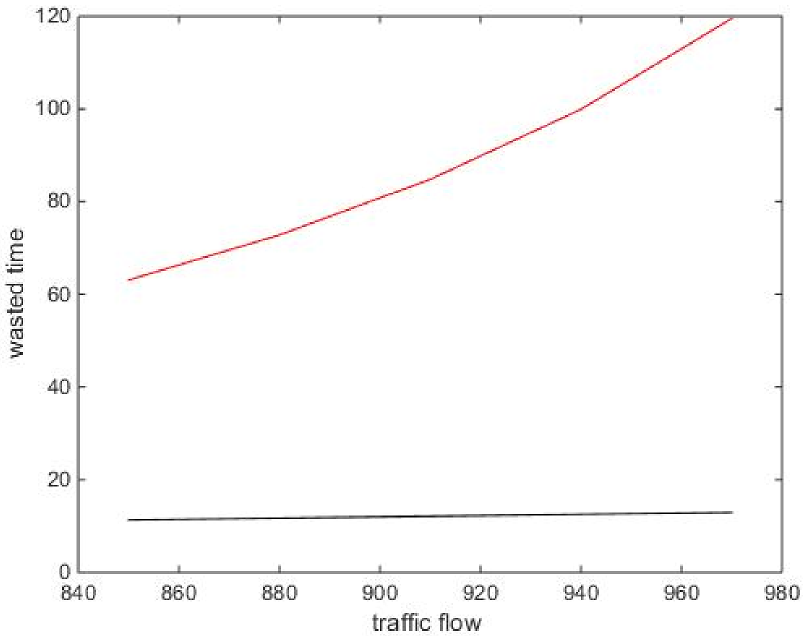


Figure 13: Simulation test result

Table 11: Simulated accident rate when two car merge

number of experiment	1	2	3	4	5
	28%	24%	20%	14%	24%

Cost Analysis It is obviously that our design will lead to more cost than traditional pattern. Our tree-like model can accommodate more toll booth to work together, and in order to limit the width of the parkway, we increased the length of the toll plaza. However, adopting a layer by layer model, we design our plaza by decreasing the width gradually. As a result, if we just consider the cost on road construction and the building cost of toll booth we can use a roughly triangle model to estimate the total cost of changing the previous pattern. According to our estimation, the new pattern will cost 20% more than traditional design.

Multi-step model Factors: Annual Average Weekday Traffic (ADDT): the amount of cars entering the Gateway State Parkway, corresponding to the factor of our model. Be-

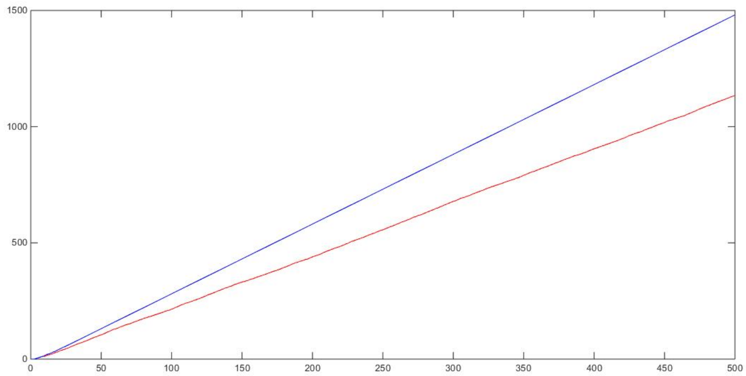


Figure 14: Simulation test result

cause different kind of vehicles have different traffic flow, we used 3 different factors to test our model. Result: the red line represents the theoretical prediction on the number of cars stuck in the congestion by the timeline with our model, while the blue one represent that which does not adopted our model:

4 Conclusions

4.1 Future traffic trend

During the theoretical computation, we discern that, assumed the road condition and toll booth service doesnt change, the critical indicator like waiting time, accident rate and throughput will excoriate inevitably, with the increasing worsen rate. The same worsening traffic also appears in our computer reality based analysis. Unfortunately, according to US government statistics, the truck and private car ownership increase rapidly in a yearly basis, and the condition is roughly the same around the globe. Consequently, unless the governments take more actions to encourage like public transportation and rail-based cargo transportation, or any other significant methods to slow down the overall traffic increase rate, the excoriation of road traffic condition cannot be solve only with effort to improve the road capacity or upgrade the traffic facility.

With the current trend, the traffic condition like the traffic accidents rate and traffic overall throughput will inevitable become worse and worse. Sadly, the reality reflect the same as our model predict. According to the statistic, in spite of the advance in traffic instruments, traffic-improvement-aimed electronic intellectual devices and road design, the traffic accident and conjunction rates decently.

Due to our models initial objectives that it is designed to improve the traffic condition in the long run, this kind of realistic traffic condition also influence our model design. In our previous work, we have already discussed the scenario like, extraordinary high traffic volume, and the great increase in traffic passing through in a relatively short period of time. And according to the same simulation experiment results, our solution can slightly enhance the performance of toll plaza throughput under scenario like extremely high traffic volume.

4.2 Summary and reflections

In our research, we have developed one model for drivers behavior analysis, one new model for toll collection and one model for traffic flow merging. The driving behavior model we have built adopted some basic physical assumption and kinematic knowledge, it can fit the real condition well, which is just the basic model we used to design and evaluate other new models in our research. The tree-like merging pattern we designed has been proved to be valid and effective to relieve the traffic congestion. One merit of our tree-like merging pattern is that it is based on stochastic process and probabilistic background. The stochastic is powerful under the condition that the real situation is too sophisticated to describe every feature of the network and transaction in detail, thus it is better to use probabilistic tool to simplify the problem. As far as the multi-layer toll design, it is a powerful model to increase the speed of toll transaction so as to aggrandize the hour volume of the toll plaza. Even though the new design will add more burden to the government revenue, the advantage it brings, such as the larger hour volume and faster toll transaction speed, still encourage the government to adopt our model.

However, there is also some limitation in our mathematical model. It is difficult for us to analyze every detail of this problem is five days, thus to some extent we have simplified the problem. The process of simplification requires some strong and perfect assumption on physical background and free of effect of other relevant elements, which means that our model may not perfect under the real condition. Despite the imperfect, our model can fit the real life situation well enough.

4.3 Future work

Given 96 hours in total, it is difficult for us to take every aspects of the problems into consideration. Some factors, due to the sophisticated nature of themselves, are left out.

4.3.1 Vibration with time line

Our model take factors like the peak and normal traffic into our main consideration. Due to the previous simulation fact that the traffic volume variance will have a significant influence on the model final performance. At the same time, we consider the different traffic composition in terms of traffic type. For instance, in our research, we found that in sample one, while in sample two. This kind of model characteristic makes our model more unstable and unpredictable. In addition to that, we fail to consider the seasonal or periodically factors may also have potential influence on model output. For example, the seasonal traffic pattern will influence the outside condition that our model will then apply. And another good example is that the seasonal whether dynamics may alter the traffic network status. On the one hand, factors above like these may have unpredictable and ambiguous direct or indirect impact on the toll plaza, undermining the reliability of our model. One the other hand, they are so sophisticated, insane and obscured that the scholar still cannot reach a consensus on these issues. It will become a rather challenging task to integrate those periodical and seasonal factors into the models.

4.3.2 Local factors(Geographical, law, demographical)

There comes a time in every model life when it must put away simple toy models and attempt to tackle the things it was built to deal with. Our multilayer model has a unique feature that it consider the individual feature and overall features separately. Hence, the individual characteristic such as the drivers behaving pattern, the general vehicle length in local traffic, the ADDT of each direction of one highway; the local legal constrain on speed, lane number and toll plaza; the geographical condition and the soil construction condition in those area, can be integrated into our model in different layer. However, on the one hand, those factors are very locally specific and thus we need to analyst them case by case, which make it impossible and infeasible to come up with a general theory to model the pattern. On the other hand, in the real practice, the policy maker have to consider those local factors and apply the theoretical model accordingly. In addition to that, this is also related to the study field of geography, demography and even weather dynamics, and therefore many experts from different study fields will be needed to collaborate in the model building and advising process.

4.3.3 Details of toll booths

According to the survey from, we only take three kinds of toll booth into consideration (i.e. ETC, coins and cash one). At the same time, thanks to s previous effort, who draws the conclusion that the best tooth booth to lanes ratio is roughly two, we barely assume that the toll booth in model is as twice as the lanes. However, in order to improve the throughput and reduce the traffic accident rate, one can also achieve the same goal by designing more reasonable safe and toll plaza. For instance, the optimal lanes to toll booths ratio under various conditions, how the change in toll charging system or methods will improve the traffic efficiency, whether the engineering parameters like the length and width of single booth will impact the safety performance or construction cost. Considering that, unfortunately, the contest time is so limit and the room of discussion is strictly restricted to 20 pages, we have to concentrate on important factors and leave out the less important factor like this, we make the finest and most realistic assumption, although we dont expand a lot in these issues.

References

[1][2]Cherng, C., Lewis, A., & Pai, S. (2005). Modeling toll plaza behavior using queuing theory. *Mathematical Contest in Modelling, University of Washington, Department of Mathematics, Washington, USA.*, 19. Retrieved from http://www.math.washington.edu/~morrow/mcm/cary05.pdf

[3][4]Hidas, P. (2005). Modelling vehicle interactions in microscopic simulation of merging and weaving. *Transportation Research Part C: Emerging Technologies*, *13*(1), 37–62. https://doi.org/10.1016/j.trc.2004.12.003

[5]State, G., Asset, P., & Report, F. (2008). NEW JERSEY TRAFFIC AND, *44*(January).

Ahmed, K. I. (1999). Modeling Drivers â€™ Acceleration and Lane Changing Behavior. *Transportation*, *Ph.D*, 189. Retrieved from http://web.mit.edu/its/papers/DRIVIN.PDF

Analysis, B.-T. P. (n.d.). Appendix B : Transportation.

Caprani, C., OBrien, E., & Lipari, A. (2011). Extension of a Lane-Changing Model To a Micro-Simulation Tool. *Proceedings of the ITRN2011*.

Corwin, I., Rozenblyum, N., & Corwin, I. (n.d.). A Single-Car Interaction Model of Traffic for a Highway Toll Plaza A Single-Car Interaction Model of Traffic for a Highway Toll Plaza, *26*(3), 299–315.

Datta, T., Hartner, C., & Grillo, L. (2007). Merge System At Freeway Construction Work Zones. *Work*.

Doboszczak, S., & Forstall, V. (2013). Mathematical modeling by differential equations, (October).

Evans, J., & Evans, J. (n.d.). A Quasi-Sequential Cellular-Automaton Approach to Traffic Modeling A Quasi-Sequential Cellular-Automaton Approach to Traffic Modeling, *26*(3), 331–344.

Fang, E. X., Liu, H., Toh, K.-C., & Zhou, W.-X. (2016). Max-Norm Optimization for Robust Matrix Recovery. Retrieved from http://arxiv.org/abs/1609.07664

Federal Highway Administration. (2013). FHWA-PL-13-015 - Traffic Monitoring Guide. *Fhwa*, 462. Retrieved from http://www.fhwa.dot.gov/policyinformation/tmguide/

FHWA ’ s 13 Vehicle Category Classification. (2013), (Tmg), 2013.

Hidas, P. (2006). Evaluation and further development of car following models in microscopic traffic simulation. *Urban Transport XII. Urban Transport and the Environment in the 21st Century*, *1*, 287–296. https://doi.org/10.2495/UT060291

Jin, W. L. (2010). Continuous kinematic wave models of merging traffic flow. *Transportation Research Part B: Methodological*, *44*(8–9), 1084–1103. https://doi.org/10.1016/j.trb.2010.02.011

Language, E. C. (2015). May 2015, (May).

Leclercq, L., Laval, J. A., & Chiabaut, N. (2011). Capacity drops at merges: An endogenous model. *Transportation Research Part B: Methodological*, *45* `(9), 1302–1313. https://doi.org/10.1016/j.trb.2011.05.007

Liu, G., Lyrintzis, A. S., & Michalopoulos, P. G. (1996). Modelling of freeway merging and diverging flow dynamics. *Applied Mathematical Modelling*, *20*(6), 459–469. https://doi.org/10.1016/0307-904X(95)00165-G

Microscopic Simulation and Calibration of an Integrated. (n.d.), *630*(732).

Modelling the Effect of Lane-changing Mechanism in a Weaving Section. (n.d.).

Osborne, R. P. (2012). IMPLEMENTING TOLL PLAZA ANALYSIS INTO FREEPLAN By.

Terminals, V. F. (n.d.). Chapter 5.10. Fairness, *520*(April 2016), 92–99.

Xa, N. T. (2003). Class II Water Treatment Plant Operator Program Manual.

code1.m:

```
%% two car merge
% two car merge
% a_1, a_2 const, t time
count = 0;a_1 = 3; a_2 = 3;t=0.2;accident_rate_car= zeros(1,100);
% range
a = 20;  c = -10;
b = 130; d = 5;
for h=1:100
for k =1:50

    tmp1 = (b-a).*rand(1) + a;
    tmp2 = (b-a).*rand(1) + a;
    v1   = (d-c).*normrnd(0.5,0.2) + c;
    v2   = (d-c).*normrnd(0.5,0.2) + c;

    x1 = min(tmp1,tmp2); x2 = max(tmp1,tmp2);
   if x1 >=  x2-5
       count = count +1;
   elseif v1 >0 && v2>0 && v1>v2
       delta1 = v2^2/2*a_2 +v2*t; delta2 = v1^2/(2*a_1) + v1*t-v2^2/(2*a_2);
       if x2 + delta1 >= 150 && x1 +delta2 <x2-5
           count = count +1;
       elseif x2 + delta1 >= 150 && x1 +delta2 >=x2-5
           count = count +1;
       elseif x2 + delta1 < 150 && x1 +delta2 >=x2-5
           count = count+1;
       else
       end
   elseif v1<0 && v2<0 && abs(v2) >abs(v1)
       delta1 = v1^2/2*a_1 -v1*t; delta2 = v2^2/(2*a_2) - v2*t-v1^2/(2*a_1);
       if x1 - 5 - delta1 <= 0 && x2 -delta2 -5 > x1
           count = count +1;
       elseif x1 - 5 - delta1 > 0 && x2 -delta2 -5 <= x1
           count = count +1;
       elseif x1 - 5 - delta1 <= 0 && x2 -delta2 -5 <= x1
           count = count+1;
       else
       end
   elseif v1<0 && v2>0
       delta1 = v1^2/2*a_1 -v1*t; delta2 = v2^2/(2*a_2) + v2*t;
       if x2 + delta2 >= 150 && x1 -delta1 -5 > 0
           count = count +1;
```

```
        elseif x2 + delta2 < 150 && x1 -delta1 -5 <= 0
            count = count +1;
        elseif x2 + delta2 >= 150 && x1 -delta1 -5 <= 0
            count = count+1;
        else
        end
    elseif v1>0 && v2<0
        delta1 = v1^2/2*a_1 +v1*t; delta2 = v2^2/(2*a_2) - v2*t;
        if x1+delta1 >= x2-5-delta2
            count =count +1;
        end
    else
    end

end
accident_rate_car(h) = count /50;
count = 0;
end


%% two truck merge
count = 0;a_1 = 3; a_2 = 3;t=0.2;accident_rate_truck = zeros(1,100);
% range
a = 20;  c = -10;
b = 130; d = 5;
len = 11;
for h=1:100
for k =1:50

     tmp1 = (b-a).*rand(1) + a;
     tmp2 = (b-a).*rand(1) + a;
     v1   = (d-c).*normrnd(0.5,0.2) + c;
     v2   = (d-c).*normrnd(0.5,0.2) + c;

     x1 = min(tmp1,tmp2); x2 = max(tmp1,tmp2);
    if x1 >=  x2-len
        count = count +1;
    elseif v1 >0 && v2>0 && v1>v2
        delta1 = v2^2/2*a_2 +v2*t; delta2 = v1^2/(2*a_1) + v1*t-v2^2/(2*a_2);
        if x2 + delta1 >= 150 && x1 +delta2 <x2-len
            count = count +1;
        elseif x2 + delta1 >= 150 && x1 +delta2 >=x2-len
            count = count +1;
        elseif x2 + delta1 < 150 && x1 +delta2 >=x2-len
```

```
            count = count+1;
        else
        end
    elseif v1<0 && v2<0 && abs(v2) >abs(v1)
        delta1 = v1^2/2*a_1 -v1*t; delta2 = v2^2/(2*a_2) - v2*t-v1^2/(2*a_1);
        if x1 - len - delta1 <= 0 && x2 -delta2 - len > x1
            count = count +1;
        elseif x1 - len - delta1 > 0 && x2 -delta2 - len <= x1
            count = count +1;
        elseif x1 - len - delta1 <= 0 && x2 -delta2 -len <= x1
            count = count+1;
        else
        end
    elseif v1<0 && v2>0
        delta1 = v1^2/2*a_1 -v1*t; delta2 = v2^2/(2*a_2) + v2*t;
        if x2 + delta2 >= 150 && x1 -delta1 - len > 0
            count = count +1;
        elseif x2 + delta2 < 150 && x1 -delta1 -len <= 0
            count = count +1;
        elseif x2 + delta2 >= 150 && x1 -delta1 -len <= 0
            count = count+1;
        else
        end
    elseif v1>0 && v2<0
        delta1 = v1^2/2*a_1 +v1*t; delta2 = v2^2/(2*a_2) - v2*t;
        if x1+delta1 >= x2 - len- delta2
            count =count +1;
        end
    else
    end

end
accident_rate_truck(h) = count /50;
count = 0;
end

%% two supertruck merge
count = 0;a_1 = 3; a_2 = 3;t=0.2;accident_rate_supertruck = zeros(1,100);
% range
a = 20;  c = -10;
b = 130; d = 5;
len = 22;
for h=1:100
for k =1:50
```

```
 tmp1 = (b-a).*rand(1) + a;
 tmp2 = (b-a).*rand(1) + a;
 v1   = (d-c).*normrnd(0.5,0.2) + c;
 v2   = (d-c).*normrnd(0.5,0.2) + c;

 x1 = min(tmp1,tmp2); x2 = max(tmp1,tmp2);
if x1 >=  x2-len
    count = count +1;
elseif v1 >0 && v2>0 && v1>v2
    delta1 = v2^2/2*a_2 +v2*t; delta2 = v1^2/(2*a_1) + v1*t-v2^2/(2*a_2);
    if x2 + delta1 >= 150 && x1 +delta2 <x2-len
        count = count +1;
    elseif x2 + delta1 >= 150 && x1 +delta2 >=x2-len
        count = count +1;
    elseif x2 + delta1 < 150 && x1 +delta2 >=x2-len
        count = count+1;
    else
    end
elseif v1<0 && v2<0 && abs(v2) >abs(v1)
    delta1 = v1^2/2*a_1 -v1*t; delta2 = v2^2/(2*a_2) - v2*t-v1^2/(2*a_1);
    if x1 - len - delta1 <= 0 && x2 -delta2 - len > x1
        count = count +1;
    elseif x1 - len - delta1 > 0 && x2 -delta2 - len <= x1
        count = count +1;
    elseif x1 - len - delta1 <= 0 && x2 -delta2 -len <= x1
        count = count+1;
    else
    end
elseif v1<0 && v2>0
    delta1 = v1^2/2*a_1 -v1*t; delta2 = v2^2/(2*a_2) + v2*t;
    if x2 + delta2 >= 150 && x1 -delta1 - len > 0
        count = count +1;
    elseif x2 + delta2 < 150 && x1 -delta1 -len <= 0
        count = count +1;
    elseif x2 + delta2 >= 150 && x1 -delta1 -len <= 0
        count = count+1;
    else
    end
elseif v1>0 && v2<0
    delta1 = v1^2/2*a_1 +v1*t; delta2 = v2^2/(2*a_2) - v2*t;
    if x1+delta1 >= x2 - len- delta2
        count =count +1;
    end
```

```
    else
    end

end
accident_rate_supertruck(h) = count /50;
count = 0;
end

%% one car merge
a = 20;  c = -10;
b = 130; d = 5;
t = 0.2; a_1 =3;
accident_rate_oneCar = zeros(1,100);
for h = 1:100
    count = 0;
    for k = 1:500
        x1 = (b-a).*normrnd(0.5,0.2) + a;
        v1   = (d-c).*normrnd(0.5,0.2) + c;
        delta = v1^2/(2*a_1)+abs(v1)*t;
        if v1 >0 && x1 + delta >=150
            count = count + 1;
        elseif v1<0 && x1-5-delta<=0
                count =count +1;
        else
        end
    end
    accident_rate_oneCar(h) = count /50;
end

%% one truck merge
a = 20;  c = -10;
b = 130; d = 5;
t = 0.2; a_1 = 3;
accident_rate_onetruck = zeros(1,100);
len = 11;
for h = 1:100
    count = 0;
    for k = 1:500
        x1 = (b-a).*normrnd(0.5,0.2) + a;
        v1   = (d-c).*normrnd(0.5,0.2) + c;
        delta = v1^2/(2*a_1)+abs(v1)*t;
        if v1 >0 && x1 + delta >=150
            count = count + 1;
        elseif v1<0 && x1- len -delta<=0
```

```
            count =count +1;
        else
        end
    end
    accident_rate_onetruck(h) = count /50;
end

%% one car merge
a = 20;  c = -10;
b = 130; d = 5;
t = 0.2; a_1 =3;
accident_rate_oneStruck = zeros(1,100);
len = 22;
for h = 1:100
    count = 0;
    for k = 1:500
        x1 = (b-a).*normrnd(0.5,0.2) + a;
        v1   = (d-c).*normrnd(0.5,0.2) + c;
        delta = v1^2/(2*a_1)+abs(v1)*t;
        if v1 >0 && x1 + delta >=150
            count = count + 1;
        elseif v1<0 && x1- len -delta<=0
                count =count +1;
        else
        end
    end
    accident_rate_oneStruck(h) = count /50;
end
```

code2.m:

```
%%
a = 20;  c = -10;
b = 130; d = 5;
t = 0.2; a_1 =3;
accident_rate = zeros(1,100);
for h = 1:100
    count = 0;
    for k = 1:500
        x1 = (b-a).*normrnd(0.5,0.2) + a;
        v1   = (d-c).*normrnd(0.5,0.2) + c;
        delta = v1^2/(2*a_1)+abs(v1)*t;
        if v1 >0 && x1 + delta >=150
            count = count + 1;
        elseif v1<0 && x1-5-delta<=0
                count =count +1;
        else
        end
    end
    accident_rate(h) = count /50;
end
```

code3.m:

```
function D = dis(v,v0,d,a1,a2)
% model of merging
c = a1/a2;
t1 = (1/c)*(v0/a2 + (sqrt(2*a2*c*d*(1-c)+v0^2 *c)-v0)/(a2*(1-c)) );
t2 = (sqrt(2*a2*c*d*(1-c)+c*v0^2)-v0)/(a2*((1-c)));
t  = t1 + t2;
D  = d+ v * t;
```

code4.m:

```
function mcm
    mu1 = [100, 100, 100, 100, 100, 100, 100, 100]/3600;
    mu2 = [1184, 1184, 1184, 1184]/3600;
    mu3 = [1184, 1184]/3600;
    mu4 = [1184]/3600;

    car1 = [0, 0, 0, 0];
    car2 = [0, 0];
    car3 = [0];
    sum1 = zeros(1,10);sum2 = zeros(1,10);
    for i = 1:10,
       car3 = max([0], [car3(1) - generator(mu4(1),i)]);
       car2 = max([0, 0], [car2(1) - generator(mu3(1),i), car2(2) -
generator(mu3(2),i)]);
       car1 = max([0, 0, 0, 0], [car1(1) - generator(mu2(1),i), car1(2) -
generator(mu2(2),i), car1(3) - generator(mu2(3),i), car1(4) -
generator(mu2(4),i)]);
       car1 = car1 + [generator(mu1(1),i) + generator(mu1(2),i),
generator(mu1(3),i) + generator(mu1(4),i), generator(mu1(5),i) +
generator(mu1(6),i), generator(mu1(7),i) + generator(mu1(8),i)];
       car2 = car2 + [generator(mu2(1),i) + generator(mu2(2),i),
generator(mu2(3),i) + generator(mu2(4),i)];
       car3 = car3 + [generator(mu3(1),i) + generator(mu3(2),i)];

       sum1(i) = sum([car1, car2, car3]);

    end
    car3_base =[0];
    for i = 1:10
        car3_base = max(0, car3_base + sum(-log(rand(1, 8))./mu1) -
generator(mu4(1)));
        sum2(i) = car3_base;
    end
    figure(1);
    plot(1:10,sum1,'r');hold on;
    plot(1:10,sum2,'k');
end

function [value] = generator(coefficient,time)
    value = poissrnd(coefficient);
end
```

code5.m:

```
function mcm
    mu1 = [100, 100, 100, 100, 100, 100, 100, 100];
    mu4 = [1184];

    car3 = [0];
    sum1 = zeros(1,10);
    for i = 1:10,
         car3 = max(0, car3 + sum(-log(rand(1, 8))./mu1) -
generator(mu4(1)))
        sum1(i) = 100 * car3  ;
    end
    figure(1);
    plot(1:10,sum1,'r');
end

function [value] = generator(coefficient,time)
    value = -log(rand)/coefficient;
end
```

code6.m:

```
% calculate related data
clc; close all; clear all;
T1 =4; T2 = 4; T3 = 4;
N1 = 2; N2 = 2; N3 = 2;


ph1 =400 ; ph2 = 200; ph3 = 200;      % traffic flow (vehicle / hr)
A=0.3 ; B= 0.7; alpha= 0.1; beta = 0.5;%(A stands for î B stands for Иũ
mub = 1184;mu0 = 3017;
traffic_flow    = zeros(1,5);
wasted_time     = zeros(1,5);
wasted_timeBase = zeros(1,5);
driver = zeros(1,5);
for h = 1:5
    traffic_flow(1,h) = ph1 +ph2 +ph3 ;
    %% expected number in the whole merging process(including car,bus and
truck)
    % assume all the tollbooth are of the same type
    number = 0;
    for k = 1: floor(log2(T1/N1))
        lambda = 2^k * ph1 / T1;
        number = number + T1/(2^k) * Number_driver(lambda, mub,mu0);
    end
    for k = 1: log2(T2/N2)
        lambda = 2^k * ph2 / T2;
        number = number + T2/(2^k) * Number_driver(lambda, mub,mu0);
    end
    for k = 1: log2(T3/N3)
        lambda = 2^k * ph3 / T3;
        number = number + T3/(2^k) * Number_driver(lambda, mub,mu0);
    end
    driver(h) = traffic_flow(1,h) - number;

    %% total wasted time for a driver in the whole merging process
    w1 = 0 ;w2= 0; w3 = 0 ;

    for k1=1 : floor(log2(T1/N1))
        lambda1 = alpha * 2^k1 * ph1/T1;
        lambda2 = beta * 2^k1 * ph1/T1;
        w1 = w1 + (2^(k1)/T1)*( A * t_diff(lambda1, mu0,mub) + B *
t_diff(lambda2, mu0,mub));
    end
```

```
    for k2=1 : floor(log2(T2/N2))
        lambda1 = alpha * 2^k2 * ph2/T2;
        lambda2 = beta * 2^k2  * ph2/T2;
        w2 = w2 + (2^(k2)/T2)*( A * t_diff(lambda1, mu0,mub) + B *
t_diff(lambda2, mu0,mub));
    end
    for k3=1 : floor(log2(T3/N3))
        lambda1 = alpha * 2^k3 * ph3/T3;
        lambda2 = beta * 2^k3 * ph3/T3;
        w3 = w3 + (2^(k3)/T3)*( A * t_diff(lambda1, mu0,mub) + B *
t_diff(lambda2, mu0,mub));
    end
    wasted_time(1,h) = w1 + w2 + w3;
    wasted_timeBase(h)  = t_diff(traffic_flow(h), mu0, mub);

    ph1 = ph1 +10;
    ph2 = ph2 +10;
    ph3 = ph3 +10;

end

figure(1);
plot(traffic_flow , 3600 * wasted_time,'k');
hold on;
plot(traffic_flow,3600 * wasted_timeBase,'r');
xlabel('traffic_flow');ylabel('wasted time');
```

code7.m:

```
function L = Number_driver(lambda,mub,mu0)
L = lambda/(mub - lambda) + (lambda * (mub - mu0))/(lambda*(mub-
mu0)+mub*mu0);
```

code8.m:

```
% model of queueing
function t = t_diff(lambda,mu0,mub)

t = 1/(mub - lambda) +(mub-mu0)/(mu0*mub + lambda*(mub-mu0))-1/mu0;
```

code9.py:

```
import random
import math
import numpy

class Customer:
    def __init__(self, waitCoe):
        self.timeNeed = numpy.random.exponential(waitCoe,1)
        self.timeWait = 0


class Server:
    def __init__(self, index, generateCoe, waitCoe):
        self.index = index
        self.generateCoe = generateCoe
        self.waitCoe = waitCoe
        self.customers = []

time = 0
##
bound = 500
counterNum = 3
##
waitBound = 200
counters = [Server(0, 0.5, 10), Server(1, 5, 2), Server(2, 10, 1)]

while time < bound:
    # print 'Now time {0}'.format(time)
    for server in counters:
        if len(server.customers) > 0 and server.customers[0].timeWait >= 
server.customers[0].timeNeed:
            server.customers.pop(0)
        elif len(server.customers) > 0:
            server.customers[0].timeWait += 1
        if numpy.random.exponential(server.generateCoe,1) < time:
            if len(server.customers) < waitBound:
                server.customers.append(Customer(server.waitCoe))
            else:
                shortest = min(counters, key=lambda p: len(p.customers))
                shortest.customers.append(Customer(shortest.waitCoe))
    time += 1
    print sum([len(x.customers) for x in counters])
```

code10.py:

```
import random
import math
import numpy

class Customer:
    def __init__(self, waitCoe):
        ##
        self.timeNeed = numpy.random.exponential(waitCoe,1)
        self.timeWait = 0


class Server:
    def __init__(self, index, generateCoe, waitCoe):
        self.index = index
        self.generateCoe = generateCoe
        self.waitCoe = waitCoe
        self.customers = []

time = 0
##
bound = 500
counterNum = 3
##
waitBound = 200
counters = [Server(0, 0.5, 10), Server(1, 5, 2), Server(2, 10, 1)]

while time < bound:
    for server in counters:
        if len(server.customers) > 0 and server.customers[0].timeWait ==
server.customers[0].timeNeed:
            server.customers.pop(0)
        elif len(server.customers) > 0:
            server.customers[0].timeWait += 1
        if numpy.random.exponential(server.generateCoe,1) < time:
            server.customers.append(Customer(server.waitCoe))

    time +=1
    print sum([len(x.customers) for x in counters])
```

January 23, 2017

New Jersey Turnpike Authority

Known for its tremendous traffic flow, Gateway State Parkway has been a traffic hot spot for a long time. Even though the government has carried out policies such as one-way toll collection, the situation is still quite harsh. As the statistics of the Asbury Toll Plaza goes, the hour volume of central section of Gateway State Parkway during the summer peak time period can be still up to 8000~9000 transactions. If there is a way to increase the speed of toll transaction, the problem of traffic congestions on Gateway State Parkway will be relieved to a great extent. Besides, it is reasonably to get the conclusion that the traffic flow on Gateway State Parkway will be heavier in the future.

Our team was happy to claim that we have designed several mathematical models to simulate the traffic condition of Gateway State Parkway. Based on our study of the Asbury Toll Plaza on the central section of Gateway State Parkway, we found out some new road design which can not only accelerate the process of toll transaction but also make vehicles merge more safely.

According to our study and evaluation on the mathematical models we have built, we have the following suggestions:

1. **Adopt tree-like design to the shape of the road.** The idea of our tree-like merging model comes from the observation of waters. The basic idea of the tree-like merging model is to release the traffic congestions layer by layer so as to increase the hour volume of the toll plaza. According to our evaluation on this model, it is reasonable to believe that our model can relieve the traffic congestion to a considerable extent. Even though our tree-like merging pattern need longer distance, which requires more cost on the road construction than before, we found that the cost is still under the control of the government revenue, thus it is still feasible to implement our new design of the parkway.
2. **Increase the length of the merging part of the parkway.** We have also studied the behavior of the drivers during the merging process. Based on our theoretical test on the mathematical model we built, we found that a longer merging part of the road can significantly reduce the probability of car collision with other factors remain the same. Thus, in order to minimize the car accident, the government should consider our plan to lengthen the merging part of the road.
3. **A multi-layer toll procedure can increase the speed of toll transaction.** Our team has also designed a new toll transaction pattern, which is implemented by collect the toll fee layer by layer, instead of make the toll transaction at the same location on the parkway. By doing this, the toll transaction process can be accelerated, which means that there will be less traffic congestion and larger hour volume for the toll plaza. We believe this new toll pattern, together with the tree-like parkway shape design we put forward above, can increase the efficiency of toll collection and serve more Gateway State Parkway user.

In spite of the fact that our mathematical model is based on some pure theoretical assumption, and the limitation that we only choose Asbury Toll Plaza as our case study, our model can still fit the real Gateway State Parkway condition, which will contribute to the traffic improvement of the New Jersey State. According to our theoretical computation, adopting our ideas can increase 20%~30% of the hour volume of Gateway State Parkway.