

For office use only

Team Control Number

For office use only

T1 \_\_\_\_\_

**0000**

F1 \_\_\_\_\_

T2 \_\_\_\_\_

F2 \_\_\_\_\_

T3 \_\_\_\_\_

Problem Chosen

F3 \_\_\_\_\_

T4 \_\_\_\_\_

**B**

F4 \_\_\_\_\_

---

**2017**  
**MCM/ICM**  
**Summary Sheet**

## Modeling merge after toll

### Summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Keywords:** keyword1; keyword2

# Modeling merge after toll

January 23, 2017

## Summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Keywords:** keyword1; keyword2

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Other Assumptions . . . . .	3
<b>2</b>	<b>Analysis of the Problem</b>	<b>3</b>
2.1	建立模型 . . . . .	4
<b>3</b>	<b>The Model Results</b>	<b>5</b>
3.1	并道方式选择 . . . . .	5
3.1.1	单次并道 . . . . .	7
3.1.2	多次并道 . . . . .	7
3.2	事故预防 . . . . .	7
3.3	车流量大小的影响 . . . . .	8
3.4	自动驾驶车辆的影响 . . . . .	10
3.5	电子支付通道的影响 . . . . .	11
<b>4</b>	<b>Conclusions</b>	<b>12</b>
<b>5</b>	<b>Evaluate of the Mode</b>	<b>12</b>
<b>6</b>	<b>Strengths and weaknesses</b>	<b>12</b>
6.1	Strengths . . . . .	12
6.2	Weaknesses . . . . .	13
	<b>Appendices</b>	<b>15</b>
	<b>Appendix A First appendix</b>	<b>15</b>
	<b>Appendix B Second appendix</b>	<b>15</b>

# 1 Introduction

高速公路的收费站会通过过卡收费的方式收取驾驶员的高速费，过卡收费时，由于收费窗口的数量通常要比车道数量多，因此当汽车交费后从收费站驶出时，车流必须从较宽的呈扇形收费站广场快速并入较少的常规的机动车车道，收费广场的建立就是为了解决在这个过程中产生的拥堵现象。优化收费广场的建立方案，使得在更小的占地面积内达到最高的交通效率。

为了增强车辆通行能力，增大吞吐量，减小交通事故发生的可能性，简单的解决方案就是增加收费广场长度，以减小并线压力，增加长度同样也增大了收费广场的面积，但收费广场面积越大，建设成本也会随之升高，因而需要寻找在车辆通行能力最佳的条件下，使得收费站形状最优、大小最小的收费站区域设计方案。除此之外，车道合并的模式对收费站的形状大小起着决定作用，因而对合并车道方案进行详细说明，评价方案包括下列指标。

- 不同合并方式下的出流量。
- 不同合并方式所占用的面积。

我们研究了在不同形状、大小、合并方式的情况下对收费站拥堵情况的影响，提出了基于离散化车流密度的元胞自动机模型。

对于电子收费窗口，与不找零收费窗口，由于其通过速度快，可能反而会造成合并区域的拥堵现象 [1]。到目前为止，全世界绝大部分地区的收费方式存在人工收费窗口与电子收费窗口比例不协调的现象，这使得过多电子收费窗口闲置而人工收费窗口拥堵。通过收费窗口后，几乎所有收费广场都是采取扇形区域来实施汽车的合流，因此当遇到高峰时，收费广场拥堵现象仍然普遍存在。

An autonomous vehicle is a vehicle that is capable of sensing its environment by using a variety of techniques such as radar, lidar, GPS, odometry, and computer vision and navigating without human input. Those techniques can interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage to arrive the desired destination.[2]

通常情况下，收费站有三种不同收费方式：传统收费方式、不找零收费方式和etc 收费方式，它们各自占有一定的比例。

- 传统收费方式是指车辆行驶到收费亭时需要停车，等待通知账单，现金支付费用，等待找零，收到零钱后离开。(最慢) 见图 \*
- 不找零 (coins-only)：车辆行驶到收费亭时需要停车，需要投入大于或等于要求的价格，且机器不找零，投币后离开。(比传统方式快，但比 ETC 慢)
- Etc：当车辆通过收费站时，车内的收发器接收到收费站内可发出收费信息的感应器的信号，即可完成缴费，这种收费方式道路可大大提高收费站的通行能力。(最快)

## 1.1 Other Assumptions

为了简化模型，我们做出了如下假设

- 假设所有从收费窗口流出的汽车车流量服从泊松分布
- 假设高速公路的出入分别是分离的。
- 高速公路入口收费是分离的，出入互不影响。
- 

## 2 Analysis of the Problem

收费站区域一般都为扇形区域，如图所示，收费站  $B$  条出口，并入  $L$  条常规路线，其中  $B$  大于  $L$ 。Toll merge area show as 1,[3] 我们假设所有通过收费窗口的车流

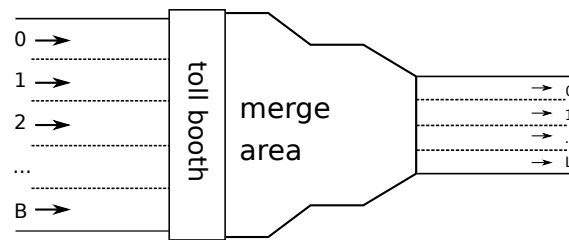


Figure 1: merge area

分布程泊松分布，即越靠近中间车道的收费窗口通过的车流量越大。从收费站通过在扇形区域的合并模式，由于收费站广场的长度不同，分为两种合并方式。单次并道与多次并道，由合并位置的不同分为单侧并道与双侧并道，如 Figure 2所示。We had considered 3 way to merge, 1) direct merge, 2) single side multiple merge, 3) double side multiple merge. 衡量收费窗口通行能力的主要因素就是考虑车辆通

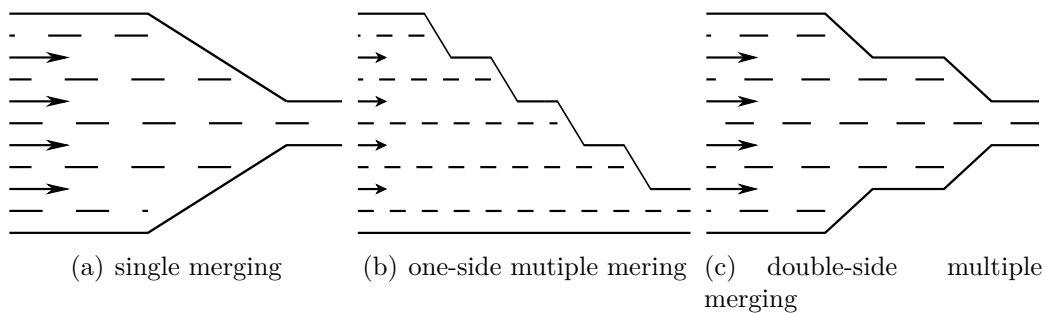


Figure 2: merging ways

过收费广场的最大通行容量，即最大不发生拥堵的车流量，将合并区域离散化，建立元胞自动机模型。

## 2.1 建立模型

车辆通过收费窗口之后，车流量对于收费窗口的分布服从泊松分布。为定量模拟收费广场合并时广场形状和面积对车流量的影响，将收费广场网格化，分割成为有限个、状态连续的单元格，每一小格可近似看作一个单元（unit），每个单元的取值为该单元的车流密度（Density  $K$ ）（“元胞”的车流量定义为单位时间内在该元胞上通过的机动车数量）。散布在规则格网中的每一单元都遵循同样的运动规则，依据确定的局部规则作同步更新。大量元胞通过简单的相互作用而构成动态系统的演化。由左侧第一列产生进入流量  $Q$ ，所有流量的传播方向如 Figure 3所示。所

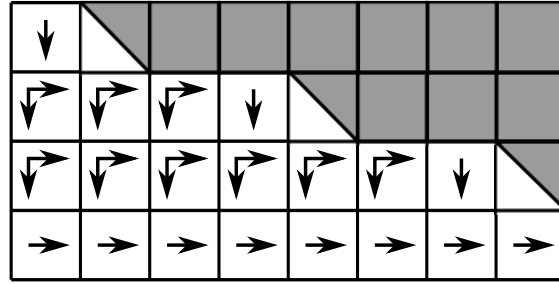


Figure 3: road discrete

有单元按照传播方向与周围格子的状态决定传播流量，传播流量的大小由传播函数控制，如 Equation 1所示，其函数图像为 Figure 4。

$$F(K) = \frac{2}{1 + e^K} \quad (1)$$

对于传播流量  $Q$  有

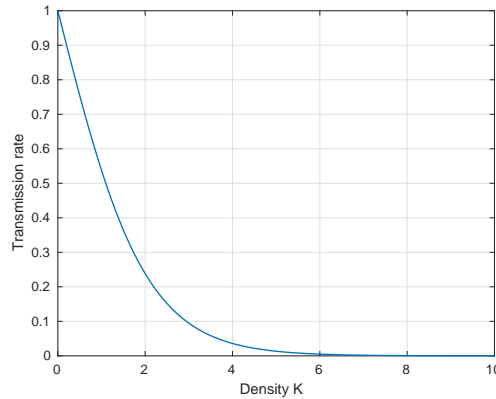
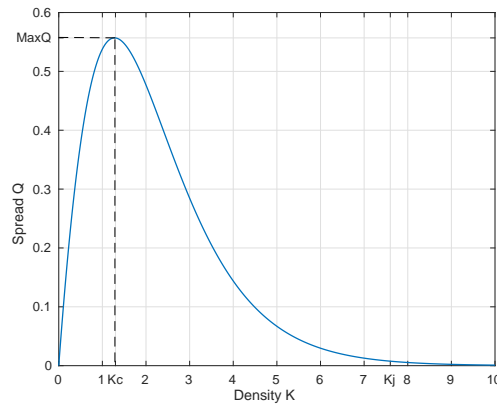


Figure 4: s function

$$Q = K \cdot F(K)$$

所以 Density  $K$  与 Flow  $Q$  的关系如5所示，其中  $K_c$  为 critical density,  $K_j$  为 jam density,  $MaxQ$  为最大可能传播流量。图像如 Figure 5所示。每个单元格达到 7 的车流量视为饱和。如果某单元格只有一个传播方向，那么此单元格只需简单通

Figure 5: Spread flow  $Q$  for density  $K$ 

过传播函数决定传播流量的多少到下一个单元格，如 Figure 6(a)所示，颜色越深代表流量越大。

若某单元格具有不同的选择，即有两个传播方向，单元格首先会根据传播方向上的流量的不同而决定一个主要的传播方向，比如：如果某单元格下方流量较少而前方流量较多，则传播流量的主要部分会集中在下方，伪代码如 Algorithm 1所示，传播方式如 Figure 6(b)所示，其中  $Q' > Q$ 。

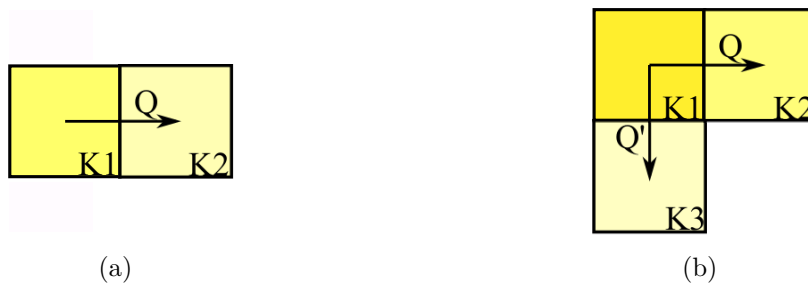


Figure 6: spread mode

当道路流量上升时，发生事故的概率也会增大，ETC may lead[1]

### 3 The Model Results

#### 3.1 并道方式选择

结合所建立的模型，为求解合并区域的最优形状，定量分析当  $B=10$ ,  $L=2$  时进行仿真，车辆到来概率如下图，横坐标代表收费窗口  $B_i$ ，纵坐标代表收费窗口  $B_i$  的来车率，如 Figure 7所示。

利用上述建立的模型，对合并区域长度对道路通行能力的影响进行仿真，将合并区域长度分为单次并道（短）和多次并道（长），仿真结果如下。

---

**Algorithm 1** Flow Control

---

**Require:** Self density  $k$ , Forward density  $kl$ , Downword density  $kd$ **Ensure:** 流量控制结果

```

function FLOWCTL( $k, kd, kl$ )
  if  $kd < kl$  then
     $k, kd \leftarrow \text{DOWNWARD}(k, kd)$ 
     $k, kl \leftarrow \text{FORWARD}(k, kl)$ 
  else
     $k, kl \leftarrow \text{FORWARD}(k, kl)$ 
     $k, kd \leftarrow \text{DOWNWARD}(k, kd)$ 
  end if
  return  $k, kd, kl$ 
end function

```

---

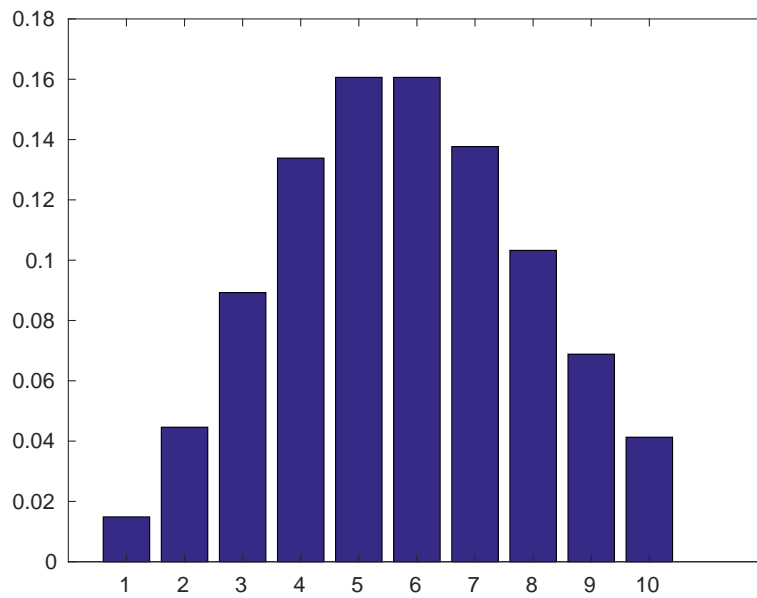


Figure 7: poisson distribution



### 3.1.1 单次并道

单次并道是指从高速收费亭的  $B$  个收费口出来到  $L$  个车道的高速公路的过程中, 只需要一次车道合并。在此种情况下, 在合并过程中, 多个车道同时合并, 容易发生交通事故, 且多车同时并道容易造成道路拥挤。通过以上建立的模型利用 matlab 仿真实现, 如 Figure 8所示。最终出流量为 71.4。

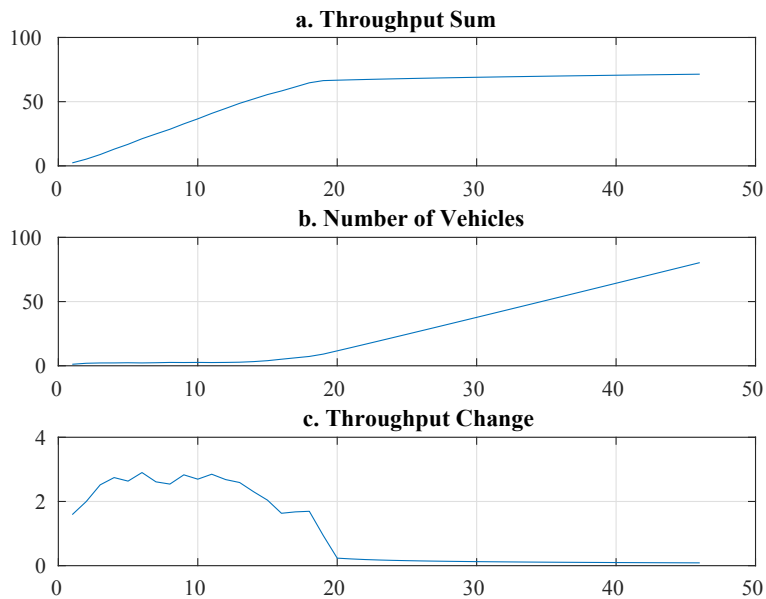


Figure 8: direct out

### 3.1.2 多次并道

多次并道是指从高速收费亭的  $B$  个收费口出来需要多次车道合并才能到达  $L$  个车道的高速公路的并道模式。在此种情况下, 道路合并次数可以根据高速公路的车道数而定, 且在并道过程中, 机动车驾驶人可选择道路中车辆数少且更加便捷的路线并道, 即更加便捷、选择更多的减少交通事故的发生。在此情况下, 可对单侧并道、双侧并道的情形分别研究, 通过 matlab 程序实现, 结果 Figure 9所示。最终出流量为 93.1。

以上结果显示, 多次并道, 即适当增加合并区域的长度, 会有效改善道路的通行能力, 因此在造价可以接受的范围内, 应尽量增加合并区域的长度, 以提高道路的通行能力。

## 3.2 事故预防

根据 Zhou, Min and Sisiopiku 的研究 [4] 当交通密度增大时事故发生的概率也会增高。由仿真得到交通密度如 Figure 10所示。根据流量判断哪条道路上发生事故

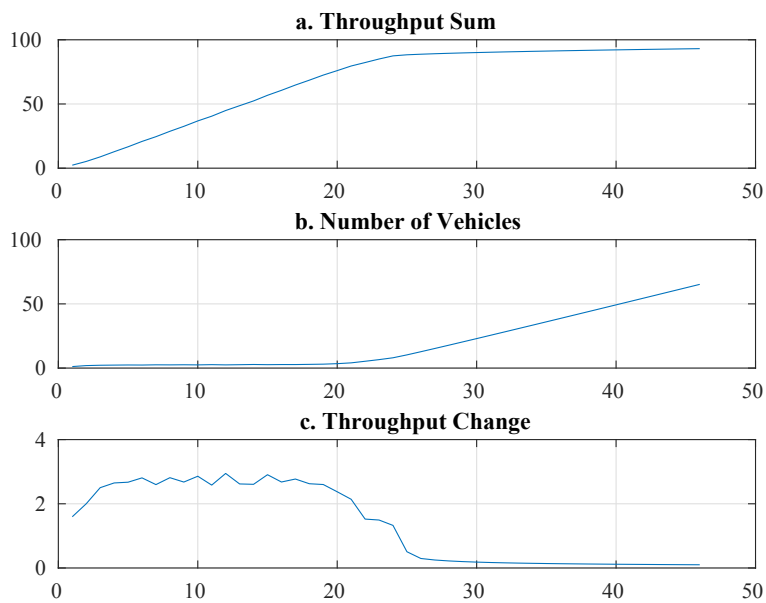


Figure 9: multiple out

的概率更高，提前做好预防。

图中黑白色区域代表出收费站后的收费广场区域，图中黄色部分表示随着时间的推移，道路上车流量的变化，颜色的深浅代表车流量多少。颜色越深表示车流量越大。

### 3.3 车流量大小的影响

根据上述所建立的模型，分别研究收费站在大、小车流量下的通行效率。当车流量较小时，道路车流密度较小，车辆能优先选择便捷、安全、合适的收费口进入收费站，出收费站后的道路合并也可优先选择道路畅通、便捷的车道通行。这种情况下，车流量小、道路密度小，车辆通行快。改变上述模型中进入收费广场的车辆数目，通过 matlab 运行，结果如 Figure 11，可见当车流量较小时，道路通行能力维持在最高水平。

当车流量较大时，道路车流密度大，道路较为拥挤，结果如 Figure 12所示，可见当车流量较高时，道路通行能力很快饱和。

在 Figure 11和12中，Throughput Sum（图像的名字可能还会有变动）图像表示随着时间的变化，经过收费广场的总车流量变化；Number of vehicles 图像表示收费广场上的车流密度的和即机动车数量随时间变化的曲线；Throughput Change 图像为流出收费广场的车流量随时间变化的图像。

Figure 10 中为小流量的情况，Figure 11 为大流量的情况。通过图像给出的直观信息，由 Figure 10 中的 throughput sum 可以看出，道路流量较小时，通过收费广场的总车流量随时间整体呈现近乎线性增长趋势，说明流量较小时，收费广场压力小，车辆能自由/无压力通行；从 Number of vehicles 中可以看出，收费广

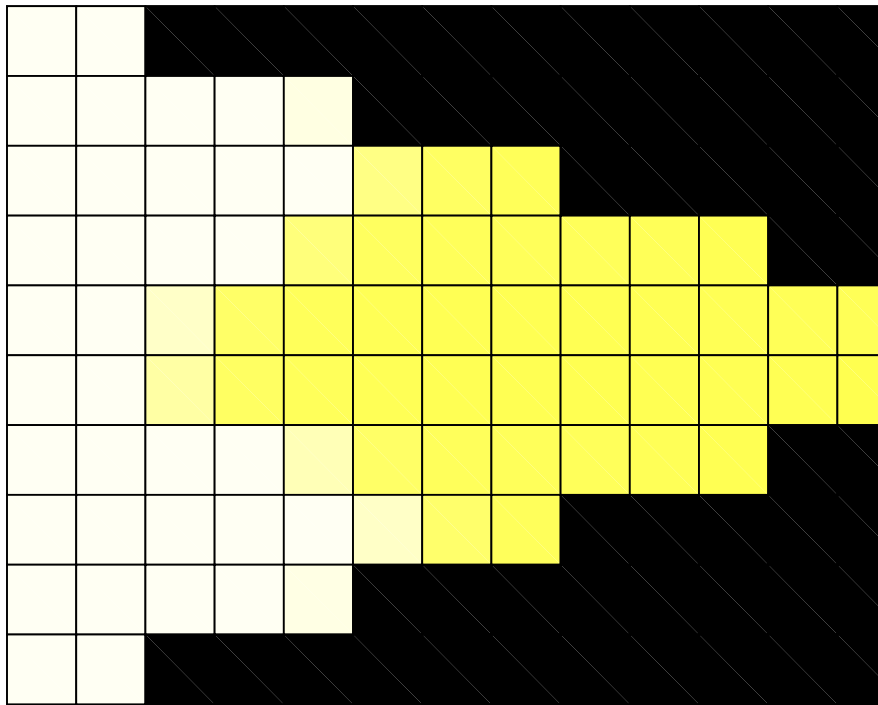


Figure 10: q map

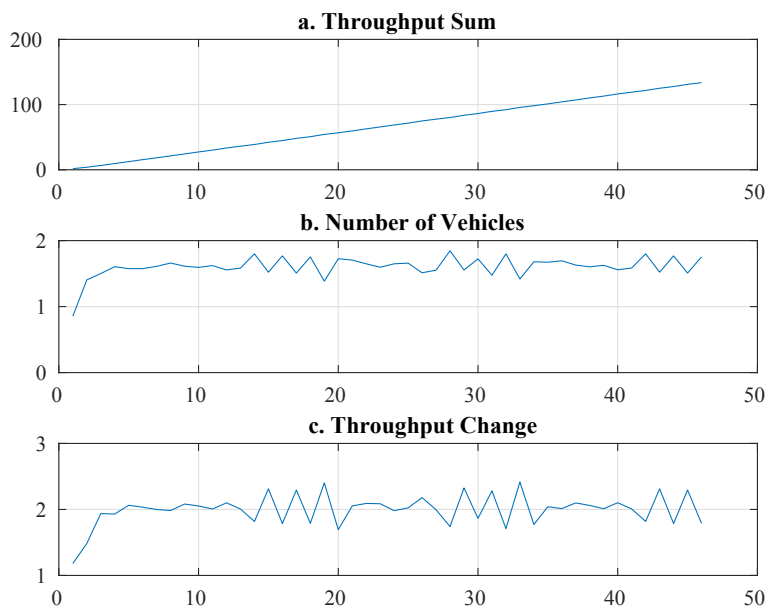
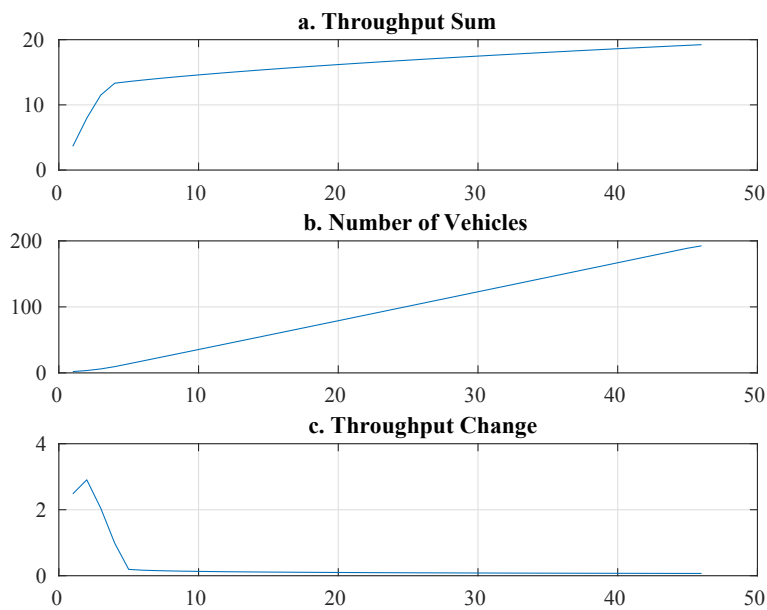


Figure 11: low q

Figure 12: high  $q$ 

场上的流量密度在几乎一直在一个流量密度附近波动；从 Throughput Change 图像得出，流出收费广场的车流量也几乎始终在最大车流量值附近波动。说明在较小流量情况下，收费广场几乎不存在交通拥挤问题。由 Figure 11 的 a 图可以看出，随着时间的变化，通过收费广场的车辆数在刚开始的一小段时间内，呈线性递增趋势，之后变化速率减缓（斜率变小），说明经过收费广场的车流量受到阻碍，有明显减少。B 图展示了收费广场流量密度近乎线性增加，说明此时收费广场上车辆数呈线性增加，即车辆进出收费广场不均衡，“进多出少”；图 c 指出流出收费广场的车流量从一开始的比较大迅速降到很低的数值，说明此种情况下，车辆流出收费广场特别慢，显然为堵车状态，不利于车辆通行。

### 3.4 自动驾驶车辆的影响

自动驾驶是一项新兴技术，随着自动驾驶汽车数量逐渐增加，由于自动驾驶技术的特性，道路容量将会下降自动驾驶车辆通过传感器识别与其他车辆之间的距离，保持车距和车速。自动驾驶汽车与人工驾驶汽车的主要区别在于：自动驾驶汽车通过汽车车身的传感器，感知与周围车辆的距离，保持与周围车辆的车距、与障碍物之间的距离，以保证车及车内乘客的安全，且在车流量大、道路密度较大的情形下，比如收费站，自动驾驶车辆会与周围车辆保持一定距离以确保车辆与人的安全，增大了车间距，导致道路车流密度降低，因而道路通行能力降低。加入自动驾驶车辆后，会使道路饱和和密度降低，

Table 1: 自动驾驶车辆与普通车辆的区别

self-driving		人工
原理	传感器感知周围路况	人对周围交通状况的感知
在车流量较大时	保持安全车距 (较大)	安全车距更小

### 3.5 电子支付通道的影响

当车辆从传统收费站通过时，车辆需要停车接受账单，并向收费亭工作人员支付现金，等待找零。服务时间较长，且容易受服务人员工作效率的影响，使得道路通行能力降低。通过不找零收费站时，车辆需要停车接受账单，但是此时，司机只能支付 coins，不需找零，相较于传统服务时间，此种时间有所减少。通过 etc 通道时，车辆无需停车，且不使用现金交易，避免找零，减少停车等待时间，提高了道路的通行能力，使得道路车流密度增大 [1]。

不同支付方式的流程如 Figure 13

增加电子支付通道，会提高该道路流量。下图是开通之后

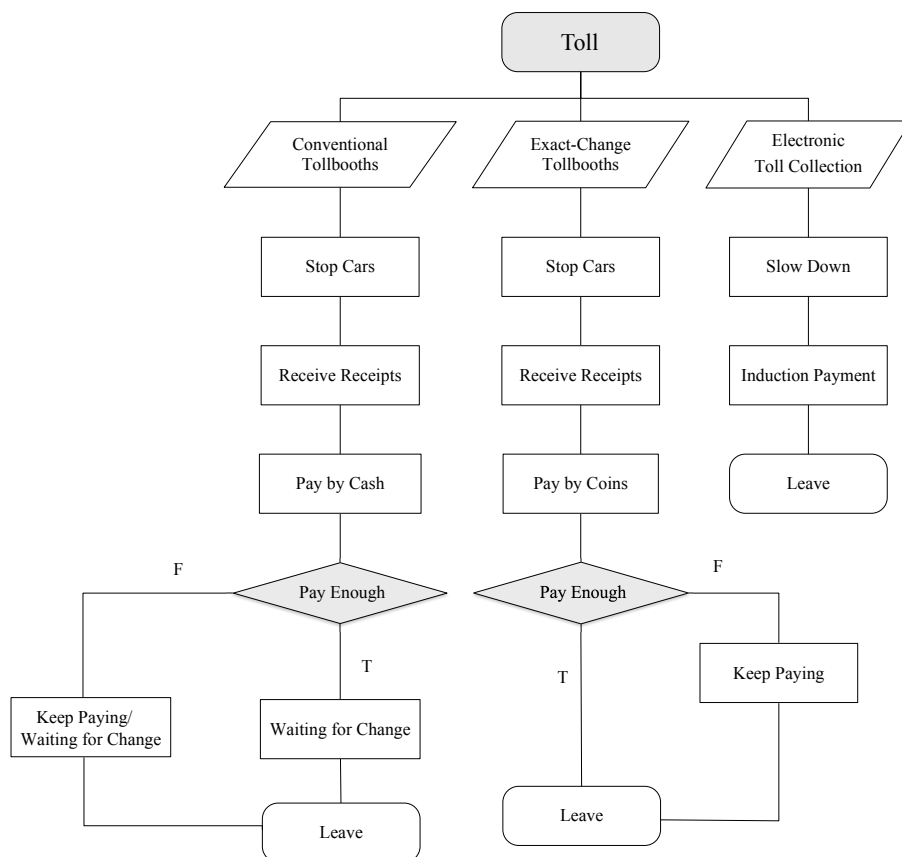


Figure 13: flow chart of three different toll methods

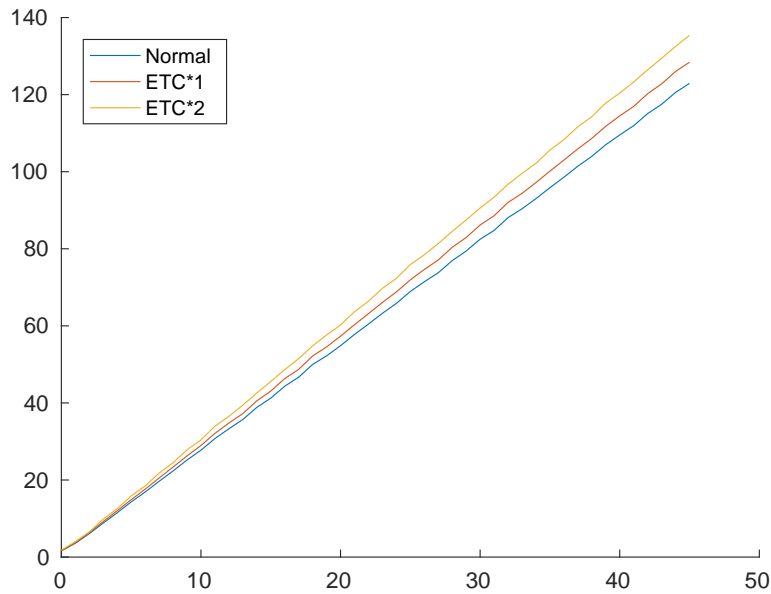


Figure 14: etc

## 4 Conclusions

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 5 Evaluate of the Mode

## 6 Strengths and weaknesses

### 6.1 Strengths

- **Applies widely**

利用了差分的思想。收费广场上车辆通行过程中，道路车流量时刻发生变化，整体考虑整个收费广场的车流量变化十分困难，利用局部车流量代表整体车流量，也不具有实际意义。我们采取将收费广场网格化，将整体分解成有限个单元格，能充分分析每个单元格上车流量的变化，进而得到整体的流量变化。

- **Improve the quality of the airport service**

基于元胞自动机的改进模型，由元胞自动机的连续状态转移，即 0 代表无，1 代表有，联系收费广场的车流量，将元胞的取值连续化，单位时间内每个元胞上的车流量赋值给元胞，元胞间的转移，延伸为元胞间车流量的转移。

- **balabala**

考虑自动驾驶汽车时，从自动驾驶汽车与人工驾驶汽车的区别入手，将自动驾驶汽车简化为人工驾驶汽车时元胞车流量减少的情形，量化自动驾驶与人工驾驶的区别。

## 6.2 Weaknesses

- **balabala**

基于元胞自动机的收费广场规划模型对于解决收费广场面积大小不具有优势，因而解决成本有一定的难度，只能从车流量的角度衡量收费区域设置的合理性。

## References

- [1] Anastasia D Spiliopoulou, Ioannis Papamichail, and Markos Papageorgiou. Toll plaza merging traffic control for throughput maximization. *Journal of Transportation Engineering*, 136(1):67–76, 2009.
- [2] Wikipedia. Autonomous car — Wikipedia, the free encyclopedia, 2017. [Online; 22 January 2017].
- [3] T Hugh Woo and Lester A Hoel. *Toll plaza capacity and level of service*. Number 1320. 1991.
- [4] Min Zhou and Virginia Sisiopiku. Relationship between volume-to-capacity ratios and accident rates. *Transportation Research Record: Journal of the Transportation Research Board*, (1581):47–52, 1997.



# Appendices

## Appendix A First appendix

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Here are simulation programmes we used in our model as follow.

### Input matlab source:

---

```
function [t,seat,aisle]=OI6Sim(n,target,seated)
pab=rand(1,n);
for i=1:n
    if pab(i)<0.4
        aisleTime(i)=0;
    else
        aisleTime(i)=trirnd(3.2,7.1,38.7);
    end
end
end
```

---

## Appendix B Second appendix

some more text **Input C++ source:**

---

```
//=====
// Name      : Sudoku.cpp
// Author     : wzlf11
// Version    : a.0
// Copyright  : Your copyright notice
// Description : Sudoku in C++.
//=====
```

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int table[9][9];

int main() {

    for(int i = 0; i < 9; i++){
```

```
        table[0][i] = i + 1;
    }

    srand((unsigned int)time(NULL));

    shuffle((int *)&table[0], 9);

    while(!put_line(1))
    {
        shuffle((int *)&table[0], 9);
    }

    for(int x = 0; x < 9; x++){
        for(int y = 0; y < 9; y++){
            cout << table[x][y] << " ";
        }

        cout << endl;
    }

    return 0;
}
```

---