For office use only	Team Control Number	For office use only
T1 ________________	**66005**	F1 ________________
T2 ________________		F2 ________________
T3 ________________	Problem Chosen	F3 ________________
T4 ________________	**C**	F4 ________________

2017
MCM/ICM
Summary Sheet

A Modified Traffic Flow Model for Self-driving Cars

Summary

In this paper we use self-driving cars to alleviate the congestion and increase the traffic flow. We modify typical cellular-automatic model to measure the effects on traffic flow of the percentage of self-driving and cooperating cars, the number of lanes as well as the peak traffic volume.

First, we find out two index to describe the condition of road traffic:vehicle density which represents cars' utilization rate of the road and traffic flow which intuitively shows the level of road congestion. We need to measure the effects of self-driving cars on vehicle density and traffic flow respectively.

To measure the effects scientifically, we modify typical cellular-automatic model to simulate the non-self-driving cars on the road. The main differences between these two types of cars contain whether to change lanes when there is more space on neighbor lanes, whether to keep a safe distance from the back car, and whether to reduce the speed randomly. We build different speed and position changing rules for human drivers and self-driving cars. After simulation, we conclude that the growth of the percentage of self-driving cars leads to the decline of density and increase of flow.

Finally, we apply our model to the data provided. We pick out the start of I90 in downtown Seattle and part of Interstate 5 as two congested sections, then use our model to analyze the effects of self-driving cars under a congestive road condition. Satisfyingly, we find when the percentage of self-driving cars is over 90%, density will reduce, which means that it would be better to add at least 90% self-driving cars in order to solve these two congestion problems.

A Modified Traffic Flow Model for Self-driving Cars

Control Number : # 66005

January 24, 2017

Contents

1 Introduction

1.1 Statement of the Problem

Traffic problems have attracted considerable attention during the past decades, and there is no doubt that traffic congestion becomes a growing worry for the residents of most urban areas. Major roads in many regions of the United States are regularly choked with traffic in rush hours due to the limited transport capacity. For example, as the problem illustrated, in the Greater Seattle area drivers experience long delays during peak traffic hours because the volume of traffic exceeds the designed capacity of the road networks. This is particularly pronounced on Interstates 5, 90, and 405, as well as State Route 520, the roads of particular interest for this vexing problem.

Traffic flow is complex and contains rich features, which is currently a heated issue under wide public concern. Increasing traffic flow enables cars to operate closer to optimal speed and to reach their destinations sooner, consequently reducing fuel consumption and greenhouse gas emissions. In a word, increasing highway efficiency will definitely have a significant impact on environment. Typical solutions, such as the expansion of roadway systems or road improvement, do not work well anymore for several social, physical and economical reasons.(Rillings,1997)[1] In addition, human behavior is conspicuously difficult to change, so there was limited way to dramatically increase highway traffic flow. Using self-driving cars is an effective method to solve the problem and maintaining the original lanes or roads at the same time. Self-driving cars are able to detect surroundings with a variety of science techniques such as radar, GPS, odometry and computer vision, which represent a technological leap forward, having the potential to improve sustainability and efficiency. What is really useful in panning a path to the desired destination is the advanced control system Autonomous cars, which is capable of analyzing sensory data to distinguish between different cars on the road. Once a critical mass of self-driving vehicles system has been established, network benefits enable environmental, safety, and travel time improvements.

The goal of this paper is to investigate how the traffic flow is influenced by the number of lanes, peak average traffic volume and percentage of vehicles using self-driving, cooperating systems. Meanwhile, the model also address cooperation between self-driving cars as well as the reciprocal action between self-driving and non-self-driving vehicles, being applied to the data for the specified roads in Washington.

1.2 Flow Chart

To begin with, we need to figure out factors which influence the traffic flow, vehicle density and car speed. Following are our primary ideas.

Since vehicle density is the ratio of total car space to the whole road space, it poses impact on traffic flow. Transportation phenomenon and some other objective factors are part of the factors which influence the car density. For example, Highway 5 is a rather complex road, which is correlate with several intersection roads like Highway 510 and Highway 101. It is clear that the number of cars on Highway 5 definitely has an increasing trend with more intersection roads. Meanwhile, the fact that the same road with more cars suggests high possibility of congestion. It is hard to get a satisfied traffic flow. However, vehicle density is changing. In our modified traffic flow model, we investigate the relation of vehicle density and different proportion of self-driving cars later.

Since we do not want to build more roads, we try to introduce self-driving cars to improve vehicle speed so that the traffic flow can be improved as well. Self-driving cars are with hypersensitivity, which means they can take more timely reactions. Equipped with cooperating systems, self-driving cars are able to interact with each other and make timely decisions on how to change their speed and whether to change a lane or not, thus the average speed can be improved. It is difficult for human beings to make a perfect decision whether to change a lane because compared with non-self-driving cars, they do not know the speed of other vehicles. Sometimes what can human beings do is only to wait in line. However, self-driving cars can overcome this shortcoming.

In this paper, we use modified cellular automaton model and our main purpose is to investigate the relationship between traffic flow and the different proportion of self-driving cars. We conclude:

- The relation between traffic flow and 10%-90% self-driving cars.
- The relation between vehicle density and 10%-90% self-driving cars.

In the end, we explore the differences among them.

2 Baseline CA Model

2.1 Background

2.1.1 NS Model and BML Model

During the past decades, scientists have been trying to develop several models of traffic flow by incorporating only the most essential ingredients which are

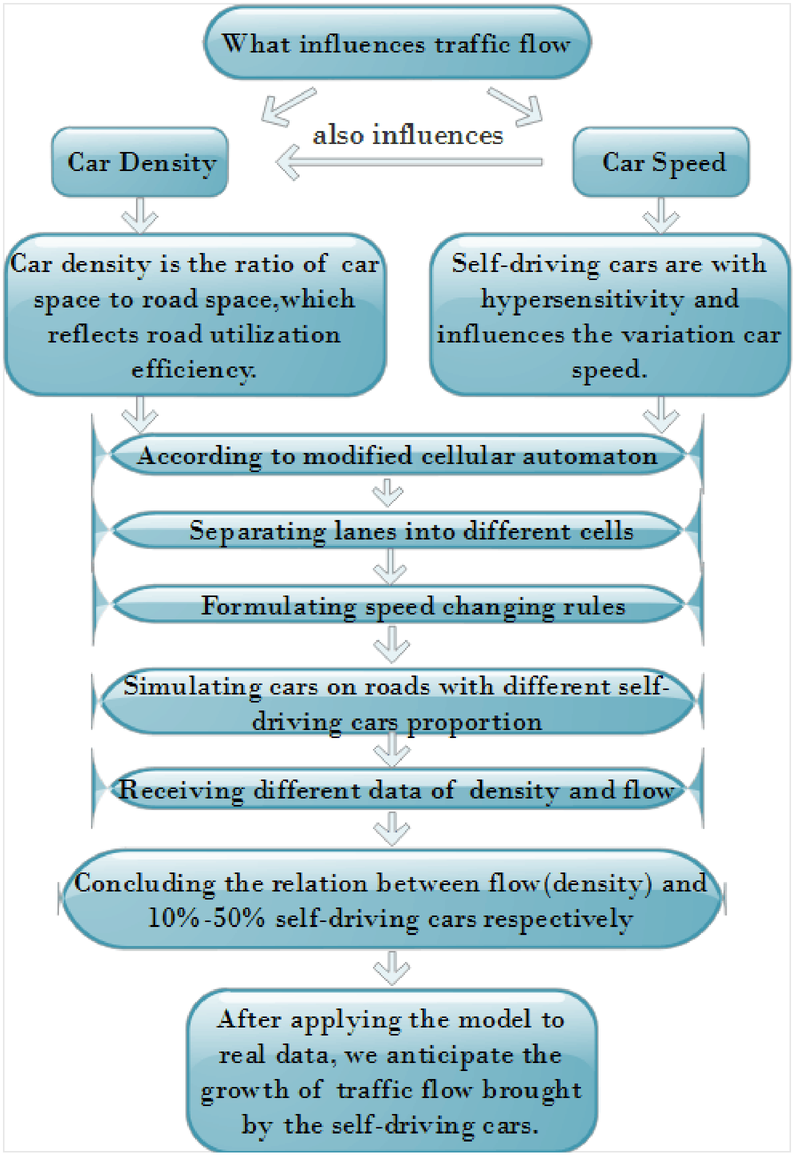

Figure 1: Flow Chart

absolutely necessary to describe the general features of typical real traffic. In the 1940s, J. von Neumann was the first one who proposed the cellular automata theory in order to simulate the self-replicating function of living system. In the late 60s, the Game of Life programed by J. H. Conway became the most famous cellular automata model. Then the US scientist S. W-olfram systematically studied cellular automata and classified the original ideals in academic way, finally promoted the theory to scientific methodology in the 1980s.

The cellular automata model (CA) started in statistical physics in the study of particle behavior, then developed from dynamic models to describe transportation phenomena. Space is discrete and consists of a regular grid of cells, every one of which can be in one of a finite number of possible states. The number and array of cells in the grid depends on the specific traffic phenomena being modeled. All cell states are updated contemporaneously in discrete time steps. Updating obeys a limited set of local interaction rules that can have probabilistic influence. The new state of a cell is determined by the actual state of the cell itself and its neighbor cells. This local interaction allows to capture micro-level dynamics and transmit it to macro-level behavior. From the perspective of traffic flow, it is possible to relate the cell states with significant quantities, taking the travel time, vehicle speed, pollution and throughput for instance. Nowadays, the CA model has been proved to be quite useful in the field of vehicular traffic flow as well as in other fields such as escape and panic dynamics, pedestrian behavior, etc.[2]

In 1922, there is not only the one-dimensional traffic flow CA model (NS model) put forward by German scholars K. Nagel and M. Schreckenberg to describe the movement of vehicles in traffic flow, but also the two-dimensional traffic flow CA model (BML model) proposed by O. Biham and other American scholars. The Biham−Middleton−Levine (BML) model serves as a theoretical foundation for the physicists' approach to simulate urban traffic. There are two species of vehicles, eastbound and north-bound, populating on the two-dimensional square lattice randomly with the same densities at the beginning of the simulation in the BML model. Both eastbound and northbound vehicles are not allowed to change either their directions or rows (columns). Otherwise, if a vehicle is blocked ahead by other vehicles and its side nearest-neighbor space is unoccupied, it seldom remains stationary. But in actual transportation phenomenon, if drivers encounter this situation and not break traffic regulations, most of them will choose to change lanes, thus the behavior of vehicles' changing lanes has significant influence on the spatial correlation. Because of the fact that vehicles will interact with each other directly or indirectly, the spatial correlation is strong in real traffic situation. Consequently, it is necessary to study its impact on the traffic flow, which is one of the motivations of this paper.[3]

2.1.2 Reason of Choosing CA Model

Cellular Automaton Model has following assumptions:

- Time, space, velocity are all discretized.
- Roads are divided into discrete cells, each of which is empty or occupied by one car.
- The speed of every car can be taken to be $0, 1 \ldots v_{max}$.(v_{max} is the maximum velocity)

From the assumptions above, it can be seen that cellular automaton model focuses on dynamic behavior of every single car, which is the main reason we choose to build our model. Based on cellular automata model, we have diverse dynamic behaviors such as different speed changing norms for human drivers and self-driving cars. Then we start from dynamic behavior through investigating the cooperation between individual cars, finally deduce the statistical property of the whole system.

In actual operation, cellular automaton allows us to intuitively observe traffic flow through the simulation on Matlab. We can also identify the vehicle density and measure the value of traffic flow under our definition. As a result, we are able to figure out the relationship between vehicle density and traffic flow, which assists us in applying our model on the provided data and predicting how much the traffic flow will increase when we raise the percentage of self-driving cars.

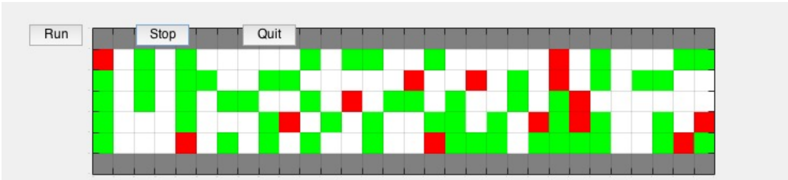

Figure 2: A Simulation of Traffic Flow on Matlab

2.2 One-lane CA Model for Human Drivers

2.2.1 Basic Structure of One-lane Model

The one lane road is viewed as a regular grid of cells, each one of which contains one car. Cars are randomly placed on it. The number of cells in the grid depends on the specific transportation phenomena being modeled.[2] So in different conditions, we need to choose a proper length for one cell and a proper interval for our time loop. Generally, the length of a cell is around 7.5 m. It can be interpreted as the length of a vehicle plus the distance between vehicles in a jam, and as we said, it can be slightly adjusted according to the practical problem. The interval is taken to be 1s.

Some definitions:

- n .... the total number of cells
- N .... the total number of cars on the road
- v_i .... the velocity of vehicle i, i=1,2...N
- x_i .... the position of the vehicle i, i=1,2...N
- v_{max} . the limit speed of cars, often values 5 based on the assumption above
- p .... the probability of deceleration of human drivers

- d_i .... the distance between vehicle i and the vehicle ahead vehicle i
- $\bar{v}$ .... average velocity of cars in the whole loop

So we have the following model:

- Step 1
  If $v_i < v_{max}$, the velocity of vehicle i is increase by one
  $v_i = \min(v_{i+1}, v_{max})$
- Step 2
  If v_i>0, the velocity of vehicle i is decreased randomly by one with probability p
  $v_i = \max(v_{i-1}, 0)$with probability p
- Step 3
  If $d_i < v_i$, reduce sufficiently to avoid the crash
  $v_i = d_i$ when $d_i < v_i$
- Step 4
  Vehicle movement. Each vehicle is moved forward according to its new velocity determined in steps 1-3

Now we set $v_{max} = 5, p = 0.2, N = 5000$,and use 5000 iterations to simulate the variation of traffic flow. Under different density(from 0.1 to 0.9,calculated by N/n),we get different values of traffic flow(calculated by $\frac{N \times \bar{v}}{n}$).Then we can have an intuitive judgement of the relationship between traffic density and traffic flow.

From the Figure 3, when the density of cars is very low, an increase in density leads to a proportional increase in flow. On the contrary, when the density is high enough, meaning that the road is quite congested, the increase in density decreases the traffic flow. This is consistent with common sense.

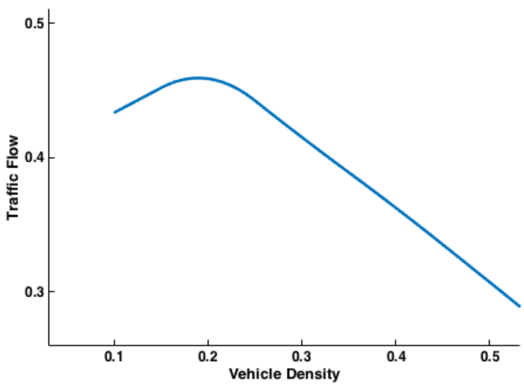

Figure 3: Relationship of Traffic Flow and Vehicle Density

2.2.2 Defect of One-lane Model

In the model above, every time when a car gets out of the road, we let the car return to the road with the same speed. In this way, the road becomes an enclosure space and the vehicle density can remain consistent so that we can make it an independent variable and adjust its value as we want.

However, the typical CA model is only valid when we study a small part of the road, which can ensure the invariability of vehicle density. In practice, roads are open space and the density of roads changes all the time. Different densities represent different efficiencies of road utilization. Low density means that it is easier for cars to accelerate and pass through the road, which leads to an increase of traffic flow. So now, we will not regard vehicle density as an independent variable that can be easily controlled. In order to be closer to the reality, we want to use both two index (density and traffic flow) to measure the flow of highways.

3 Modified CA Model for Self-driving Cars

3.1 Model Description

Our model is loosely based on the Nagel Schreckenberg model(CA model), and was used to simulate the traffic flow with two types of cars: self-driving cars and cars driven by human beings. Human drivers acted as the simple NS model mentioned in 2.2, but since the real road has not only one lane, we also consider changing lane numbers. Self-driving cars behave according to the rules we set up for them in special purpose.

In the practical operation, we distinguish human drivers and self-driving cars by giving them different sets of characteristics which determine how they move. For example, self-driving cars are more likely to change into a new lane if there is more space on the new lane. Then, their possibility of changing lanes is far more than human drivers. Besides, cooperating self-driving cars can detect the position and speed of each other, which allow them to take more sensible and efficient reaction compared with human drivers.

Our model have following basic assumptions:

- Overlook the complexity of human behavior,which means we do not separate human drivers into more specific groups and set up same rules of moving for all human drivers.
- The probability of car crashes is small enough that is negligible.
- Other assumptions of CA model mentioned in 2.1.2.

The structure of our simulation is simple:

- The simulation starts with an empty road, the cars appear at one end of the road with probability q.
- The speed of car is updated according to the rules that depend on the type of car, the speeds and the positions of neighbor cars, either driven by human or self-driving.
- The position of the car is updated according to the updated speed.

The latter two will be applied to every car on the road and be repeated enough times to avoid uncertainty of randomness. After all the iterations, we measure the number of cars that leave the road, which can be interpreted as traffic flow.

3.2 Speed Changing Rules

Since most of roads have less than 5 lanes in real life, we take road with 5 lanes as an example to explain the different speed changing rules for human drivers and self-driving cars. The rules for roads with less than 5 lanes are easy to deduce from the rules below.

3.2.1 Rules Followed by Human Drivers

For cars in Lane 5(Lane 1):

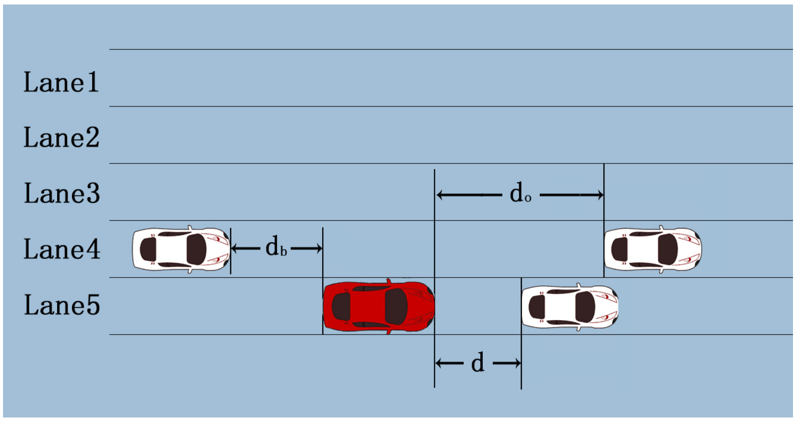

Figure 4: Cars in Lane 5

- Step 1
  If v is less than v_{max}, increase v by 1.

$$L = \min(v + 1, v_{max}) \qquad \text{when} \quad v < v_{max}$$

- Step 2
  If the car has limited space in its own lane and lane 4 can provide it more space, human drivers tend to change lanes with probability p_{c1}. If the next car ahead on lane 4 is within L cells, human driver will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_o > d \quad \text{and} \quad d_b > 0$$
$$v = \min(v+1, d_o, v_{max}) \quad \text{with probability} \quad p_{c1}$$

- Step 3
  If the three conditions in step 2 cannot be satisfied simultaneously, human drivers tend to remain in lane 5. If the next car ahead on lane 5 is within L cells, human driver will reduce the speed sufficiently to avoid a crash. Contemporarily, human driver will reduce the speed by 1 with probability p.

$$\text{when conditions in step 2 are not established}$$
$$v = \begin{cases} \min(v+1, d, v_{max}) & \text{with probability} \quad 1-p \\ \max(v-1, 0) & \text{with probability} \quad p \end{cases}$$

The same rules can be easily applied to cars on lane 1.

For cars in Lane 3(Lane 2,4):

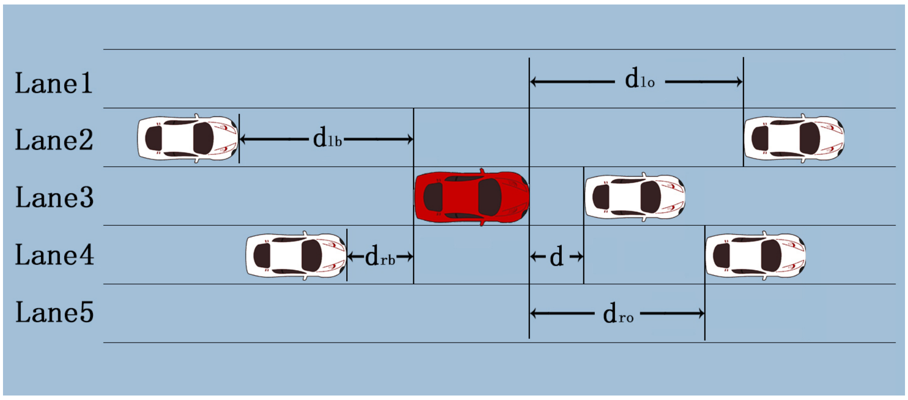

Figure 5: Cars in Lane 3

- Step 1
  If v is less than v_{max}, increase v by 1.

$$L = \min(v+1, v_{max}) \qquad \text{when} \quad v < v_{max}$$

- Step 2

  If the car has limited space in its own lane, and lane 2 and lane 4 both can provide more space for it, human drivers tend to change to lane 2 or lane4 randomly with probability p_{c1}. If the next car ahead on lane 2 or lane 4 is within L cells, human driver will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_{ro} > d, d_{rb} > 0, d_{lo} > d, d_{lb} > 0$$

$$v = \begin{cases} \min(v+1, d_{ro}, v_{max}) & \text{with probability} \quad \frac{p_{c1}}{2} \\ \min(v+1, d_{lo}, v_{max}) & \text{with probability} \quad \frac{p_{c1}}{2} \end{cases}$$

- Step 3

  If the conditions in step 2 cannot be satisfied and the lane 2 can provide the car more space, human drivers tend to change to lane 4 with probability p_{c1}. If the next car ahead on lane 2 is within L cells, human driver will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_{lo} > d, d_{lb} > 0, d_{ro} \leq d$$

$$v = \min(v+1, d_{lo}, v_{max}) \quad \text{with probability} \quad p_{c1}$$

- Step 4

  If the conditions in step 2 and 3 cannot be satisfied simultaneously and the lane 4 can provide the car more space, human drivers tend to change to lane 4 with probability p_{c1}. If the next car ahead on lane 4 is within L cells, human driver will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_{ro} > d, d_{rb} > 0, d_{lo} \leq d$$

$$v = \min(v+1, d_{ro}, v_{max}) \quad \text{with probability} \quad p_{c1}$$

- Step 5

  If the conditions in step 2, 3 and 4 cannot be satisfied simultaneously, human driver tend to remain in lane 3. If the next car ahead on lane 3 is within L cells, human drivers will reduce the speed sufficiently to avoid a crash. Contemporarily, human driver will reduce the speed by 1 with probability p.

$$\text{when conditions in step 2, 3 and 4 are not established}$$

$$v = \begin{cases} \min(v+1, d, v_{max}) & \text{with probability} \quad 1-p \\ \max(v-1, 0) & \text{with probability} \quad p \end{cases}$$

The same rules can be easily applied to cars on lane 2 and lane 4.

3.2.2 Rules Followed by Self-driving Cars

Self-driving cars mainly have these differences from human drivers:

- It is more likely for self-driving cars to change lanes when there is more space in the neighbor lane.
- Self-driving cars will leave a safe distance from the back cars to avoid crashes.
- Human drivers will reduce their speed randomly but self-driving cars will move on steadily.

For cars in Lane 5(Lane 1):

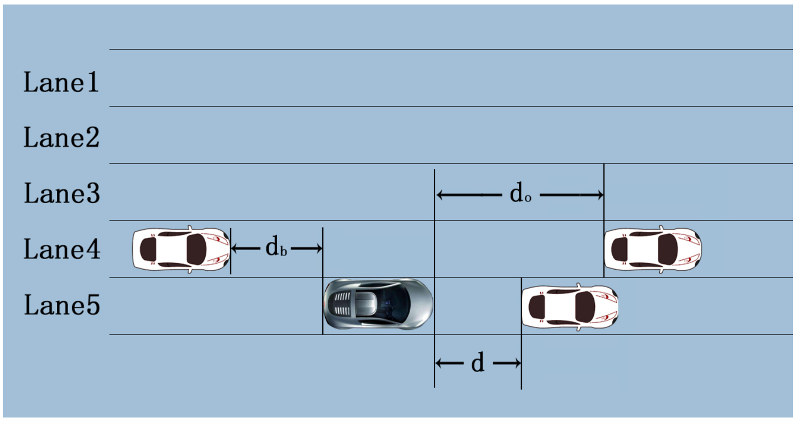

Figure 6: Self-driving Cars in Lane 5

- Step 1
  If v is less than v_{max}, increase v by 1.

$$L = \min(v+1, v_{max}) \qquad \text{when} \quad v < v_{max}$$

- Step 2
  If the car has limited space in its own lane and lane 4 can provide it more space, meanwhile the back car on lane 4 is within safe distance, self-driving cars tend to change lanes with probability p_{c2}. If the next car ahead on lane 4 is within L cells, self-driving cars will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_o > d \quad \text{and} \quad d_b > 1$$
$$v = \min(v+1, d_o, v_{max}) \quad \text{with probability} \quad p_{c2}$$

- Step 3
  If the three conditions in step 2 cannot be satisfied simultaneously, self-driving cars tend to remain in lane 5. If the next car ahead on lane 5 is within L cells, self-driving cars will reduce the speed sufficiently to avoid a crash.

$$\text{when conditions in step 2 are not established}$$
$$v = \min(v + 1, d, v_{max})$$

The same rules can be easily applied to cars on lane 1.

For cars in Lane 3(Lane 2,4):

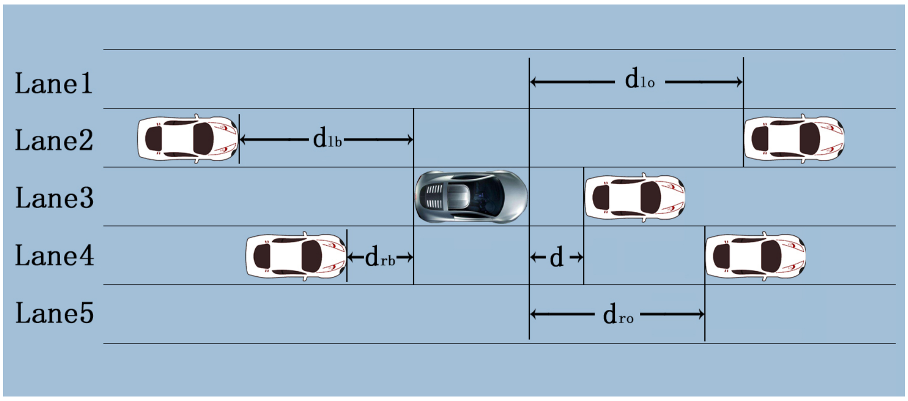

Figure 7: Self-driving Cars in Lane 3

- Step 1
  If v is less than v_{max}, increase v by 1.

$$L = \min(v + 1, v_{max}) \qquad \text{when} \quad v < v_{max}$$

- Step 2
  If the car has limited space in its own lane, and lane 2 and lane 4 both can provide more space for it, meanwhile the back cars on lane 2 and lane 4 are within safe distance, self-driving cars tend to change to lane 2 or lane4 randomly with probability p_{c2}. If the next car ahead on lane 2 or lane 4 is within L cells, self-driving cars will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_{ro} > d, d_{rb} > 1, d_{lo} > d, d_{lb} > 1$$
$$v = \begin{cases} \min(v + 1, d_{ro}, v_{max}) & \text{with probability} \quad \frac{p_{c2}}{2} \\ \min(v + 1, d_{lo}, v_{max}) & \text{with probability} \quad \frac{p_{c2}}{2} \end{cases}$$

- Step 3
  If the conditions in step 2 cannot be satisfied and the lane 2 can provide the car more space, meanwhile the back car on lane 4 is within safe distance, self-driving cars tend to change to lane 4 with probability p_{c2}. If the next car ahead on lane 2 is within L cells, self-driving cars will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_{lo} > d, d_{lb} > 1, d_{ro} \leq d$$
$$v = \min(v + 1, d_{lo}, v_{max}) \quad \text{with probability} \quad p_{c2}$$

- Step 4
  If the conditions in step 2 and 3 cannot be satisfied simultaneously and the lane 4 can provide the car more space, self-driving cars tend to change to lane 4 with probability p_{c2}. If the next car ahead on lane 4 is within L cells, self-driving cars will reduce the speed sufficiently to avoid a crash.

$$\text{when} \quad d < L, d_{ro} > d, d_{rb} > 1, d_{lo} \leq d$$
$$v = \min(v + 1, d_{ro}, v_{max}) \quad \text{with probability} \quad p_{c2}$$

- Step 5
  If the conditions in step 2, 3 and 4 cannot be satisfied simultaneously, self-driving cars tend to remain in lane 3. If the next car ahead on lane 3 is within L cells, self-driving cars will reduce the speed sufficiently to avoid a crash.

$$\text{when conditions in step 2, 3 and 4 are not established}$$
$$v = \min(v + 1, d, v_{max})$$

The same rules can be easily applied to cars on lane 2 and lane 4.

3.3 Algorithm Flow

- Step 1: Assigning Values to Parameters

  Note: p_{c1} and p_{c2} are under satisfied conditions.

- Step 2: Initialization

  First, the road is initialized as a zero matrix of length m and width n. Every car occupies one element of the road matrix. In the iterations, this matrix will save the position of every car. To distinguish self-driving cars and human drivers, we need to give them different values. In our algorithm, we give the self-driving cars "2", human drivers "1", and empty space "0".

  The initialized speed of every car is assigned zero.

Denotation	Definition	Value
q	The percentage of self-driving cars in all cars	0.1 to 0.9 by 0.02
p_{c1}	The probability for human drivers to change lane	0.2
p_{c2}	The probability for self-driving cars to change lanes	1
q_i	The probability for new car's appearance on lane i	1 i=1,2,3,4,5
p	The probability to reduce speed by 1	0.2
n	The number of lanes	5
m	The number of cells on the road	10000
v_{max}	Speed limit	4

Table 1: Variables

- Step 3: Updating the Speed and Position

  The speed of each car is updated according to the rules mentioned in 3.2.1 and 3.1.2. Also, we can know whether the car change lanes and which lane it is changing to according to the same rules.

  The position of each car is updated by moving forward according to the new speed and changing lanes according to the rules mentioned above.

  After repeating step 3 for enough times, we are able to measure the traffic flow by counting the number of cars that get out of the road. Contemporarily, we can measure the density of the road by calculating the ratio of the number of cars on the road to the whole space of the road. (calculated by $m \times n$)

- Step 4: Changing Percentage of Self-driving Cars

  Finally, we change the percentage of self-driving cars from 0.1 to 0.9 by 0.02, then repeat step 2, 3 and 4 to see if there is an increase of traffic flow and a decline of vehicle density.

3.4 Results

3.4.1 How Self-driving Cars Influence Vehicle Density

Since density of cars intuitively show the congestion level of the road, the worse the congestion is, the larger density the road has. Therefore, one of our goals is to find out that whether the vehicle density will decrease when we add self-driving cars to the road. With a constant number of cars we put on the beginning of the road every time step, the vehicle density, calculated by the ratio of the number of cars on the road to $m \times n$, will fluctuate slightly around a fixed value over long time scales.Then we take down the density for different percentages of self-driving cars.

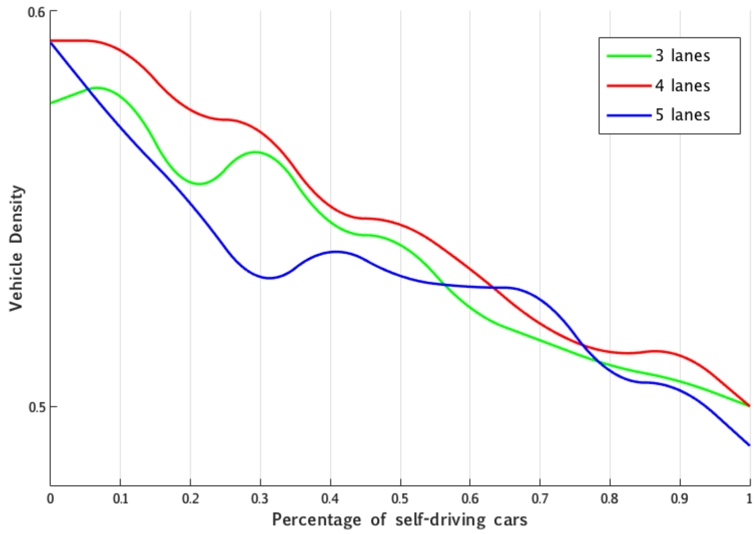

Figure 8: Relation between Vehicle Density and Percentage of Self-driving Cars

Note: Due to the randomness of our simulation, there exists fluctuation of the curve.

But we can clearly notice:

- The density of roads is not strongly correlated to the number of lanes.
- Vehicle density drops from 0.6 to 0.5 noticeably when we increase the percentage of self-driving cars from 0 to 1.
- When we add the percentage of self-driving cars to 100%, which means replacing all human drivers with self-driving cars, theoretically the vehicle density will be minimized.

Consequently, it is reasonable for us to conclude that self-driving cars can obviously enhance the road utilization rate and alleviate the traffic congestion.

3.4.2 How Self-driving Cars Influence Traffic Flow

Another goal of our model is to find out that whether traffic flow will increase when we increase the percentage of self-driving cars. We calculate the traffic flow by counting the number of cars getting out of the road.

- From the curves in Figure 9, we can notice that the traffic flow will increase when the road has more lanes. Since we calculate the traffic flow by counting the number of cars getting out of the road, it is reasonable to receive this result.

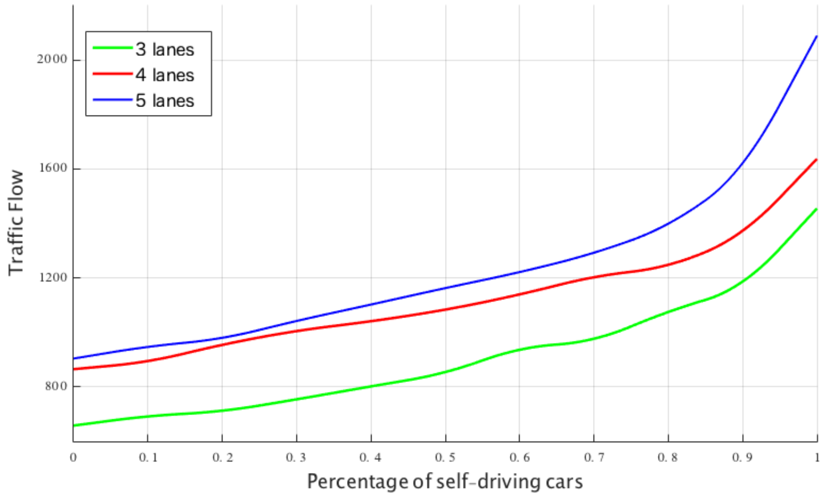

Figure 9: Relation between Traffic Flow and Percentage of Self-driving Cars

- The traffic flow shows an obvious trend of growth when we increase the percentage of self-driving cars, meaning that when we let more self-driving cars into the road, there are more cars passing through the road within a fixed time scale. When the percentage is less than 0.2, the trend of growth is not so obvious, it seems that it takes a large number of self-driving cars to significantly increase the traffic flow. And when the percentage is more than 0.8, the traffic flow will increase faster.
- Generally, it is reasonable for us to conclude that self-driving cars can increase traffic flow, and when the percentage is large enough, the trend of growth will be significant.

4 Applying Our Model to the Provided Data

4.1 Analysis of the Provided Data

Basically, the goal of this paper is to alleviate traffic congestion and increase traffic capacity of highways. Congestions are especially severe on Interstates 5, 90, 405 and State Route 520. And since 8% of the daily traffic volume occurs during peak hours on average, we only consider how to alleviate congestion of these 4 routes during peak hours. Then, based on the data of average daily traffic counts, we find that not all the sections of the highway are congested. Thus the first step is to find out the sections with congestion problems.

As we mentioned in 3.4.1, density of cars intuitively shows the congestion level of the road. We decide to calculate the density of every section of these four routes according to the theory below.

- Let L be the length of the one-lane road, N be the number of cars on the road, it is easy to calculate the average distance between two cars noted as "d" and the vehicle density noted as "ρ" according to the equations below:

$$\begin{cases} \rho = \frac{N}{L} \\ d = \frac{L-N}{N} \end{cases} \tag{1}$$

- Let F be the traffic flow of this one-lane road, meaning the number of cars getting out of the road per time step, then we have:

$$F = \rho \times d = \frac{N}{L} \times \frac{L-N}{N} = 1 - \frac{N}{L} = 1 - \rho \tag{2}$$

First, we calculate the average traffic counts of every section in peak hours, then we separate it into INCR direction and DECR direction according to the number of lanes. Finally, we calculate the flow which means the number of cars getting out of the road per time step and obtain the density of each section according to the equation (2).

In the Figure 10, we highlight the congestion sections in red:

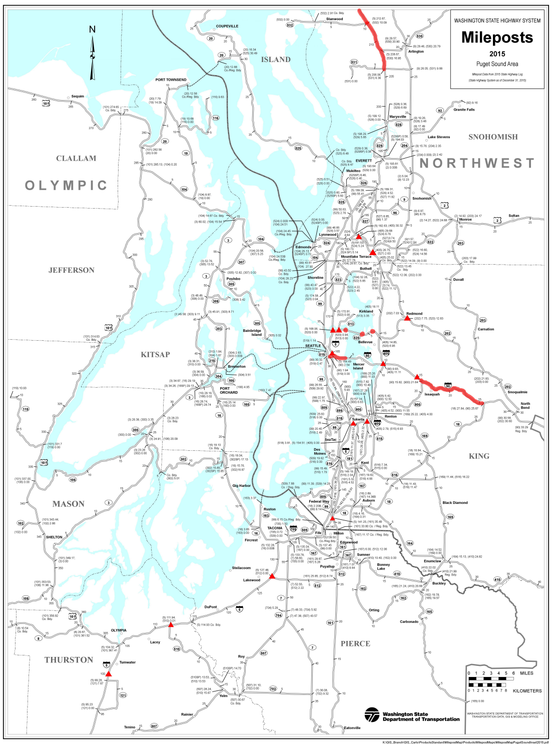

Figure 10: Washington State Highway System

4.2 Fitting the Data

According to the map in Figure 10, we choose 3 typical congested sections to fit the data, and try to find a scientific solution to the congestion problem.

4.2.1 Start of I90 in Downtown

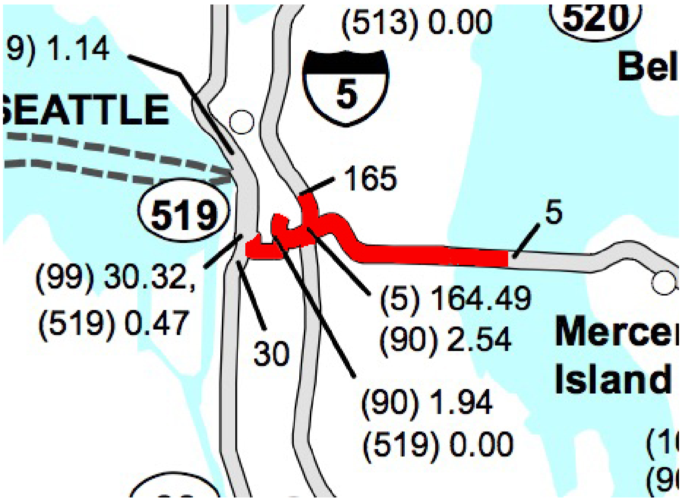

Figure 11: Traffic Condition around the Start of I90 in Downtown Seattle

ρ	Density of the road	0.62
F	Flux per second	0.38
f	Flux during the peak hour	208

Table 2: Density and Flux of the Start of I90 in Downtown Seattle

- The start of I90 is a typical junction of 3 roads, some sections of which have only 2 lanes. When the cars approach to the junction, it is easy to cause congestion. We suppose that when cars are away from the junction, v_{max} can be reached and they can move on smoothly. When the cars enter the junction, they will be clogged and cannot reach v_{max}. In simulation, we set maximum speed of cars in junction smaller than v_{max}, adjust the value of parameters to fit the density and flux of this road.

- Then we change the percentage of self-driving cars, the result shown by the Figure 12 is quite meaningful. When the percentage is less than 90%, density and average velocity change a little, however, after 90%, the density reduces sharply and the velocity increases significantly to $vmax$. We can conclude that 90% is a point where the influence of self-driving cars changes markedly, which means we should add at least 90% self-driving in order to solve the congestion problem of the start of I90.

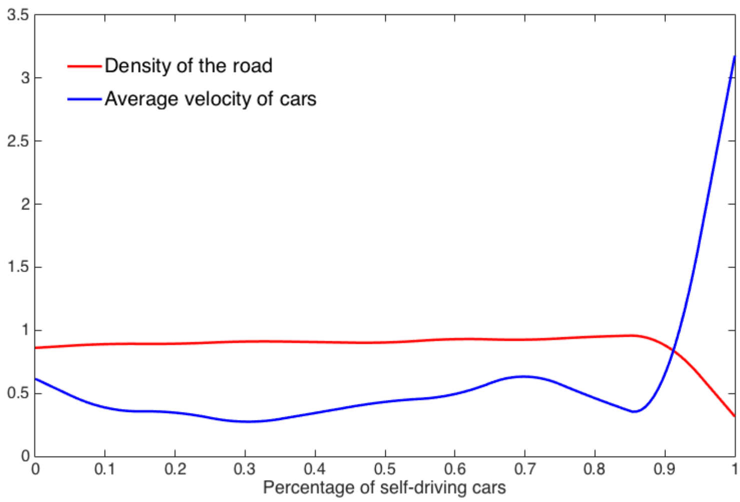

Figure 12: Relation between Density, Speed and Percentage of Self-driving cars

4.2.2 202-212 Miles of Interstate 5

The 202-212 miles of Interstate 5 is an aggregation of 10 sections in data. When the cars approach to the intersections with 530, 531, 532, cars will slow down because of the congestion. In simulation, we set a smaller v_{max} for cars when they enter intersections and adjust the value of parameters to fit the density and the flux of this road.

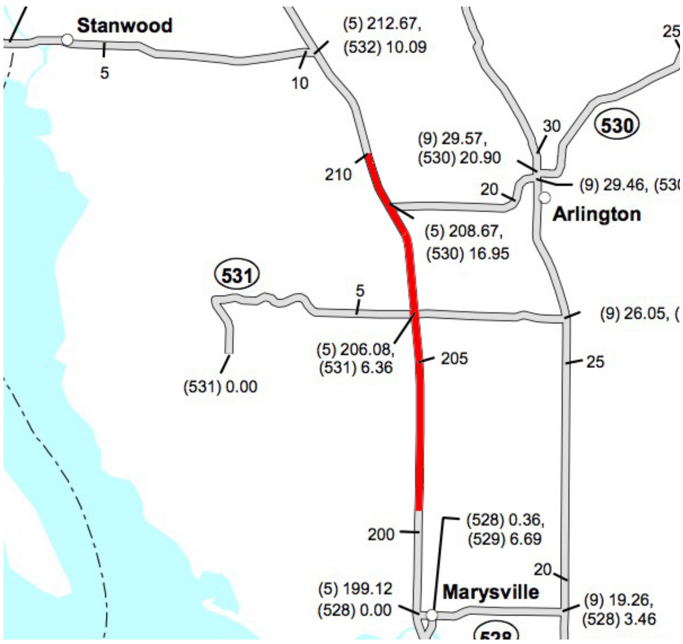

Figure 13: Traffic Condition around 202-212 Miles of Interstate 5

ρ	Density of the road	0.68
F	Flux per second	0.33
f	Flux during the peak hour	1050

Table 3: Density and Flux of 202-212 Miles of Interstate 5

Then we change the percentage of self-driving cars, the result shown by the Figure 14 is similar to Figure 12. That means, 90% is still a turning point of self-driving cars' percentage. When there are more than 90% self-driving cars, the density of the road will drop significantly to make the road unobstructed. Meanwhile cars are able to reach their speed limit again.

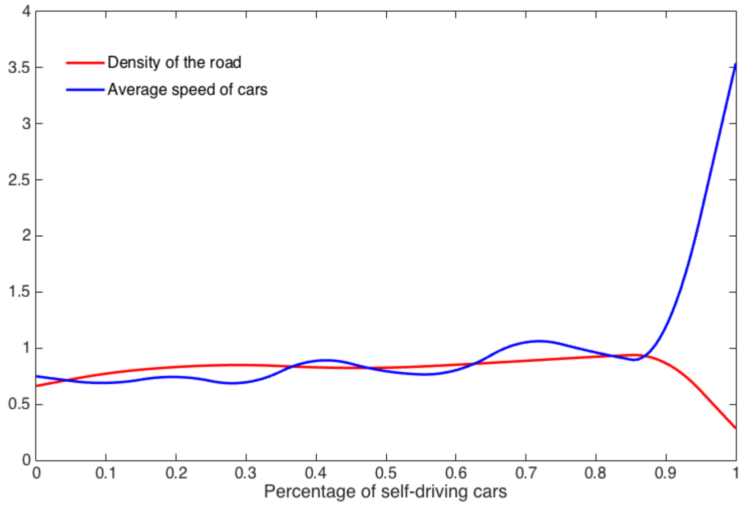

Figure 14: Relation between density,speed and percentage of self-driving cars

In conclusion, to alleviate the congestion of 202-212 miles of Interstate 5, we also need at least 90% self-driving cars.

References

[1] Rillings, J., 1997. Automated highway systems. Scientific American 365, 60−63.

[2] M.E, Lárraga, Del Río J.A., and Alvarez-lcaza L. "Cellular Automata for One-lane Traffic Flow Modeling." *Elsevier*. 20 Mar. 2004. Web. 22 Jan. 2017. <https://www.journals.elsevier.com/transportation-research-part-c-emerging-technologies>.

[3] Li, Qi−Lang, Rui Jiang, and Zhong−Jun Ding. "Effect of Vehicles' Changing Lanes in the BihamMiddletonLevine Traffic Flow Model." *ScienceDirect.com*. World Scientific, 12 Aug. 2014. Web. 22 Jan. 2017. <http://www.sciencedirect.com/>.

Appendices

Input matlab source:

```
flag=1000;
n=5;
m=300;
cars=zeros(5,m);
v=zeros(5,m);
id=zeros(5,m);
vmax_m=4*ones(1,m);
vmax_m(250-8:250)=2;
vmax_m(250:250+8)=1;

p=0.3;
d2=1;
p_c1=0.2;
p_c2=0.9;
q1=0.8;
q2=0.8;
q3=0.8;
q4=0.8;
q5=0.8;
sumv=0;
ncar=0;
rho1=0;rho2=0;rho3=0;rho4=0;rho5=0;
rho=0;
va=0;
J1=0;

cars_out=0;
cars_in=0;
cars_on1=0;cars_on2=0;cars_on3=0;cars_on4=0;cars_on5=0;
```

```
cars_on=0;



while flag>0

        temp_cars=cars;
        temp_v=v;
        temp_id=id;
        cars=zeros(5,m);
        v=zeros(5,m);
        id=zeros(5,m);
        cars_out=cars_out+sum(temp_cars(:,m));
                         % alway out when car in in the last cell
        for ii=m-1:-1:1  % update from back to front
              vmax=vmax_m(ii);
            %% update lane 5
            if temp_cars(5,ii)==1
                    d=0;
                    if cars(5,ii+1:m)==0
                        d=vmax;
                    else
                        for jj=ii+1:m
                            if cars(5,jj)==0
                                d=d+1;
                            else
                                break;
                            end
                        end
                    end

                    l=min(temp_v(5,ii)+1,vmax);
                    d_o=0;
                    d_b=0;
                    if d<l
                        if cars(4,ii+1:m)==0
                            d_o=vmax;
                        else
                            for jj=ii+1:m
                                if cars(4,jj)==0
                                    d_o=d_o+1;
                                else
                                    break;
                                end
                            end
                        end

                        if ii<=vmax
                            d_b=0;
                        else
                            for jj=ii:-1:1
                                if temp_cars(4,jj)==0
                                    d_b=d_b+1;
                                else
```

```
                                    break
                                end
                            end
                        end
                    end
                 if temp_id(5,ii)==1
                       if rand(1)<p
                       l=max(l-1,0);
                       end
                    if d<l && d_o>d && d_b>=1 && rand(1)<p_c1
                        temp_vv=min([l,vmax,d_o]);
                        if ii+temp_vv<=m
                            cars(4,ii+temp_vv)=1;
                            v(4,ii+temp_vv)=temp_vv;
                            id(4,ii+temp_vv)=temp_id(5,ii);
                        else
                            cars_out=cars_out+1;
                        end
                    else
                        temp_vv=min([l,vmax,d]);
                        if ii+temp_vv<=m
                            cars(5,ii+temp_vv)=1;
                            v(5,ii+temp_vv)=temp_vv;
                            id(5,ii+temp_vv)=temp_id(5,ii);
                        else
                            cars_out=cars_out+1;
                        end
                   end
                    else %id(5,ii)==2
                   if d<l && d_o>d && d_b>=d2 && rand(1)<p_c2
                       temp_vv=min([temp_v(5,ii)+1,vmax,d_o]);
                        if ii+temp_vv<=m
                            cars(4,ii+temp_vv)=1;
                            v(4,ii+temp_vv)=temp_vv;
                            id(4,ii+temp_vv)=temp_id(5,ii);
                        else
                            cars_out=cars_out+1;
                        end
                        else
                        temp_vv=min([temp_v(5,ii)+1,vmax,d]);
                        if ii+temp_vv<=m
                            cars(5,ii+temp_vv)=1;
                            v(5,ii+temp_vv)=temp_vv;
                            id(5,ii+temp_vv)=temp_id(5,ii);
                        else
                            cars_out=cars_out+1;
                        end

                   end
                 end
            end


             %% update lane 4
            if temp_cars(4,ii)==1
```

```
d=0;
if cars(4,ii+1:m)==0
    d=vmax;
else
    for jj=ii+1:m
        if cars(4,jj)==0
            d=d+1;
        else
            break;
        end
    end
end

d_r=0;
d_r_b=0;

if cars(5,ii+1:m)==0
    d_r=vmax;
else
    for jj=ii+1:m
        if cars(5,jj)==0
            d_r=d_r+1;
        else
            break
        end
    end
end

if ii<=vmax
    d_r_b=0;
else
    for jj=ii:-1:1
        if temp_cars(5,jj)==0
            d_r_b=d_r_b+1;
        else
            break
        end
    end
end

l=min(temp_v(4,ii)+1,vmax);
d_o=0;
d_b=0;
if d<l
    if cars(3,ii+1:m)==0
        d_o=vmax;
    else
        for jj=ii+1:m
            if cars(3,jj)==0
                d_o=d_o+1;
            else
                break;
            end
        end
    end
```

```
                    if ii<=vmax
                        d_b=0;
                    else
                        for jj=ii:-1:1
                            if temp_cars==0
                                d_b=d_b+1;
                            else
                                break
                            end
                        end
                    end
                end
            if temp_id(4,ii)==1
                if rand(1)<p
                   l=max(l-1,0);
                end
                if  d<l && d_r>d && d_r_b>=1 && d_o>d && d_b>=1 && rand(1)<p_c1
                    if rand(1)<0.5
                    temp_vv=min([l,vmax,d_r]);
                    if ii+temp_vv<=m
                        cars(5,ii+temp_vv)=1;
                        v(5,ii+temp_vv)=temp_vv;
                        id(5,ii+temp_vv)=temp_id(4,ii);
                    else
                        cars_out=cars_out+1;
                    end
                    else
                        temp_vv=min([l,vmax,d_o]);
                    if ii+temp_vv<=m
                        cars(3,ii+temp_vv)=1;
                        v(3,ii+temp_vv)=temp_vv;
                        id(3,ii+temp_vv)=temp_id(4,ii);
                    else
                        cars_out=cars_out+1;
                    end
                    end
                elseif d<l && d_o>d && d_b>=1 && d_r<=d && rand(1)<p_c1
                    temp_vv=min([l,vmax,d_o]);
                    if ii+temp_vv<=m
                        cars(3,ii+temp_vv)=1;
                        v(3,ii+temp_vv)=temp_vv;
                        id(3,ii+temp_vv)=temp_id(4,ii);
                    else
                        cars_out=cars_out+1;
                    end
                elseif d<l && d_r>d && d_r_b>=1 && d_o<=d && rand(1)<p_c1
                    temp_vv=min([l,vmax,d_r]);
                    if ii+temp_vv<=m
                        cars(5,ii+temp_vv)=1;
                        v(5,ii+temp_vv)=temp_vv;
                        id(5,ii+temp_vv)=temp_id(4,ii);
                    else
                        cars_out=cars_out+1;
                    end
```

```
        else
            temp_vv=min([1,vmax,d]);
            if ii+temp_vv<=m
                cars(4,ii+temp_vv)=1;
                v(4,ii+temp_vv)=temp_vv;
                id(4,ii+temp_vv)=temp_id(4,ii);
            else
                cars_out=cars_out+1;
            end
        end

    else %temp_id==2
      if  d<l && d_r>d && d_r_b>=d2 && d_o>d && d_b>=d2 && rand(1)<p_c2
            if rand(1)<0.5
            temp_vv=min([temp_v(4,ii)+1,vmax,d_r]);
            if ii+temp_vv<=m
                cars(5,ii+temp_vv)=1;
                v(5,ii+temp_vv)=temp_vv;
                id(5,ii+temp_vv)=temp_id(4,ii);
            else
                cars_out=cars_out+1;
            end
            else
                temp_vv=min([temp_v(4,ii)+1,vmax,d_o]);
            if ii+temp_vv<=m
                cars(3,ii+temp_vv)=1;
                v(3,ii+temp_vv)=temp_vv;
                id(3,ii+temp_vv)=temp_id(4,ii);
            else
                cars_out=cars_out+1;
            end
            end
        elseif d<l && d_o>d && d_b>=d2 && d_r<=d && rand(1)<p_c2
            temp_vv=min([temp_v(4,ii)+1,vmax,d_o]);
            if ii+temp_vv<=m
                cars(3,ii+temp_vv)=1;
                v(3,ii+temp_vv)=temp_vv;
                id(3,ii+temp_vv)=temp_id(4,ii);
            else
                cars_out=cars_out+1;
            end
        elseif d<l && d_r>d && d_r_b>=d2 && d_o<=d && rand(1)<p_c2
            temp_vv=min([temp_v(4,ii)+1,vmax,d_r]);
            if ii+temp_vv<=m
                cars(5,ii+temp_vv)=1;
                v(5,ii+temp_vv)=temp_vv;
                id(5,ii+temp_vv)=temp_id(4,ii);
            else
                cars_out=cars_out+1;
            end
        else
            temp_vv=min([temp_v(4,ii)+1,vmax,d]);
            if ii+temp_vv<=m
                cars(4,ii+temp_vv)=1;
                v(4,ii+temp_vv)=temp_vv;
```

```
                    id(4,ii+temp_vv)=temp_id(4,ii);
                else
                    cars_out=cars_out+1;
                end
           end
          end

    end
    %% update lane 3
    if temp_cars(3,ii)==1
            d=0;
            if cars(3,ii+1:m)==0
                d=vmax;
            else
                for jj=ii+1:m
                    if cars(3,jj)==0
                        d=d+1;
                    else
                        break;
                    end
                end
            end

            d_r=0;
            d_r_b=0;

            if cars(4,ii+1:m)==0
                d_r=vmax;
            else
                for jj=ii+1:m
                    if cars(4,jj)==0
                        d_r=d_r+1;
                    else
                        break
                    end
                end
            end

            if ii<=vmax
                d_r_b=0;
            else
                for jj=ii:-1:1
                    if temp_cars(4,jj)==0
                        d_r_b=d_r_b+1;
                    else
                        break
                    end
                end
            end

            l=min(temp_v(3,ii)+1,vmax);
            d_o=0;
            d_b=0;
            if d<l
                if cars(2,ii+1:m)==0
```

```
                d_o=vmax;
            else
                for jj=ii+1:m
                    if cars(2,jj)==0
                        d_o=d_o+1;
                    else
                        break;
                    end
                end
            end

            if ii<=vmax
                d_b=0;
            else
                for jj=ii:-1:1
                    if temp_cars(2,jj)==0
                        d_b=d_b+1;
                    else
                        break
                    end
                end
            end
        end


    if temp_id(3,ii)==1
        if rand(1)<p
           l=max(l-1,0);
        end
        if  d<l && d_r>d && d_r_b>=1 && d_o>d && d_b>=1 && rand(1)<p_c1
            if rand(1)<0.5
            temp_vv=min([l,vmax,d_r]);
            if ii+temp_vv<=m
                cars(4,ii+temp_vv)=1;
                v(4,ii+temp_vv)=temp_vv;
                id(4,ii+temp_vv)=temp_id(3,ii);
            else
                cars_out=cars_out+1;
            end
            else
                temp_vv=min([l,vmax,d_o]);
            if ii+temp_vv<=m
                cars(2,ii+temp_vv)=1;
                v(2,ii+temp_vv)=temp_vv;
                id(2,ii+temp_vv)=temp_id(3,ii);
            else
                cars_out=cars_out+1;
            end
            end
        elseif d<l && d_o>d && d_b>=1 && d_r<=d && rand(1)<p_c1
            temp_vv=min([l,vmax,d_o]);
            if ii+temp_vv<=m
                cars(2,ii+temp_vv)=1;
                v(2,ii+temp_vv)=temp_vv;
                id(2,ii+temp_vv)=temp_id(3,ii);
```

```
                else
                    cars_out=cars_out+1;
                end
            elseif d<l && d_r>d && d_r_b>=1 && d_o<=d && rand(1)<p_c1
                temp_vv=min([l,vmax,d_r]);
                if ii+temp_vv<=m
                    cars(4,ii+temp_vv)=1;
                    v(4,ii+temp_vv)=temp_vv;
                    id(4,ii+temp_vv)=temp_id(3,ii);
                else
                    cars_out=cars_out+1;
                end
            else
                temp_vv=min([l,vmax,d]);
                if ii+temp_vv<=m
                    cars(3,ii+temp_vv)=1;
                    v(3,ii+temp_vv)=temp_vv;
                    id(3,ii+temp_vv)=temp_id(3,ii);
                else
                    cars_out=cars_out+1;
                end
            end

        else %temp_id==2
          if  d<l && d_r>d && d_r_b>=d2 && d_o>d && d_b>=d2 && rand(1)<p_c2
                if rand(1)<0.5
                temp_vv=min([temp_v(3,ii)+1,vmax,d_r]);
                if ii+temp_vv<=m
                    cars(4,ii+temp_vv)=1;
                    v(4,ii+temp_vv)=temp_vv;
                    id(4,ii+temp_vv)=temp_id(3,ii);
                else
                    cars_out=cars_out+1;
                end
                else
                    temp_vv=min([temp_v(3,ii)+1,vmax,d_o]);
                if ii+temp_vv<=m
                    cars(2,ii+temp_vv)=1;
                    v(2,ii+temp_vv)=temp_vv;
                    id(2,ii+temp_vv)=temp_id(3,ii);
                else
                    cars_out=cars_out+1;
                end
                end
            elseif d<l && d_o>d && d_b>=d2 && d_r<=d && rand(1)<p_c2
                temp_vv=min([temp_v(3,ii)+1,vmax,d_o]);
                if ii+temp_vv<=m
                    cars(2,ii+temp_vv)=1;
                    v(2,ii+temp_vv)=temp_vv;
                    id(2,ii+temp_vv)=temp_id(3,ii);
                else
                    cars_out=cars_out+1;
                end
            elseif d<l && d_r>d && d_r_b>=d2 && d_o<=d && rand(1)<p_c2
                temp_vv=min([temp_v(3,ii)+1,vmax,d_r]);
```

```
                    if ii+temp_vv<=m
                        cars(4,ii+temp_vv)=1;
                        v(4,ii+temp_vv)=temp_vv;
                        id(4,ii+temp_id(3,ii);
                    else
                        cars_out=cars_out+1;
                    end
                else
                    temp_vv=min([temp_v(3,ii)+1,vmax,d]);
                    if ii+temp_vv<=m
                        cars(3,ii+temp_vv)=1;
                        v(3,ii+temp_vv)=temp_vv;
                        id(3,ii+temp_vv)=temp_id(3,ii);
                    else
                        cars_out=cars_out+1;
                    end
               end
           end

        end
        %% update lane 2
        if temp_cars(2,ii)==1
                d=0;
                if cars(2,ii+1:m)==0
                    d=vmax;
                else
                    for jj=ii+1:m
                        if cars(2,jj)==0
                            d=d+1;
                        else
                            break;
                        end
                    end
                end

                d_r=0;
                d_r_b=0;

                if cars(3,ii+1:m)==0
                    d_r=vmax;
                else
                    for jj=ii+1:m
                        if cars(3,jj)==0
                            d_r=d_r+1;
                        else
                            break
                        end
                    end
                end

                if ii<=vmax
                    d_r_b=0;
                else
                    for jj=ii:-1:1
                        if temp_cars(3,jj)==0
```

```
                d_r_b=d_r_b+1;
            else
                break
            end
        end
    end

    l=min(temp_v(2,ii)+1,vmax);
    d_o=0;
    d_b=0;
    if d<l
        if cars(1,ii+1:m)==0
            d_o=vmax;
        else
            for jj=ii+1:m
                if cars(1,jj)==0
                    d_o=d_o+1;
                else
                    break;
                end
            end
        end

        if ii<=vmax
            d_b=0;
        else
            for jj=ii:-1:1
                if temp_cars(1,jj)==0
                    d_b=d_b+1;
                else
                    break
                end
            end
        end
    end

    if temp_id(2,ii)==1
    if rand(1)<p
       l=max(l-1,0);
    end
    if  d<l  && d_r>d  && d_r_b>=1  && d_o>d  && d_b>=1  && rand(1)<p_c1
        if rand(1)<0.5
        temp_vv=min([l,vmax,d_r]);
        if ii+temp_vv<=m
            cars(3,ii+temp_vv)=1;
            v(3,ii+temp_vv)=temp_vv;
            id(3,ii+temp_vv)=temp_id(3,ii);
        else
            cars_out=cars_out+1;
        end
        else
            temp_vv=min([l,vmax,d_o]);
        if ii+temp_vv<=m
            cars(1,ii+temp_vv)=1;
            v(1,ii+temp_vv)=temp_vv;
```

```
                id(1,ii+temp_vv)=temp_id(2,ii);
            else
                cars_out=cars_out+1;
            end
            end
        elseif d<l && d_o>d && d_b>=1 && d_r<=d && rand(1)<p_c1
            temp_vv=min([l,vmax,d_o]);
            if ii+temp_vv<=m
                cars(1,ii+temp_vv)=1;
                v(1,ii+temp_vv)=temp_vv;
                id(1,ii+temp_vv)=temp_id(1,ii);
            else
                cars_out=cars_out+1;
            end
        elseif d<l && d_r>d && d_r_b>=1 && d_o<=d && rand(1)<p_c1
            temp_vv=min([l,vmax,d_r]);
            if ii+temp_vv<=m
                cars(3,ii+temp_vv)=1;
                v(3,ii+temp_vv)=temp_vv;
                id(3,ii+temp_vv)=temp_id(2,ii);
            else
                cars_out=cars_out+1;
            end
        else
            temp_vv=min([l,vmax,d]);
            if ii+temp_vv<=m
                cars(2,ii+temp_vv)=1;
                v(2,ii+temp_vv)=temp_vv;
                id(2,ii+temp_vv)=temp_id(2,ii);
            else
                cars_out=cars_out+1;
            end
        end

    else %temp_id==2
      if  d<l && d_r>d && d_r_b>=d2 && d_o>d && d_b>=d2 && rand(1)<p_c2
            if rand(1)<0.5
            temp_vv=min([temp_v(2,ii)+1,vmax,d_r]);
            if ii+temp_vv<=m
                cars(3,ii+temp_vv)=1;
                v(3,ii+temp_vv)=temp_vv;
                id(3,ii+temp_vv)=temp_id(2,ii);
            else
                cars_out=cars_out+1;
            end
            else
                temp_vv=min([temp_v(2,ii)+1,vmax,d_o]);
            if ii+temp_vv<=m
                cars(1,ii+temp_vv)=1;
                v(1,ii+temp_vv)=temp_vv;
                id(1,ii+temp_vv)=temp_id(2,ii);
            else
                cars_out=cars_out+1;
            end
            end
```

```
                elseif d<l && d_o>d && d_b>=d2 && d_r<=d && rand(1)<p_c2
                    temp_vv=min([temp_v(2,ii)+1,vmax,d_o]);
                    if ii+temp_vv<=m
                        cars(1,ii+temp_vv)=1;
                        v(1,ii+temp_vv)=temp_vv;
                        id(1,ii+temp_vv)=temp_id(2,ii);
                    else
                        cars_out=cars_out+1;
                    end
                elseif d<l && d_r>d && d_r_b>=d2 && d_o<=d && rand(1)<p_c2
                    temp_vv=min([temp_v(2,ii)+1,vmax,d_r]);
                    if ii+temp_vv<=m
                        cars(3,ii+temp_vv)=1;
                        v(3,ii+temp_vv)=temp_vv;
                        id(3,ii+temp_vv)=temp_id(2,ii);
                    else
                        cars_out=cars_out+1;
                    end
                else
                    temp_vv=min([temp_v(2,ii)+1,vmax,d]);
                    if ii+temp_vv<=m
                        cars(2,ii+temp_vv)=1;
                        v(2,ii+temp_vv)=temp_vv;
                        id(2,ii+temp_vv)=temp_id(2,ii);
                    else
                        cars_out=cars_out+1;
                    end
               end
                end

        end
        %% updata lane 1
        if temp_cars(1,ii)==1
            d=0;
                if cars(1,ii+1:m)==0
                    d=vmax;
                else
                    for jj=ii+1:m
                        if cars(1,jj)==0
                            d=d+1;
                        else
                            break;
                        end
                    end
                end

                l=min(temp_v(1,ii)+1,vmax);
                d_r=0;
                d_r_b=0;
                if d<l
                 if cars(2,ii+1:m)==0
                    d_r=vmax;
                else
                    for jj=ii+1:m
                        if cars(2,jj)==0
```

```
                d_r=d_r+1;
            else
                break
            end
        end
    end

    if ii<=vmax
        d_r_b=0;
    else
        for jj=ii:-1:1
            if temp_cars(2,jj)==0
                d_r_b=d_r_b+1;
            else
                break
            end
        end
    end
    end


  if temp_id(1,ii)==1

    if rand(1)<p
       l=max(l-1,0);
    end
    if d<l && d_r>d && d_r_b>=1 && rand(1)<p_c1
        temp_vv=min([l,vmax,d_r]);
        if ii+temp_vv<=m
            cars(2,ii+temp_vv)=1;
            v(2,ii+temp_vv)=temp_vv;
            id(2,ii+temp_vv)=temp_id(1,ii);
        else
            cars_out=cars_out+1;
        end
    else
        temp_vv=min([l,vmax,d]);
        if ii+temp_vv<=m
            cars(1,ii+temp_vv)=1;
            v(1,ii+temp_vv)=temp_vv;
            id(1,ii+temp_vv)=temp_id(1,ii);
        else
            cars_out=cars_out+1;
        end
    end
    else %id(5,ii)==2
  if d<l && d_r>d && d_r_b>=d2 && rand(1)<p_c2
       temp_vv=min([temp_v(1,ii)+1,vmax,d_r]);
        if ii+temp_vv<=m
            cars(2,ii+temp_vv)=1;
            v(2,ii+temp_vv)=temp_vv;
            id(2,ii+temp_vv)=temp_id(1,ii);
        else
            cars_out=cars_out+1;
        end
```

```
                        else
                        temp_vv=min([temp_v(1,ii)+1,vmax,d]);
                        if ii+temp_vv<=m
                            cars(1,ii+temp_vv)=1;
                            v(1,ii+temp_vv)=temp_vv;
                            id(1,ii+temp_vv)=temp_id(1,ii);
                        else
                            cars_out=cars_out+1;
                        end
                    end
                    end

            end

        end

        %% new car coming
        if rand(1)<q1
            if cars(1,1)==0
                cars(1, 1)=1;
                v(1,1)=randi(vmax,1);
                if rand(1)>q        %1 is human driver,2 is self-driving cars
                id(1,1)=1;
                else
                id(1,1)=2;
                end
                cars_in=cars_in+1;
            end
        end
        if rand(1)<q2
            if cars(2,1)==0
                cars(2, 1)=1;
                v(2,1)=randi(vmax,1);
                if rand(1)>q
                id(2,1)=1;
                else
                id(2,1)=2;
                end
                cars_in=cars_in+1;
            end
        end
        if rand(1)<q3
            if cars(3,1)==0
                cars(3, 1)=1;
                v(3,1)=randi(vmax,1);
                 if rand(1)>q
                 id(3,1)=1;
                 else
                 id(3,1)=2;
                 end
                cars_in=cars_in+1;
            end
        end
       if rand(1)<q4
            if cars(4,1)==0
```

```
                cars(4, 1)=1;
                v(4,1)=randi(vmax,1);
                if rand(1)>q
                id(4,1)=1;
                else
                id(4,1)=2;
                end
                cars_in=cars_in+1;
            end
       end
        if rand(1)<q5
            if  cars(5,1)==0
                cars(5, 1)=1;
                v(5,1)=randi(vmax,1);
                if rand(1)>q
                id(5,1)=1;
                else
                id(5,1)=2;
                end
                cars_in=cars_in+1;
            end
        end

        cars_on=sum(cars(:));
        sumv=sum(v(:));
        cars_on5=sum(cars(5,:));
        cars_on4=sum(cars(4,:));
        cars_on3=sum(cars(3,:));
        cars_on2=sum(cars(2,:));
        cars_on1=sum(cars(1,:));
        rho1=cars_on1/m;
        rho2=cars_on2/m;
        rho3=cars_on3/m;
        rho4=cars_on4/m;
        rho5=cars_on5/m;
        rho=(rho1+rho2+rho3+rho4+rho5)/5;
        J1=sumv/cars_on;
        flag=flag-1;

end
```