A simulation on traffic signal control

Xiangyu Kong GT# 903322862
Mingyi Liu GT# 903332202
Peiyuan Sun GT# 903319579

1. Introduction

For traffic simulation, there exists many vehicular traffic models on this topic like the traditional Biham-Middleton-Levine Traffic Model model(BML)[1] and the Nagel–Schreckenberg model (NS)[2]. From the above basic models, many researchers extent their study on the optimization of the traffic lights using the language of cellular automata (CA). The goal of our project is to simulate the cars stream in a one direction road with several traffic lights and the main focus lies on the traffic condition (i.e. whether it jams or not) under different circumstances, like the number of traffic lights, the velocity of cars or speed limits, etc., especially the duration of the traffic light time. In addition, we consider the traffic lights with two special conditions those are synchronously and asynchronously working (will be discussed later). Finally, we analyze the global traffic light control strategies by recording some meaningful data like average number of cars in the road, average speed of cars, the number of cars exiting the road at the end of simulation time, etc.

Generally speaking, what we have done in this project gives us a more comprehensive understanding about the traffic condition and the result of our project can be used to determine some parameters like the number of traffic lights in a specific road and also can be used to predict the road capability for vehicles.

2. Simulation Approach

The project consisted of two main parts. First was the simulation of the traffic flow, which was established on the concept of cellular automaton (CA). The implementation of it was based on C programming language. Second was the visualization of the traffic condition, which relied on the Matlab‘s imagesc and videowriter function. The whole block of road analysis was also based on the matlab’s visualization function. We also calculated the flow of the traffic and the average speed or the traffic flow.

For the first part, our grid cell size would be 100*6 (road length * number of lanes), square shaped, with the capacity of one single car in each cell. With the goal of simulating the traffic flow in a road, we set three different type of the cars:

a. private car: high speed; b. bus: medium speed; c. track: low speed.

The C program consisted mainly five parts.

- *globles.h* : store the globle parameters
- *main.c* : do the whole simulation
- *matrix.c (matrix.h)* : road settings
- *gap.c (gap.h)* : calculation of the gap between car and its front car/ the front light
- *trafficlight.c (trafficlight.h)* : traffic light setting

The output of the C program is:

- *road_condition.txt*: store the road condition results of all time steps
- *roadgapcar.txt*: store the gap car matrix result (numbers of the grid between every two cars)
- *roadgaplight.txt*: store the gap_light matrix result (numbers of the grid between the car and the closest traffic light)
- *velocity_car.txt*: store the velocity matrix result (value of the speed of every car, the car location is same as the position in the position grid)

For the visualization part, the matlab function did the following staff:

- *averge_cal.m*: function to calculate the number of cars and the average velocity on road
- *dove_film.m*: function to write the road_condition.txt to the video for visualization
- *plot_traffic_flow.m*: function to plot the traffic flow all the time
- *plot_average_speed_per_car.m*: function to plot the average speed per car
- *running.m*: function to get all the function together to get all the results

The output of the matlab program is:

- The whole video of the road condition
- The line graph of the traffic flow
- The line graph of the average car speed on road
- The mean traffic flow after 40 time steps and the total average car speed

3. Simulation Process

The simulation process consisted mainly three parts:

3.1 Initialization:

Before the main simulation part, we initialized the roads' condition with the lanes and the distance of the roads we expected. Then, we determined the location of the traffic light based on the number of the light we want. The strategy we used here was to let the distance between each close traffic light is the same. Later, we standardized the change of the traffic light. There were two primary rules to set it:

a) synchronized traffic light: all the traffic light changed at same time
b) asynchronized traffic light: all the traffic light changed separately, for example, the neighbored traffic light changes conversely at the same time step

3.2 Simulation:

Follow the rule of the cellular automation model, each time step the whole road condition updated once. The road matrix here stored the car location and the type of the car. The gap_car matrix here represented the gap between the car and its front car. When there was no front car or the car reached the end of the road, the gap car was set to be a very long distance, e.g., 1000. The gap_light matrix here was to document the gap between the car and the closest traffic light in the front. If the car reached one of the location of the traffic light, the gap_light would be zero. We calculated the gap between the car and the traffic light and its front car each time step. Then we determined the velocity of the car by the following rules:

a) If the gap_light of the car was zero. That was when the car stopped at the front of the traffic light, the car behavior will be determined by the traffic light condition, red stop, green restart. The restarted velocity of the car will be determined by the type of the car and a constant global parameter: the restarted speed before the light.
b) If the gap_light was not zero while its velocity_car equaled to zero. In other words, the car was waiting in line. The car will also restart. The restarted speed of the car will be decided by the minimum of the gap_light and new gap_car minus one. The minus one action is the avoid the car collision.
c) In other situation, the car speed will be determined by the gap_light, gap_car minus one, the previous velocity of the car and a restricting on the car speed. The restriction on the car speed is a constant number and it is a speed limit on the car when there is no car and no light in close front.

After making sure the velocities of the cars, the road condition will be updated. The updating of the road was according to the velocity of each car. The order of the updating of road looped from the front car to the later car. If the car in the position (i, j) has a velocity m, then it will move to the position (i, j+m). When j+m exceeded the distance minus one, the car will move out of the block and disappear.

The total simulation time step is 200 in our research. As it gets stable after 200 time steps in most situation.

3.3 Output results and video

The simulation results were outputted as the txt form. We mainly focused on the traffic changes with the time. So we use the dove_film in Matlab to visualize the road condition. We also plotted the traffic flow and the average speed of the car's in the lanes.

4. Simulation Results

For better analysis on the influence of the traffic light control, we set different values of the critical parameters in our model and get the following simulation results. Here, our focus was mainly on the arrival rate, the time interval, the synchronized attributes of the traffic lights and the traffic flow numbers.

Table 4.1 the initial simulation condition

Parameters	Values
Lanes of road	6
Distance of the road	100
Type of car	3 (1, 2, 3 in cells)
Initial car speed	2 + (car type value)
Car restart speed	2 + (car type value)
Velocity upper bound	6
Velocity lower bound	0
Number of light	5

Traffic light time interval	10
Car generation rate	Poisson distribution per second

And we denote the following items:

a) Traffic Flow: The total number of car at each time step
b) Average Speed: The average speed of cars on roads
c) Car density: The density of the cars on roads

4.1 Analysis of the traffic flow and mean car speed

First, we considered the most primary road condition, when the lights on road were all green all the time. This situation was equal to the condition that there is no light on the road. Then we focused on the impact from the synchronized and asynchronized condition of the traffic light. The results are shown below:

4.1.1 All road green

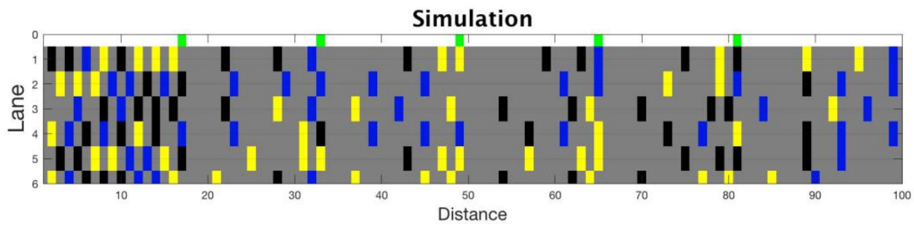

Fig 4.1.1(a) All green road condition screenshot

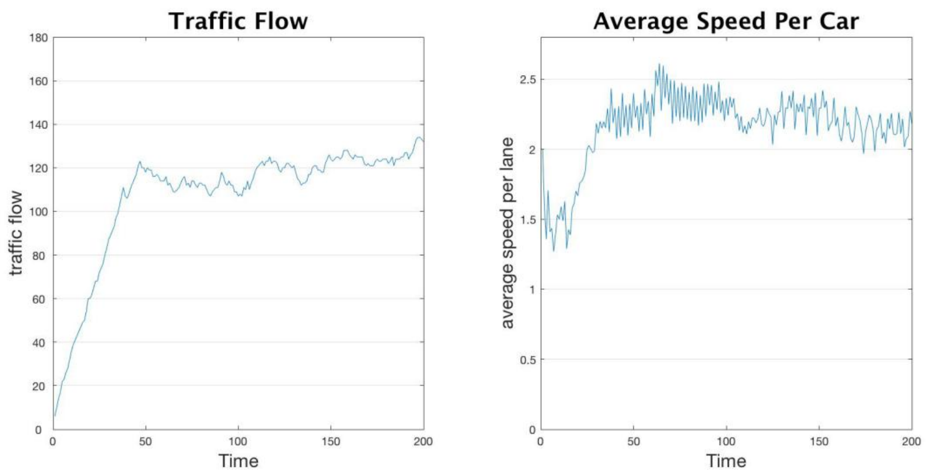

Fig 4.1.1(b) Traffic flow and mean car speed (All green)

From Fig 4.1.1(a) and Fig 4.1.1(b), we could see that different colors on road showed three different types of the car. The traffic flow is denser in the front then the back part. It was easy to understand as different car had different speed, so the distance between one lane cars would get larger as the time went by. The stagnate in the begin of the road is mainly from the reasons of the gap car.

The calculation of the average traffic flow (after 40 timestep) was 119, and the average car speed per time was 2.1603. The sum of the number of cars which moved out of scope is 433, which showed a good flux of the roads.

4.1.2 Synchronized traffic light

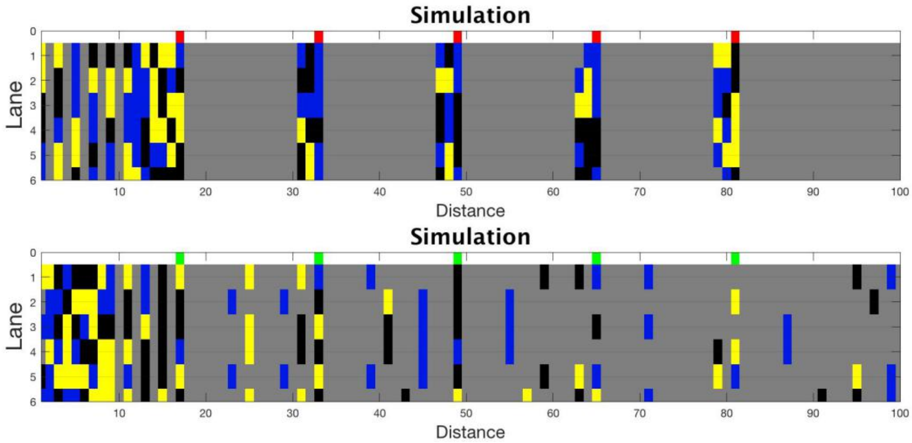

Fig 4.1.2 (a) Synchronized traffic light screenshot

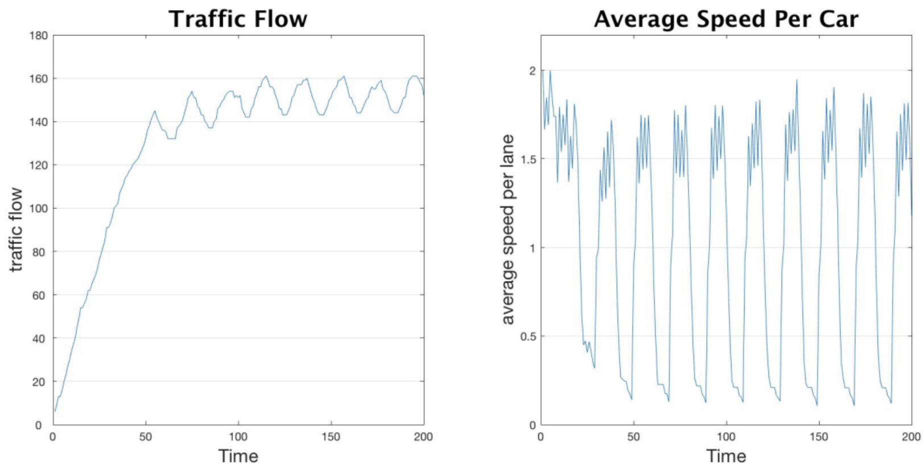

Fig 4.1.2 (b) Traffic flow and mean car speed (Synchronized traffic light)

The Fig 4.1.2(a) and 4.1.2(b) showed the simulation results under synchronized traffic light condition. The statistic result of it is as follows: 40 time steps later, the average traffic flow is 151, the average car speed per second is 1.0434, the highest car speed and traffic flow is 2.00 and 161 separately.

It was obvious in the Fig 4.1.2(a) that the different traffic light condition would cause different traffic flow. From the Fig 4.1.2(b), we could see that the line has larger amplitude. In the first 50 seconds, the traffic flow would increase from almost 0 to the 140 increasing rapidly and start fluctuating around the 150 with range of -10 and 10. The initial car average speed would be 2 because all the car would keep the velocity at 2 until they have to stop at the first traffic light. It shows that the traffic flow would first drop down steadily, because the proportion of the car with speed of zero would increase and then the average speed would fluctuate around 1. The fluctuation means the cars would have similar reaction and are inclined to speed up and stop at the same time.

4.1.3 Asynchronized traffic light

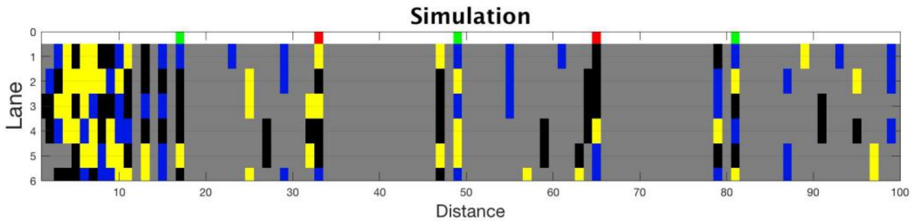

Fig 4.1.3 (a) Asynchronized traffic light screenshot

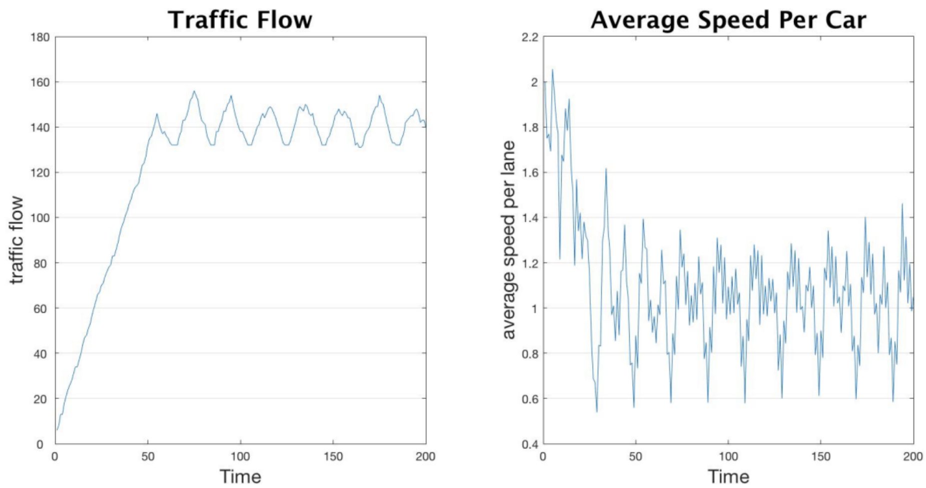

Fig 4.1.3 (b) Traffic flow and mean car speed (Asynchronized traffic light)

The Fig 4.1.3(a) and 4.1.3(b) showed the simulation results under synchronized traffic light condition. The statistic result of it is as follows: 40 time steps later, the average traffic flow is 140, the average car speed per second is 1.0394, the highest car speed and traffic flow is 2.03 and 156 separately.

According to the Fig 4.1.3(b), the traffic flow had several fluctuations which were determined by the traffic light time interval, gap of the car, and the gap of the light. The traffic flow in the first 50 seconds would increase steadily and at the time of 50, and then fluctuated around 140.

4.2 Analysis at the end of the traffic flow

We recorded the number of cars out of the scope at the final simulation time, i.e., simulation time 200, to represent the capability of cars in the specific road. We discussed some parameter's effect on the average capability (total capability divides number of lanes = 6) like time interval, number of lights and car velocity by only changing the discussed parameter at the initial condition.

4.2.1 Time interval

Since the simulation time is 200, we choose the time interval from 1 to 200 with lights under both synchronous and asynchronous condition. And we could draw the graph below.

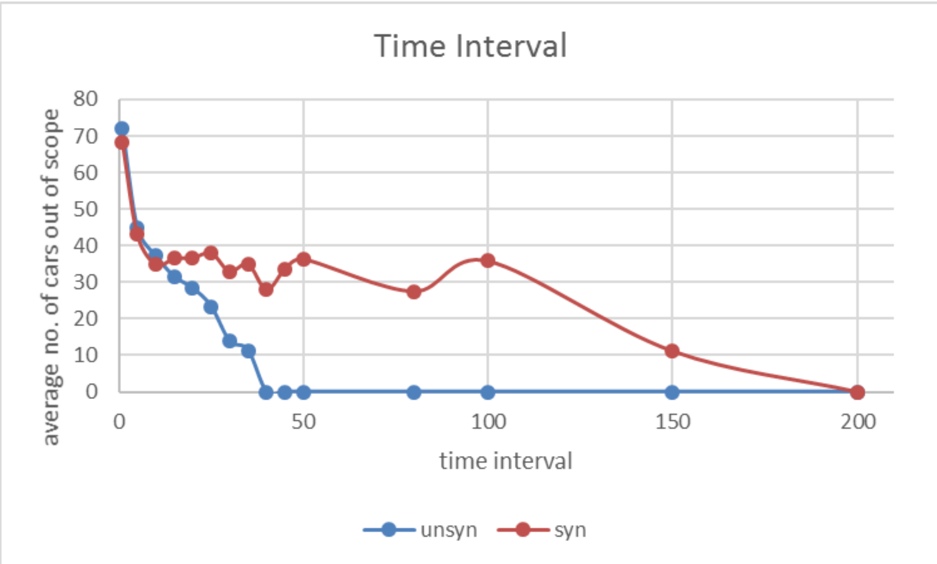

Fig 4.2.1 Average no. of cars out of scope-time interval

We can see that the time interval effect on the average number of cars is significant. For the curve under asynchronous condition, it is correct that the number of cars can out of scope is zero when time interval is 40 since simulation time is 200 and the number of lights is 5. To be specific, the cars are still waiting for the last green signal when the simulation time is up. In addition, it is logical that the time interval effect on asynchronous condition is greater than that of synchronous condition since cars move forward when the current signal becomes green but they will wait for the next traffic light turning green because of asynchronization of adjacent lights. For the curve under synchronous condition, it can keep some number of cars even when the time interval is large (e.g., time interval is 150) since the remain simulation time is sufficient for cars to pass the road when light is green (all green signals actually). And the number of exiting cars is of course zero when time interval is 200.

4.2.2 Number of lights

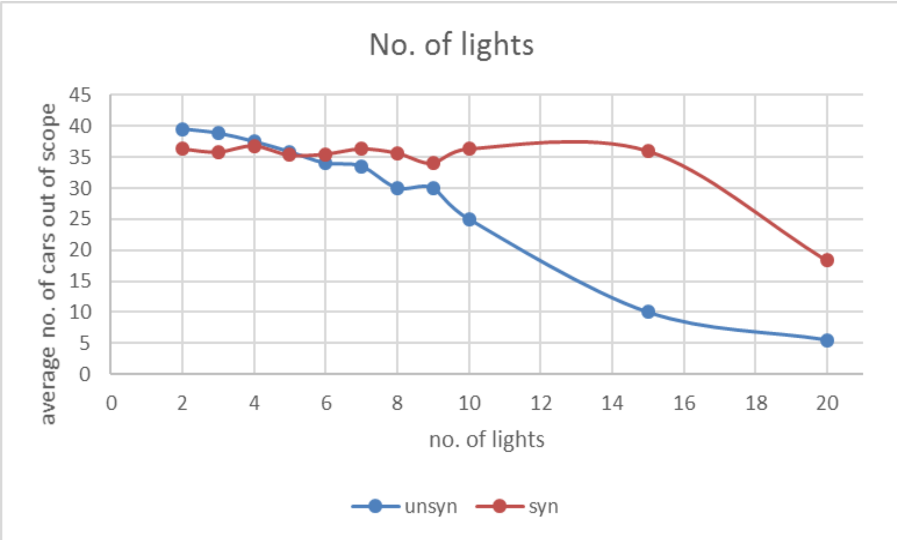

Fig 4.2.2 Average no. of cars out of scope-no. of lights

We selected the number of lights from 2 to10 with reality perspective and chose 15 and 20 for testing our guess. It showed that the effective number of lights has on the average number of existing cars is more significant under asynchronous condition than that under synchronous condition. It can be explained that adding number of lights when lights are asynchronously working is something like increasing the time interval and for the synchronous condition that the number of cars decreases slightly when the number of lights is small because of the synchronization property and when the number of lights is large enough (e.g. 20), the effect caused by something like time interval increment defeats the effect caused by such synchronization, thus the average number of cars out of scope decreased heavily

4.2.3 Car start velocity

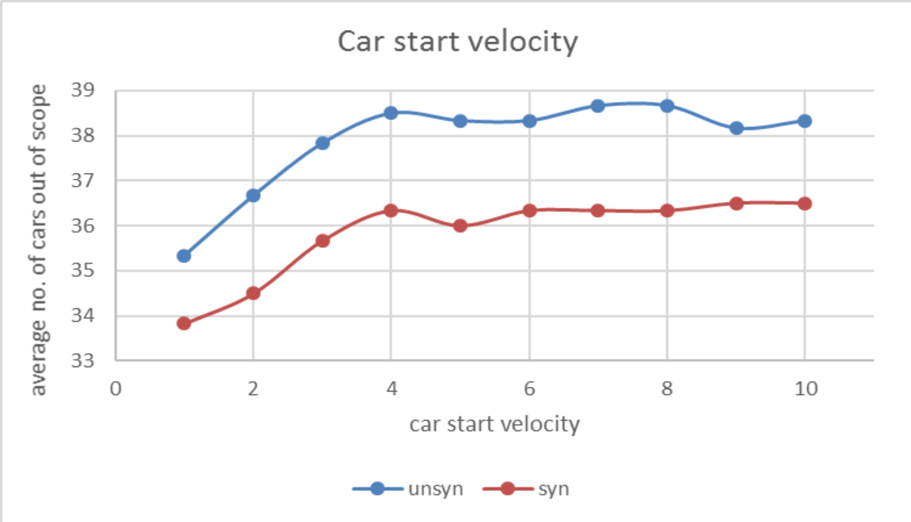

Fig 4.2.3 (a) Average no. of cars out of scope-car start velocity

We changed the car start velocity while keeping the car restart velocity as 2 and the velocity upper bound at 6 and then drew the graph above. We can find that the trend of the two curves is similar and asynchronous condition has a better performance. And the average number of exiting cars doesn't change heavily along with the velocity increment from 4 since we set the velocity upper bound as 6 (when the car start velocity is 4, the velocity of three cars is 5, 6 and 7 respectively) and the car restart velocity is always 2.

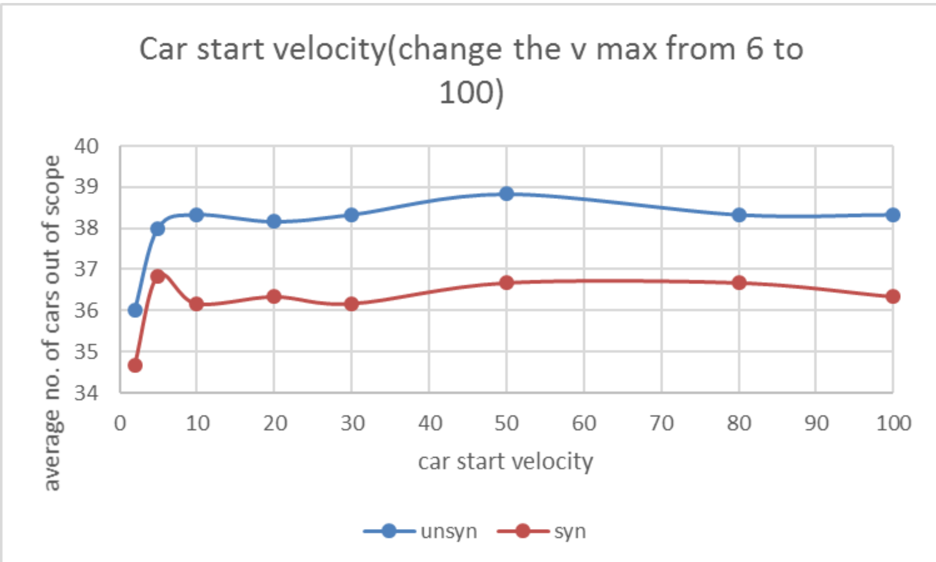

Fig 4.2.3 (b) Average no. of cars out of scope-car start velocity(v_max=100)

We changed the car start velocity while keeping the car restart velocity as 2 and the velocity upper bound at 100 and then drew the graph above. We can find that the result is similar as the result above when the upper bound is 6 since the car restart velocity determines the average amount of exiting cars compared with car start velocity and in this condition the car restart velocity keeps the value 2.

4.2.4 Car restart velocity

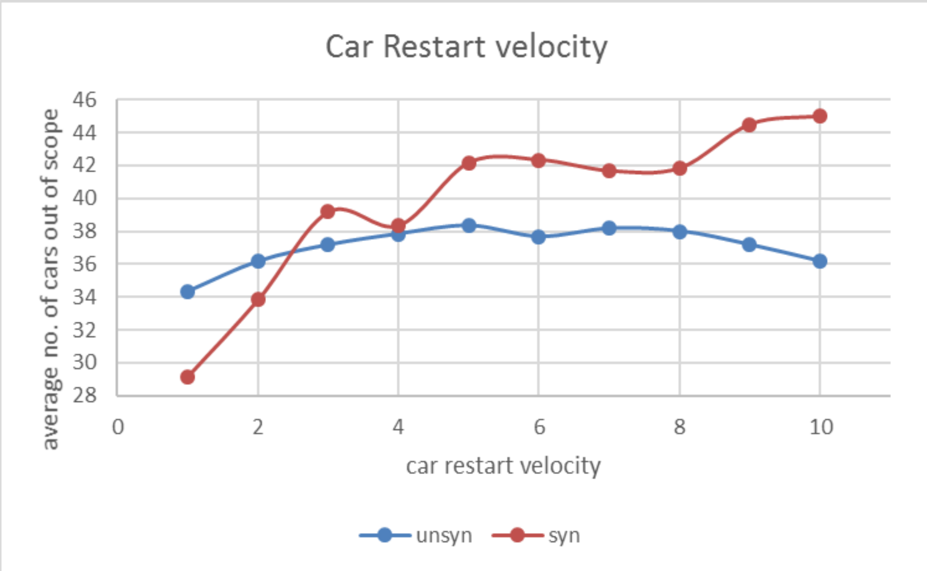

Fig 4.2.4 (a) Average no. of cars out of scope-car restart velocity

We changed the car restart velocity while keeping the car start velocity as 2 and the velocity upper bound at 6 and then drew the graph above. We can find that this change has a greater effect under synchronous condition than asynchronous condition and the average amount of exiting cars under synchronous condition is much larger than changing the car start velocity which has been discussed above.

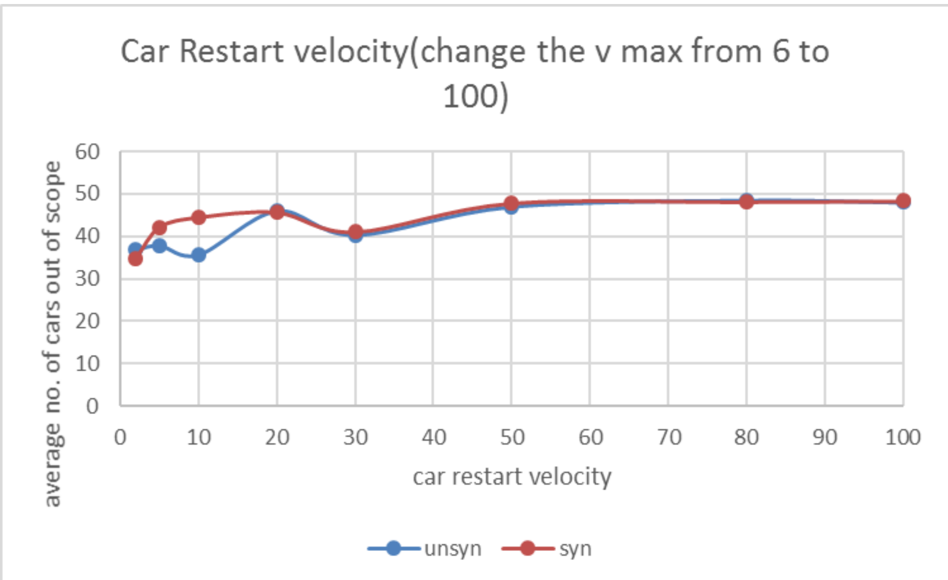

Fig 4.2.4 (b) Average no. of cars out of scope-car restart velocity(v_max=100)

We changed the car restart velocity while keeping the car start velocity as 2 and the velocity upper bound at 100 and then drew the graph above. We can find that the effect caused by restart velocity is great since both asynchronous condition and synchronous condition have the average amount of exiting cars increase to around 50 when the restart velocity is in a large value.

4.2.5 Car velocity

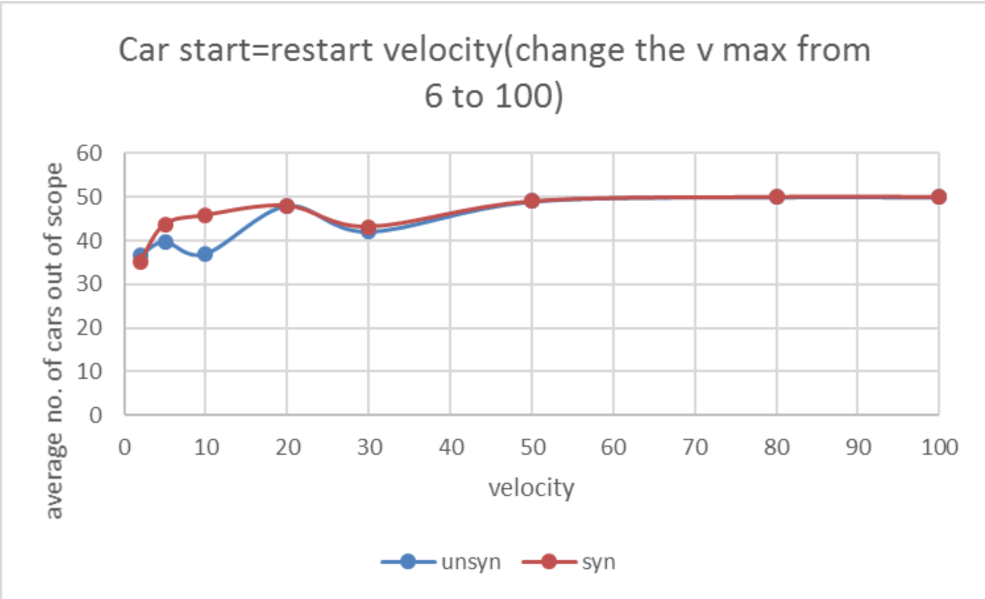

Fig 4.2.5 Average no. of cars out of scope-car velocity

We changed the car start velocity and car restart velocity while keeping the two with the same value and the velocity upper bound is 100 and then drew the graph above. We can find that the following graph is extremely similar with the result obtained by only changing the restart velocity from which we can draw a confident conclusion that the car restart velocity has a dominant effect on the average amount of exiting cars compared with start velocity.

References

[1]Angel O, Holroyd A, Martin J. The jammed phase of the Biham-Middleton-Levine traffic model[J]. Electronic Communications in Probability, 2005, 10: 167-178.

[2]Chowdhury D, Schadschneider A. Self-organization of traffic jams in cities: Effects of stochastic dynamics and signal periods[J]. Physical Review E, 1999, 59(2): R1311.