CVEN 618 TERM PAPER

Spring 2022

TITLE:
COMPARING BID-BASED SIGNAL CONTROL SYSTEM IN A CONNECTED ENVIRONMENT WITH FIX TIMED SIGNAL CONTROL SYSTEM

Submitted by Pranik Koirala

Date: 05/11/2022

Contents

Abstract iii
1. Introduction 1
  1.1. Objective 2
  1.2. Organization of the report 2
2. Literature Review 3
3. Methodology 4
  3.1. Simulation 4
    3.1.1. SUMO 4
    3.1.2. Vehicle 5
    3.1.3. Fix timed signal control 5
  3.2. Bidding based Signal 5
  3.3. Connected Vehicle Penetration 6
4. Result 7
  4.1. Bid Distribution 7
  4.2. Queue Length 7
  4.3. Delay 8
  4.4. Delay of high bidders 8
5. Discussion 9
  5.1. Queue Length 9
  5.2. Delay 9
  5.3. Delay of high bidders 9
  5.4. Cyclicity of queue dissipation 10
6. Conclusion and Recommendation 12
  6.1. Conclusion 12
  6.2. Recommendation 12
7. References 13
Appendix A Results ....
Appendex B SUMO Code ....

ABSTRACT

Traditionally signal controls like a fix timed signal control, actuated signal control is used for right of way allocation at intersections. A novel idea of signal control based on biding is also being explored by researchers where the traditional measures like delay or LOS or queue length is not used to determine efficiency of a signal control but value of time of users is used. In this term paper, an isolated intersection where green signal is allocated based on total bid in a direction is simulated using SUMO and compared to a base environment. Biding by connected vehicle is assumed to come from a normal distribution with an additional component based on the number of vehicles in the queue in front of the approaching vehicle. The average delay of different combination of penetration rate and biding approach is compared. As well as, delay and time loss at intersection of high value of time vehicles are compared. The result shows that this approach can have some advantages but mostly fix timed traffic control strategy results in less delay in average and for high value of time vehicles as well.

1. INTRODUCTION

One of the main purposes of traffic engineering is to promote efficient movement of people and goods. Vehicles, people and goods has to go through intersections to travel through a network. An intersection is a shared space where vehicles intending to travel in different direction meets. An intersection, being a shared space, need to have a system which allocates the space and time. Allocation of right of way at an intersection which needs it is usually done by a signal control.

There are different kinds of signal control methods. The oldest and most used is the fixed time control. These are simple traffic control signals that use fixed phase patterns with fixed cycles. With fixed-time signals, the lengths of the yellow, red, and green intervals remain the same during a designated time period. In a fixed-time signal control, the cycle length is optimized using methods developed by scholars such as Webster or by using different software such as Syncro. There are many limitations of a fixed timed signal control, to overcome the limitation of fixed timed signal control actuated or adaptive signal control is used.

Actuated signal operations consider real-time traffic demands to some extent. Unlike fixed time signal control, the actuated signal operators make use of detectors to sense incoming vehicle information and make changes to the signal as needed. Like actuated, adaptive signal control also is detects current traffic condition and makes changes to the signal control according but adaptive signal control is a broader term used by researchers for a more complex algorithms which makes use of many factors and input and also predicts current traffic condition in an intersection or a network to optimize the signal controls.

In this traditional method of signal control, the efficiency of the signal control is measured in terms of control delay, queue length. Most algorithms of actuated and adaptive signal control make use of these variables (control delay, individual delay, queue length, LOS) as the input but value of time of individual vehicle is not considered while optimizing the signal control. Beside these conventionally measure for allocating right of way there can be other ways of allocating right of way based on different efficiency measures. The allocation of right of way can be seen as a resource allocation problem. Resource allocation problems deal with the assignment of resources to activities and the scheduling of those activities. For example, resource allocation pertains to the allocation of landing slots at airports, CPU time slices for computers, and beds and medical personnel in healthcare facilities. Allocating the resource (in this case time and space at an intersection) is usually done on the basis of decreasing overall delay of vehicles or keeping the queue at a minimum. There could be other ways at looking at this problem. From an economic stand point, value of time (VOT) of individual driver or passenger might not be the same and minimizing the value of time (VOT) can also be a measure of efficiency for the allocation of intersection time and space. Value of time as a loss function for optimizing signal control is being used by only few research papers as of now, such as [1], [2] [3]. Since this is a novel idea and not many researchers have explored this idea, it is still not known if this way of measuring and designing a signal control is a valid way of allocating signal control.

1.1.OBJECTIVE

The primary objective of the study is to simulate a connected environment in SUMO where connected vehicles can bid according to their perceived value of time for green time at their direction and compare it with a fixed timed intersection. For this purpose, the following specific objectives should be achieved:

- To make a simulation model with connected vehicles which can bid for green light.
- To simulate the bidding of the connected vehicles in a random manner and also incorporate a component that is proportional to the length of queue in front of the vehicle.
- To simulate a model with the same vehicles but in a fixed signal intersection.
- Comparing the delay experienced by vehicles in different scenarios.
- Observing the delay experienced by high bidders.

An isolated intersection with detectors and connectivity is designed for the simulation. The bids are understood as perceived value of time of a vehicle and green time is allocated to the highest total bid in one of two directions.

1.2.ORGANIZATION OF THE REPORT

This report is composed of six sections. The first section gives an introduction including the motivation of the study and the objective. The second section is the literature review which provides a review of the previous research on bidding-based signal control and queue psychology. The third section talks about the methodology and tools utilized to realize the project. It also gives the workings of the algorithm used for the experiment. The fourth section puts out the results of the simulation. The fifth section discusses the results and provides comparative analysis. The final section gives a conclusion and recommendation for further study in this topic.

2. LITERATURE REVIEW

The notion that the idea of free market can be utilized to alleviate congestion in transportation, especially freeway has been explored plenty. Papers like [4] has explored the effectiveness of toll roads in improving operational performance of vehicles. It showed the operation of a toll road does not reduce the performance measure on near vicinity corridor. Similarly, [5] showed that toll roads made an impressive contribution to the transportation systems of some European countries. Similar to toll roads which uses the free market to decrease the travel time for those with high value of time the concept of bid-based signal control at intersection is novel idea explored by some researchers which will be discussed below.

There have been a handful of papers published which incorporates the idea of a bid based signalized traffic control in a connected environment. The idea of signalized traffic control is not new but researchers are always in search for newer and more efficient ideas for traffic control. [1] proposed a concept of assigning green time as long as the cumulative bid of a particular direction is more than its conflicting direction. Asymmetric simple exclusion process was used for the assessment to queue delay, it showed that this new approach produced a greater subjective benefit. Here subjective benefit is the benefit the user got in comparison to a base case.

[3] In their 2017 paper aims to develop a model that can expediating high value of time drivers without undue disruption of overall traffic flow. Here, the number of high bidders is also taken into account before allocating the green time. The result does not show an improvement in delay compared to base case. Similarly, [2] In their 2013 paper proposed an auction based autonomous intersection management system. The researchers used AORTA to simulate bidding-based intersection green time allocation in a network. The result did not show any improvement in high bidding vehicles but introduced some new ways on how autonomous vehicles can bid like the concept of static wallet and fair wallet. It also utilized an idea of a benevolent system agent which will not let the rich jeopardize the system and not let low bidding vehicle suffer a lot.

[6] says that time price is different from money price as the value of time for different individual can be different. The perceived value of time might depend on many factors for any individual waiting for a service. Occupied time feels shorter than unoccupied time [7] and, consequently, waiting for a green light can feel like more time lost than actual. This perception might affect a vehicle's bid in a bid-based signal control environment. Research like [8], [9] observed queue has an effect on the behavior of service seekers in a queue waiting for a service. These observation calls for a investigation on bid based signal control setting where bid is also based on queue length in front of approaching vehicle.

3. METHODOLOGY

3.1. SIMULATION

Computer model that replicates real world scenario is called a simulation. In a traffic simulation, the behavior of vehicles with respect to different control strategies, infrastructure, vehicle characteristics, driver characteristics is studied. Based on the amount of detail, there are three types of traffic simulations: macroscopic, mesoscopic, and microscopic. A macroscopic model is one in which traffic is viewed as a continuous stream with little detail. The amount of detail in a mesoscopic model is moderate, and there is less interaction between the individual cars. A microscopic model, on the other hand, provides the maximum level of detail, and the vehicles are replicated second by second.

3.1.1. SUMO

There are many simulation tools available for microscopic simulation with connected vehicle environment. SUMO was chosen for this study as it is flexible and availability. SUMO is a simulation software with can work with different other software, libraries and systems to run a dynamic traffic simulation with real time inputs. The German Aerospace Center developed SUMO. The ability to alter SUMO software at the source code level is well-known. Custom models can be added to SUMO, and the simulation can be controlled remotely using a variety of APIs. SUMO can utilize detectors to count how many vehicles went via a specific stretch. There are three files that are essential for a simulation run in SUMO. They are net file where the information about the network is stored, a route file where the information regarding the attributes of the car and the types of flows used in the simulation and the configuration file which creates the configuration for the simulation.

TraCI is a very important API in SUMO which is used for dynamic traffic control in the simulation. TraCI allows user to alter the simulation in real time. It is implemented using python script in this project. This study also utilizes TraCI to create a connected environment where vehicles can place bids and the signal is controlled with respect to the highest total bid in two directions.

Detectors in SUMO are another component which is extensively used to mimic a connected environment in the project. There are three kinds of detectors in SUMO, namely loop detectors, lane area detectors and multi-entry-exit detectors. This study makes use to loop detectors and lane area detectors. The detectors are created in the simulation via an additional file. An additional file has to be created along with the network file and this file has to be specified in the input as an additional file. There are certain attributed that have to be described in the additional file such as id-name(id) of the detector, lane id, length (in case of a lane-area detector). At the end of the simulation, data is collected via these detectors to evaluate the performance of a traffic system.

A snap shot figure of actual simulation with Human driven vehicle (HDV) in white and Connected Vehicle (CV) in blue is shown in the next page.

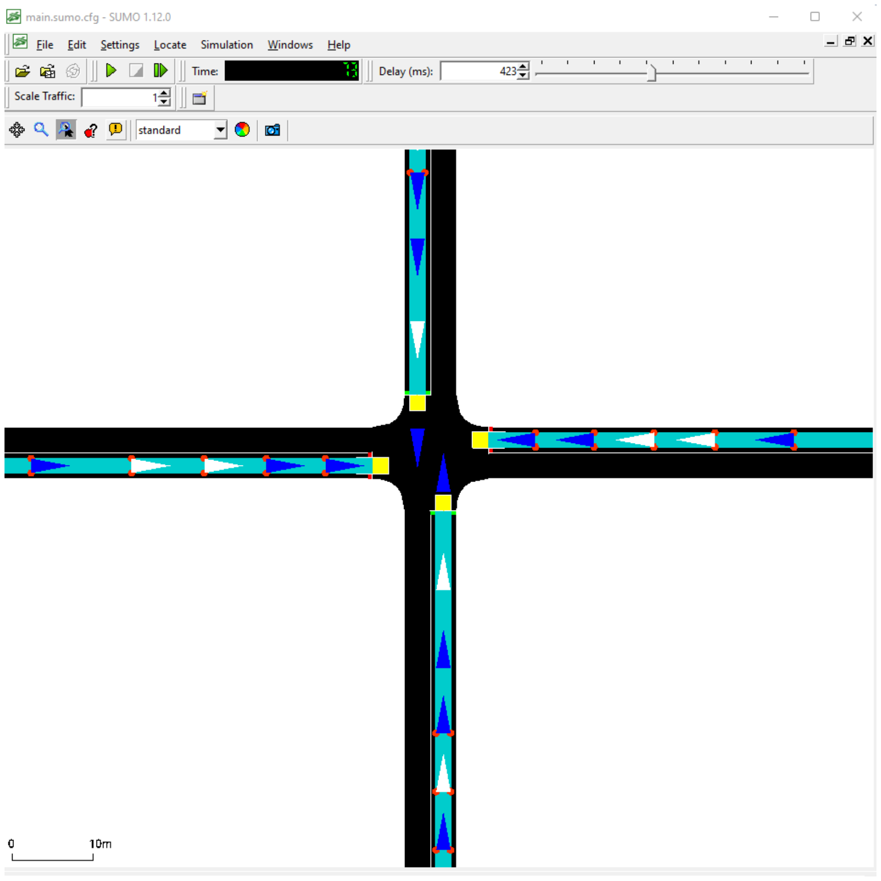


Figure 1 Snap of simulation in SUMO

3.1.2. Vehicle

The vehicles used for the simulation has mostly the same properties across the board. The properties of the vehicle are described below.

The deceleration ability of the vehicle is set to be 6.0 (m/s^2) for all types of vehicles. The minimum gap in front of the vehicle is set to be 2.5 m. The netto-length (length) of all vehicle is set to be 5.0 m. The maximum speed of each vehicle is set to be 50 m/s. The acceleration ability of the vehicle is set to be 3.0 for HDV and 2.0 for CV. The sigma, which is a car following model parameter, is set to be 0.5 for all the cars.

3.1.3. Fix timed signal control

For the base case the experiment required a fix timed signal control. The fix timed signal control was designed as a two-phase signal control with an all-yellow signal of 3 sec after each green. Since there are no left turns in the intersection for the simulation there were only two phases. Each phase got a green time of 42 sec. The cycle length was 90 sec.

3.2.BIDDING BASED SIGNAL

In this study a bidding-based signal control is simulated where connected vehicle users can place bids which is interpreted as the value of time for that user. Green time is allocated based on the total value of time on each direction. For instance, if the value of time (i.e., total bid) in North South direction is more than the value of time in West East direction, green time is allocated to

North South direction. The intricate details of the algorithm used to realize this environment is discussed below.

Firstly, when a vehicle enters the road leading to the intersection and the vehicle in question is a CV, it places a bid. The bid placed has two component, number one is the random bid which is randomly picked for each CV from a Normally distributed curve with a mean of zero and a standard deviation of 5, the bid picked is always positive. The number two component is the queue component, as discussed in the literature review queue has a psychological effect on people waiting to be served which can increase their tolerance. Therefore, the number two component is added to the bid. The study used two ways of calculating the number 2 component. Half of the number of vehicles in front and total number of vehicles in a queue in front of the approaching vehicle was used for the number 2 component. The simulation was also run excluding the number two component. The three variation of bid calculation used in the simulation is mentioned below.

Bid=Random
Bid=Random + 0.5 * No. of vehicles in queue
Bid=Random + 1 * No. of vehicle in queue

Then, the signal was designed to accommodate the experiment design. A dynamic signal was designed which gave green time to the direction with the highest bid (i.e., the highest value of time). Since turning the signal green and red whenever a direction wins the bid created tons of crashes, we added a minimum yellow time for each signal change. A minimum green and maximum green was also added to the design albeit the maximum green criteria was not used by the simulation to change the signal at any instance except when using very low penetration rate. The minimum green was set to 15 sec and the maximum green was set to 100 sec.

3.3.CONNECTED VEHICLE PENETRATION

Penetration rate is an important factor when doing any simulation or study comprising connected vehicles. In this study where vehicles have the ability to bid for green signal requires constant bidding by connected vehicle or the experiment won't run as intended. When the market penetration is less than 25% (say 15%) the simulation tends to show unintended results so the study uses market penetration rate above 25%. Also, since a 100% penetration is far and this experiment was designed to study a mixed traffic, 100% penetration is excluded from the result and discussion.

4. RESULT

4.1. BID DISTRIBUTION

The positive integer bids were assigned to each Connected Vehicle (CV) from a normal distribution with a mean of zero and standard deviation of 5. The following is a probability distribution function of bid after it was placed for green time which also includes an added queue factor. The density curve is calculated by making use of kernel density estimation, which is a non-parametric method of determining probability density function of a sample of data. The density functions mentioned below are for 25% and 50% penetration of connected vehicle during the simulation.

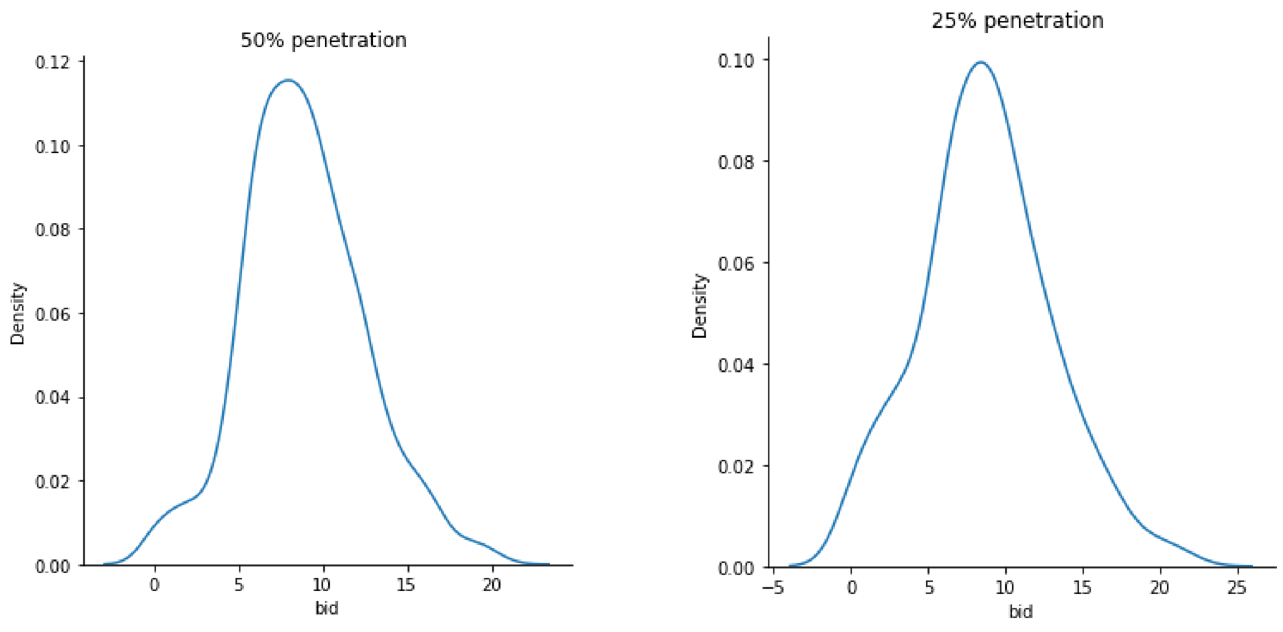


Figure 2 Probability density function of random bid with additive component

4.2. QUEUE LENGTH

Here all the Human driven and Connected vehicles are assumed to have same length of 5.0 m. Average number of vehicle in queue is determined for each scenario. The queue seems to increase as the penetration rate of connected vehicle increases.

Table 1 Queue lengths for different simulations

SN	Scenario	Bid scheme	Average Queue (No. of veh)
1	25% penetration rate	Random	11
2	25% penetration rate	Random + 0.5 No. of vehicle in queue	11
3	25% penetration rate	Random + No. of vehicle in queue	11
4	50% penetration rate	Random	11
5	50% penetration rate	Random + 0.5 No. of vehicle in queue	12
6	50% penetration rate	Random + No. of vehicle in queue	11
7	75% penetration rate	Random	12

8	75% penetration rate	Random + 0.5 No. of vehicle in queue	12
9	75% penetration rate	Random + No. of vehicle in queue	12
10	Fixed traffic control	N/A	10

4.3.DELAY

The average delay of vehicles in each scenario having a unique bid calculating scheme is tabulated below. A loop detector at the end of a intersection is used to record the loss time. Each vehicle while passing through the intersection recorded its loss time (i.e., delay).

Table 2 Average delay for different simulations

SN	Scenario	Bid scheme	Average Delay (sec)
1	25% penetration rate	Random	69.38
2	25% penetration rate	Random + 0.5 No. of vehicle in queue	70.58
3	25% penetration rate	Random + No. of vehicle in queue	73.43
4	50% penetration rate	Random	75.36
5	50% penetration rate	Random + 0.5 No. of vehicle in queue	78.56
6	50% penetration rate	Random + No. of vehicle in queue	77.21
7	75% penetration rate	Random	77.79
8	75% penetration rate	Random + 0.5 No. of vehicle in queue	82.01
9	75% penetration rate	Random + No. of vehicle in queue	82.47
10	Fixed traffic control	N/A	53.12

4.4.DELAY OF HIGH BIDDERS

In our simulation more than 11 bid points fall in the 75th percentile for 25 percent CV penetration scenario. Since we are considering this scenario as our base (based on the average delay and queue component in a bidding environment) we will be comparing the average delay of vehicles who placed a bid more than 11 in each scenario. For the fixed timed scenario, the same vehicles which placed a high bid in scenario 3 (i.e., 25% penetration and 0.5queue additive component) is considered as high bidders.

Table 3 Average delay of high bidders in different simulations

SN	Scenario	Bid scheme	Average Delay of high bidders(sec)
1	25% penetration rate	Random	62.59
2	25% penetration rate	Random + 0.5 No. of vehicle in queue	73.25
3	25% penetration rate	Random + No. of vehicle in queue	75.70
4	50% penetration rate	Random	63.70
5	50% penetration rate	Random + 0.5 No. of vehicle in queue	79.97
6	50% penetration rate	Random + No. of vehicle in queue	80.05
7	75% penetration rate	Random	57.26
8	75% penetration rate	Random + 0.5 No. of vehicle in queue	85.25
9	75% penetration rate	Random + No. of vehicle in queue	85.87
10	Fixed traffic control	N/A	54.78

5. DISCUSSION

The queue length and average delay of bid-based signal simulation environment with 25% penetration rate and bid equaling a random component and a queue component which consist of half the number of vehicles in queue (random+0.5queue) was found to be the most minimum. Therefore, the discussion below mostly focusses on comparing this simulation environment with others.

5.1.QUEUE LENGTH

The average queue length seems to increase as the penetration rate of connected vehicle increase. This might be because of the frequency of signal change increases as the number of bidders increase for right of way in an intersection. The average queue at fixed signal is seem to be 10 vehicles (each vehicle length being 5.0m), which is lower than the queue length of bid-based signal control environment.

5.2.DELAY

Delay is an important metric to compare performance of an intersection. The average delay of the intersection at every scenario and bid scheme was calculated. The average delay of fixed timed intersection was found to be the least among all. Below is a comparison if delay in a fixed timed traffic control scenario and 25% penetration and 50% penetration scenario for a bid-based signal control.

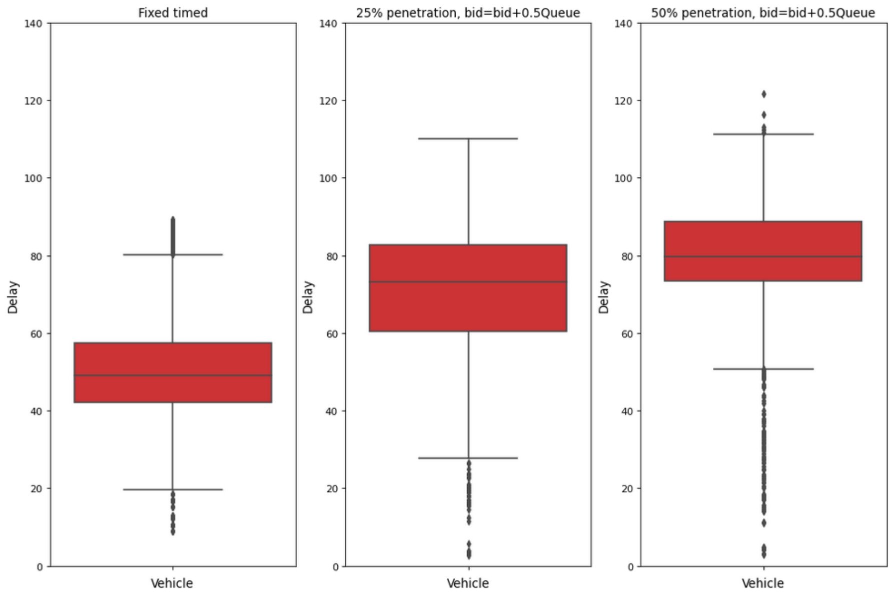


Figure 3 Box-plot of delay in different scenarios

5.3.DELAY OF HIGH BIDDERS

From the results, we can say that vehicles bidding more than 11 points in different scenario has different results. It can be noticed that average delay of high bidders is proportional to variation of penetration rate of CV but it is also directly proportional to the additive component of bid (i.e., the queue). Delay of high bidders increase as the queue in front of them is long is because the high bidders are high bidders because one reason is the high queue length in the first place. The does not give much meaning as high bids are not independent on the queue length.

Another approach would be to compare high bidding vehicle in one simulation environment with others in the same environment and fix time signal. Since average delay is not the efficiency measure, we are primarily focused as a measure of metric of evaluating a traffic control. We will be instead focusing on value of time of bidders to see if their delay is decreased because of the bid based right of way allocation approach. To do so, comparison of delay of high bidders against other vehicles in a scenario and high bidder's delay vs the same vehicle's delay at a fix timed signal control environment is analyzed. There are 92 high bidders in the simulation with 25% penetration of CV, the bid is random bid with an additive term based on queue (0.5 No. of vehicles in queue).

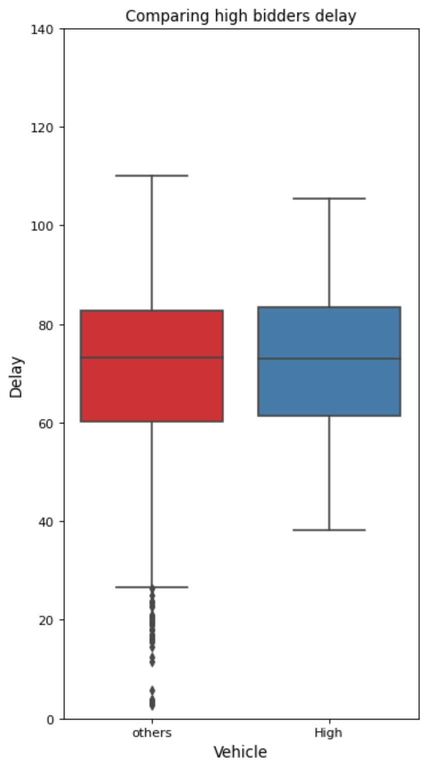


Figure 4 High bidder (75th percentile) vs others delay boxplot for 25% penetration CV

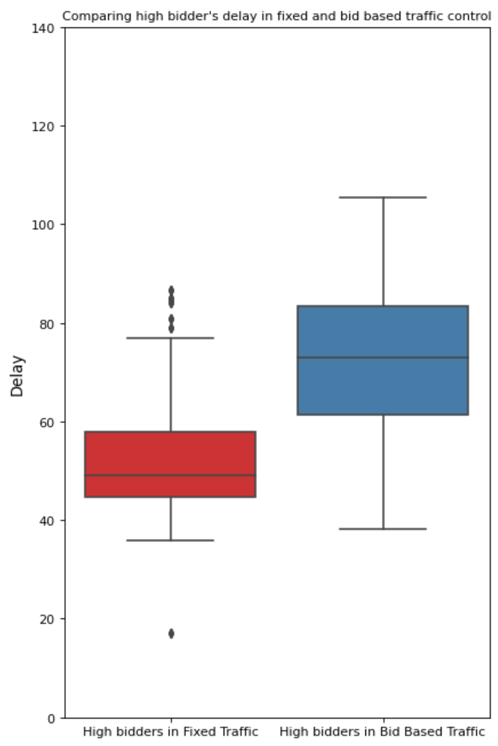


Figure 5 High bidder's delay vs same vehicle's delay in a fixed environment

As seen in the box plot above, the delay of high bidders is not very different in comparison to other vehicles in the simulation. The second box plot compares the delay of same vehicles in a fix timed signal simulation environment. The high biding vehicles seems to have much less delay in fix timed signal environment.

5.4.CYCLICITY OF QUEUE DISSIPATION

As mentioned in the methodology section the queue at direction for a 25% penetration rate scenario where bid is random and also includes an additional term proportional to half of the number of vehicles in queue in from of the bidding vehicle. The time series data of queue for every second is broken down into a trend component, cyclicity component (also known as seasonality component) and residual term (also known as noise). To find out a naïve seasonality decomposition is done for the queue data in WE approach direction. As seen in figure below, the seasonality plot shows there is indeed a cycle in queue formation which could be better addressed by a fixed timed signal or queue based actuated signal.

```
In [153]: result=seasonal_decompose(df['queueWE'], model='additive', period=60,two_sided=False)
```

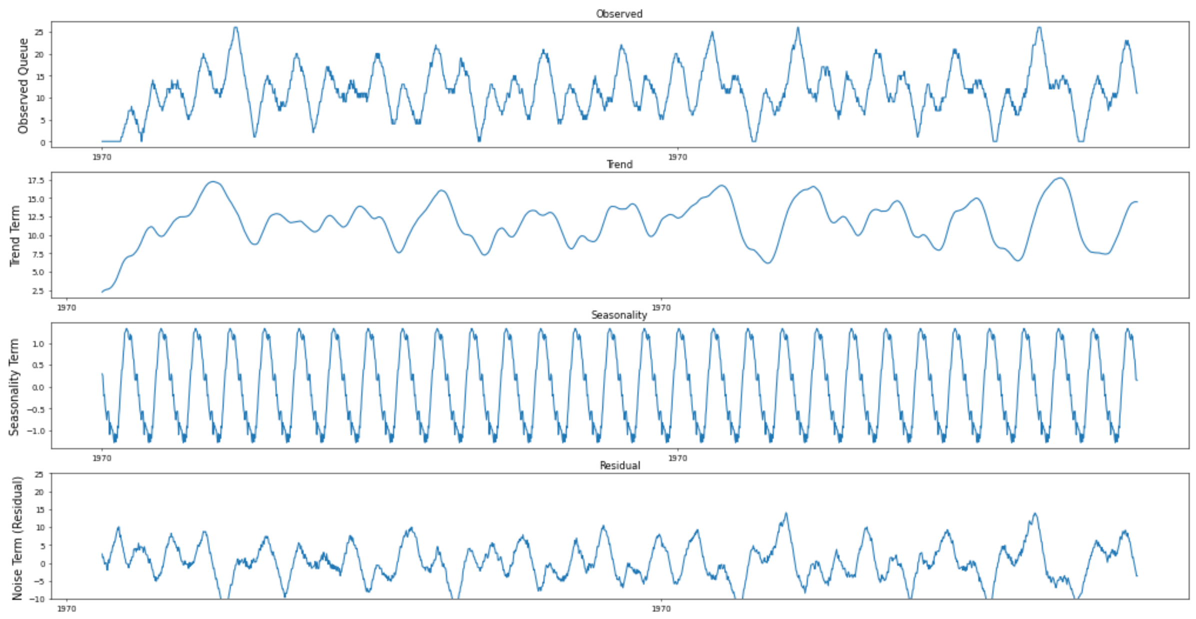


Figure 6 Observed queue dissipation graph with its trend, seasonality and residual component

Figure 7 depicts the dissipation of queue at the allocated green time at an interval of 200 sec to 600 sec.

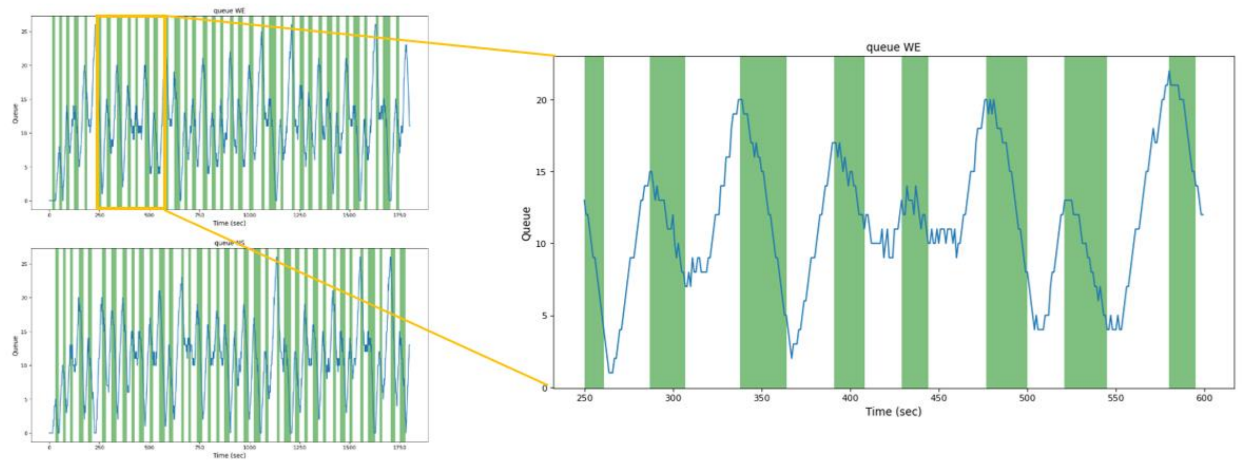


Figure 7 Queue dissipation and allocated green time

6. CONCLUSION AND RECOMMENDATION

6.1.CONCLUSION

- The queue length increases as market penetration of connected vehicle increases because of the frequent signal change as biding is more frequent with more connected vehicles.
- The average delay of the bid-based signal priority control was found to be higher than the base fix timed traffic control.
- Vehicles with high value of time did not benefit by a bid-based signal priority control.
- Queue dissipation showed a cyclic nature which can also be addressed by a fix timed signal control or a queue based actuated signal control.
- Bids which include a queue-based component does not decrease the average delay of the intersection instead it increases it as this might be because signal change will depend upon queue length and it changes in a cyclic manner.

6.2.RECOMMENDATION

- Further study on benefit on high value of time vehicles can be done by randomly identifying a sample of vehicles as high value of time vehicle and analyzing the time taken to pass the intersection in different signal control scenarios.
- The application of a bid-based signal control might be for very specific and niche, more research must be done on its application.
- The efficiency measure used in this paper did not find a bid-based system to work well in a isolated intersection, more thought must be put on the efficiency measure of traffic signal control system.

7. REFERENCES

[1] Y. Z. L. Q. Kentaro Iio, "Bid-Based Priority Signal Control in a Connected Environment: Concept," *Transportation Research Record,* pp. 737-747, 2019.

[2] S. D. B. ,. P. S. D. Carlino, "Auction-based autonomous intersection management," *16th International IEEE Conference on Intelligent Transportation Systems,* pp. 529-534, 2013.

[3] S. F. S. Isaac K. Isukapati, "Accommodating high value-of-time drivers in market-driven traffic signal control," *2017 IEEE Intelligent Vehicles Symposium (IV),* pp. 1280-1286, 2017.

[4] S. S. P. Sonu Mathew, ""How Effective are Toll Roads in Improving Operational Performance?," *Mineta Transportation Institute Publications,* 2019.

[5] S. Carpintero, "Toll Roads in Central and Eastern Europe: Promises and Performance," *Transport Reviews,* pp. 337-359, 2010.

[6] E. S. a. T. N. T. D. Nichols, "Discrimination by Waiting Time in Merit Goods," *The American Economic Review,* pp. 312-323, 1971.

[7] D. Maister, The psychology of waiting lines, Boston: Harvard Business School, 1984.

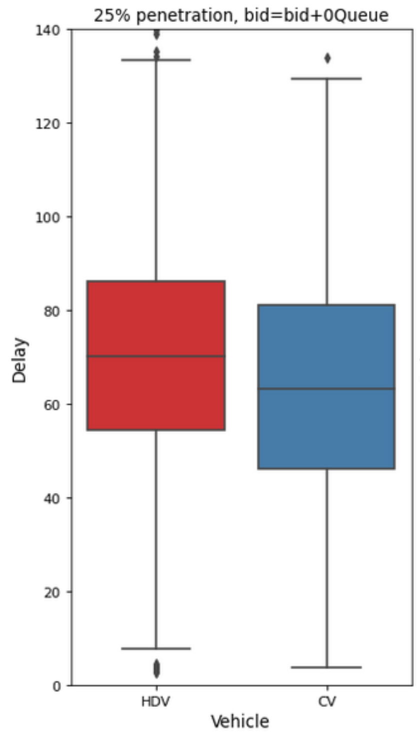
[8] T. J. A. S. A. B. T. N. W. L. Daniel J. Findley, "Evaluation of wait times and queue lengths at ferry terminals," *Research in Transportation Economics,* pp. 27-33, 2018.

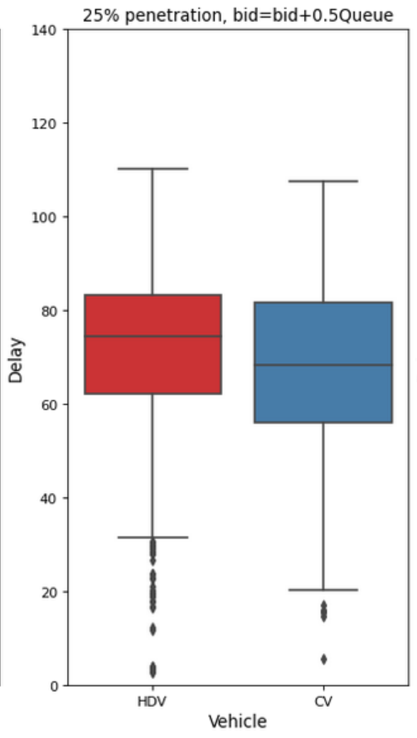
[9] B. G. E. D. G. O. Zeynep Aksin, "How Observed Queue Length and Service Times Drive Queue Behavior in the Lab," 2020.

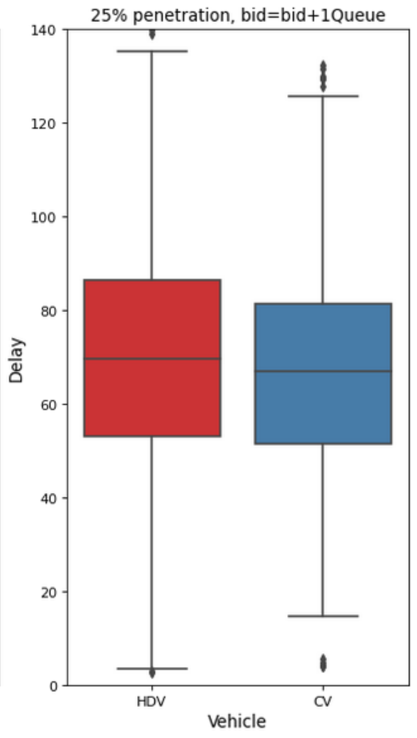
[10] H. O. E. A. Mathias Dharmawirya, "Analysis of Expected and Actual Waiting Time in Fast Food Restaurants," *Industrial Engineering Letters,* 2012.

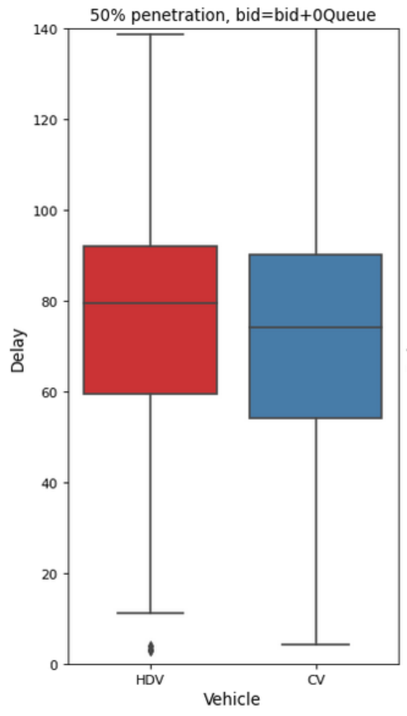
APPENDIX A RESULTS

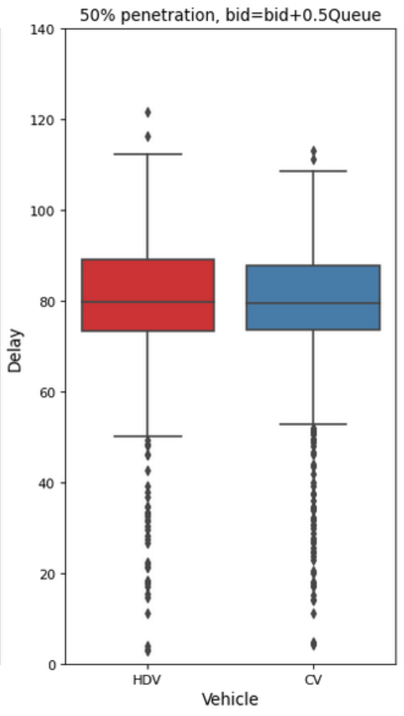
Delay Comparison

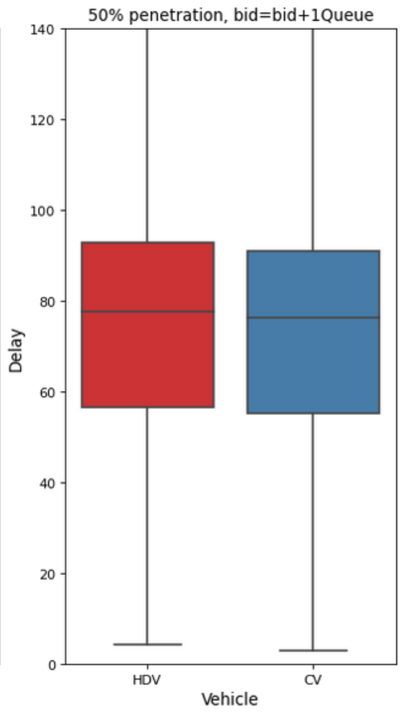


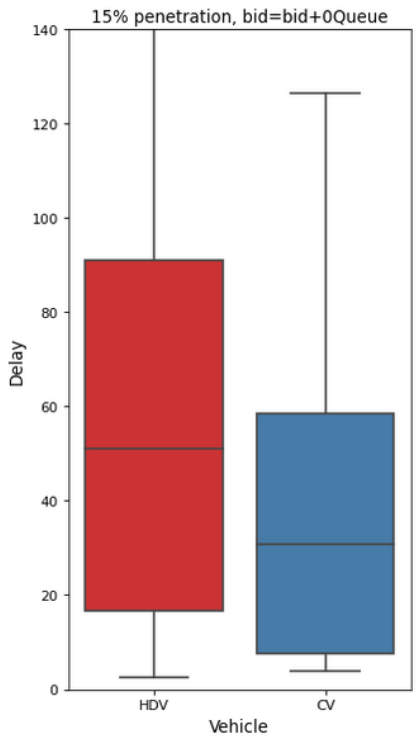


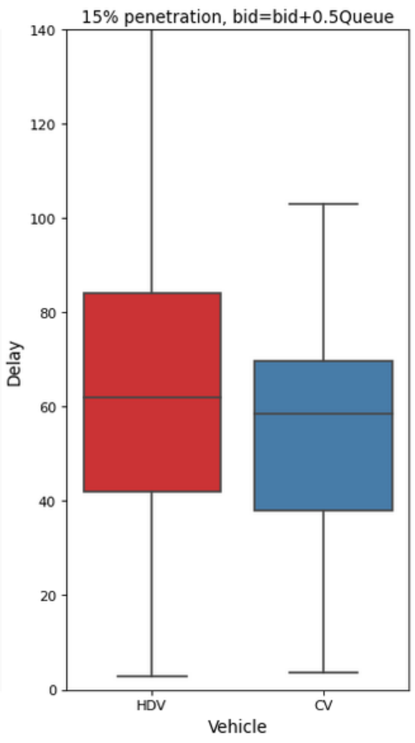


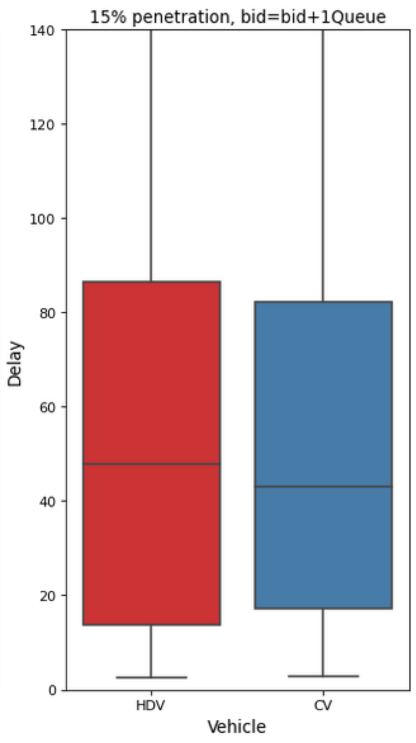


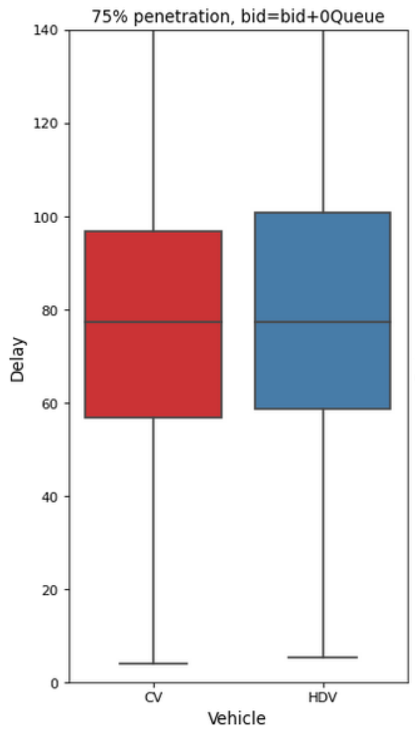


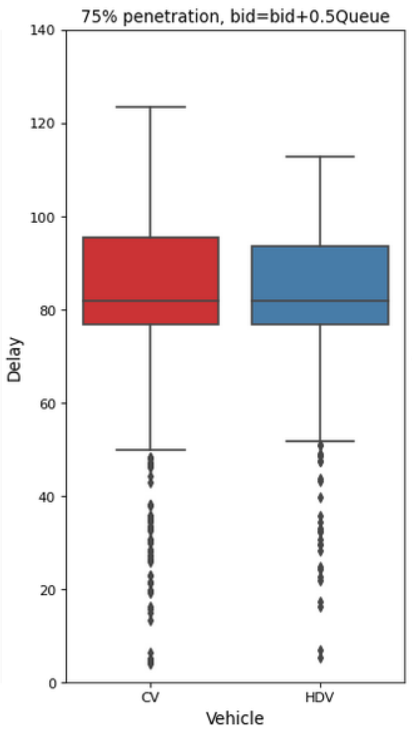


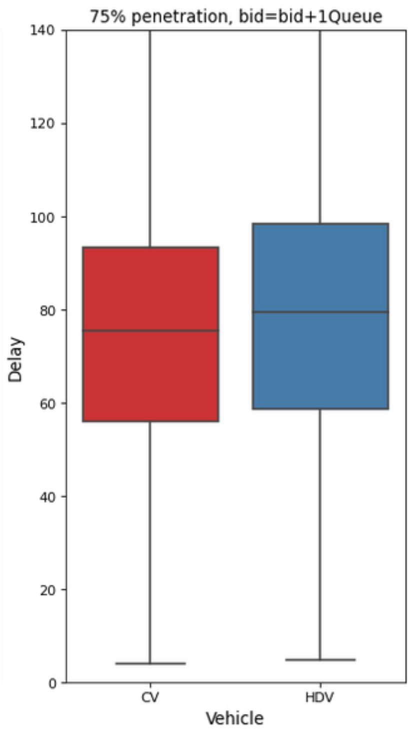

15% penetration, bid=bid+0Queue
Delay
Vehicle
HDV
CV
0
20
40
60
80
100
120
140


15% penetration, bid=bid+0.5Queue
Delay
Vehicle
HDV
CV
0
20
40
60
80
100
120
140


15% penetration, bid=bid+1Queue
Delay
Vehicle
HDV
CV
0
20
40
60
80
100
120
140

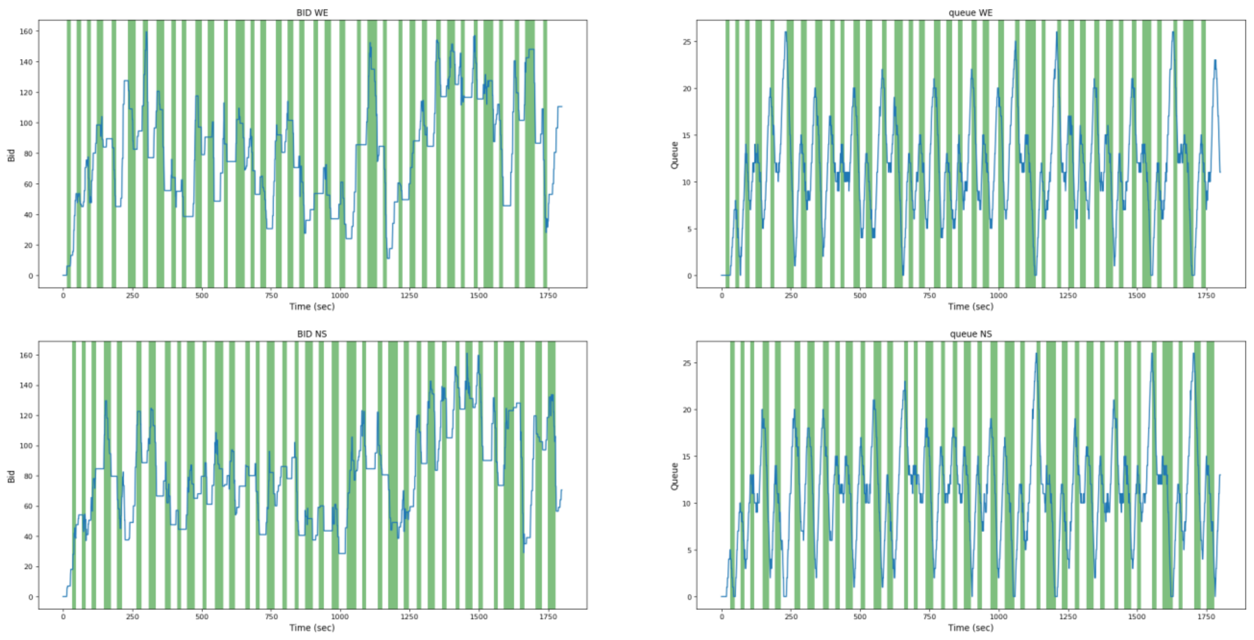

75% penetration, bid=bid+0Queue
Delay
Vehicle
CV
HDV
0
20
40
60
80
100
120
140


75% penetration, bid=bid+0.5Queue
Delay
Vehicle
CV
HDV
0
20
40
60
80
100
120
140

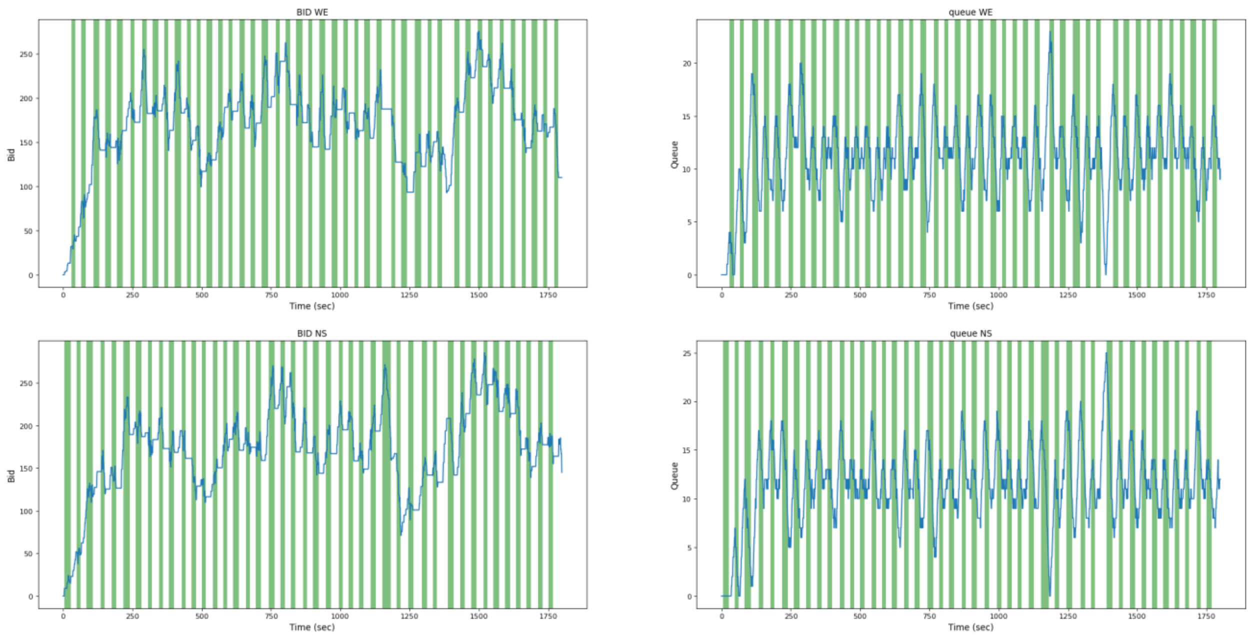

75% penetration, bid=bid+1Queue
Delay
Vehicle
CV
HDV
0
20
40
60
80
100
120
140

Queue

25% bid=random+0.5No of vehicle in queue

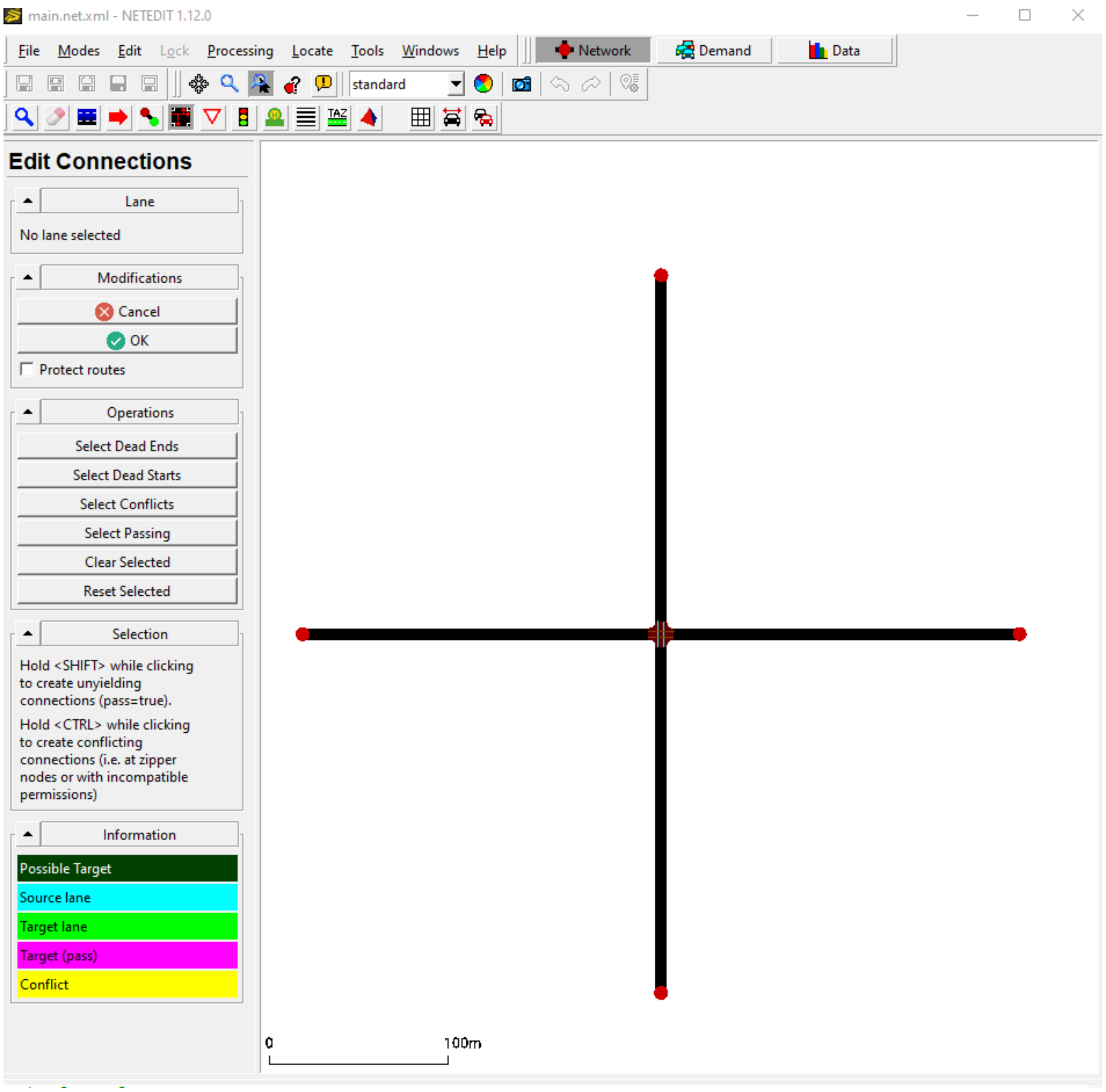


50% bid=random+0.5No of vehicle in queue

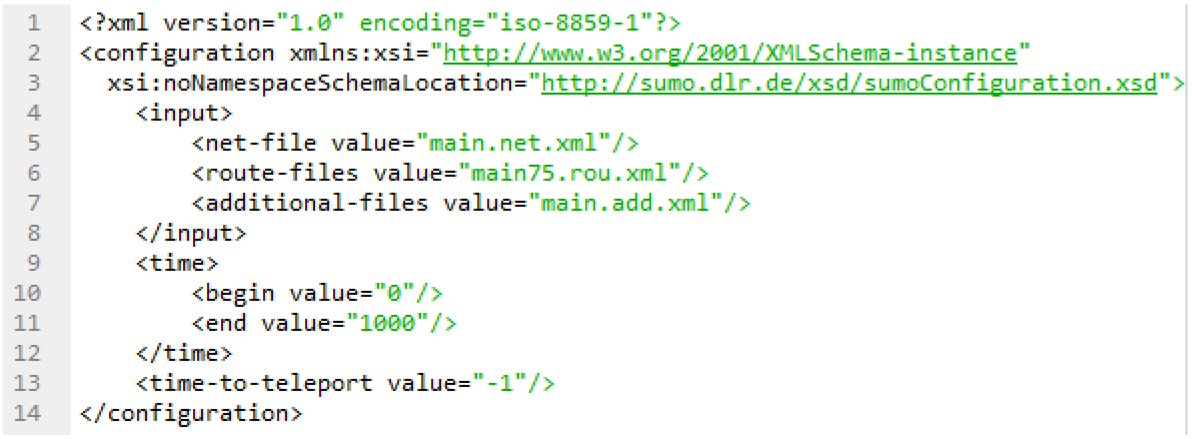


APPENDEX B SUMO CODE

A net.xml file created using NetEDIT. Following is a screenshot of the NetEDIT design used in the project with connections.

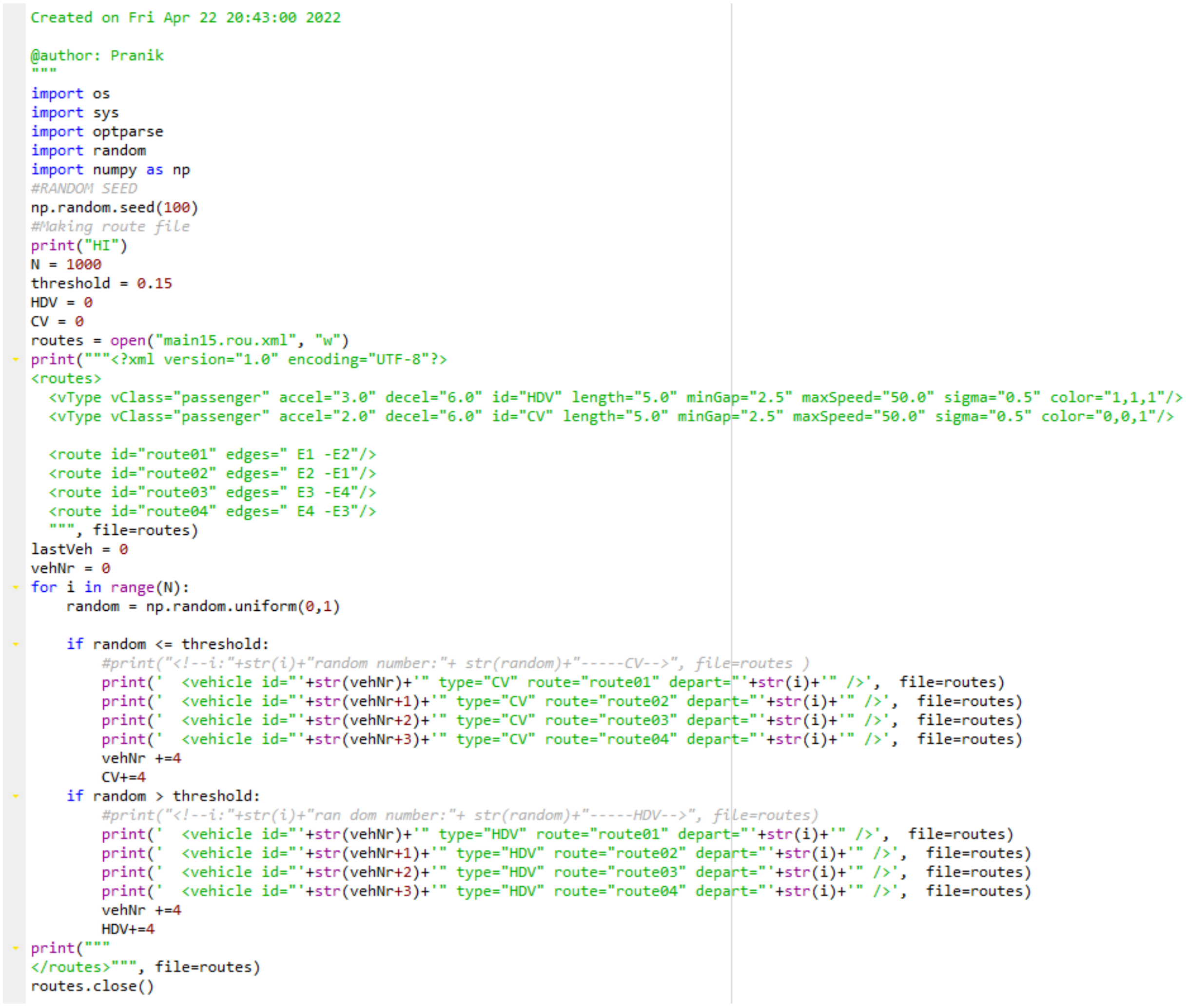


Configuration file:



```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">
    <input>
        <net-file value="main.net.xml"/>
        <route-files value="main75.rou.xml"/>
        <additional-files value="main.add.xml"/>
    </input>
    <time>
        <begin value="0"/>
        <end value="1000"/>
    </time>
    <time-to-teleport value="-1"/>
</configuration>
```

The route file is generated using python. Random vehicles are generated with market penetration as input. The code is mentioned below:



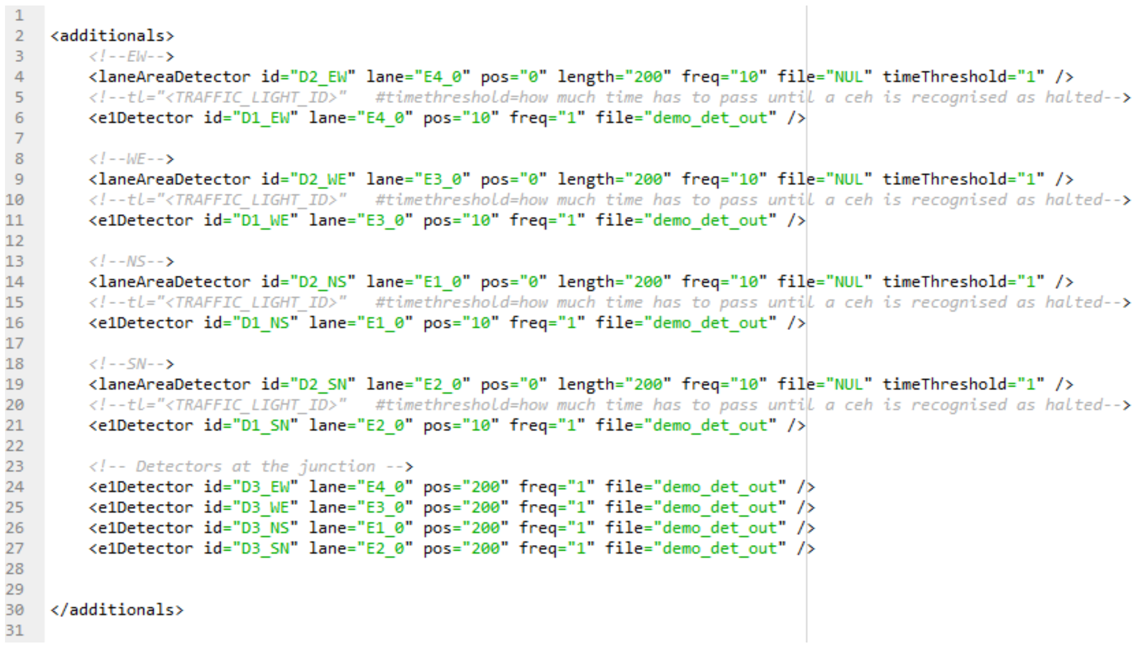
```python
Created on Fri Apr 22 20:43:00 2022

@author: Pranik
"""
import os
import sys
import optparse
import random
import numpy as np
#RANDOM SEED
np.random.seed(100)
#Making route file
print("HI")
N = 1000
threshold = 0.15
HDV = 0
CV = 0
routes = open("main15.rou.xml", "w")
print("""<?xml version="1.0" encoding="UTF-8"?>
<routes>
  <vType vClass="passenger" accel="3.0" decel="6.0" id="HDV" length="5.0" minGap="2.5" maxSpeed="50.0" sigma="0.5" color="1,1,1"/>
  <vType vClass="passenger" accel="2.0" decel="6.0" id="CV" length="5.0" minGap="2.5" maxSpeed="50.0" sigma="0.5" color="0,0,1"/>

  <route id="route01" edges=" E1 -E2"/>
  <route id="route02" edges=" E2 -E1"/>
  <route id="route03" edges=" E3 -E4"/>
  <route id="route04" edges=" E4 -E3"/>
  """, file=routes)
lastVeh = 0
vehNr = 0
for i in range(N):
    random = np.random.uniform(0,1)

    if random <= threshold:
        #print("<!--i:"+str(i)+"random number:"+ str(random)+"-----CV-->", file=routes )
        print('  <vehicle id="'+str(vehNr)+'" type="CV" route="route01" depart="'+str(i)+'" />',  file=routes)
        print('  <vehicle id="'+str(vehNr+1)+'" type="CV" route="route02" depart="'+str(i)+'" />',  file=routes)
        print('  <vehicle id="'+str(vehNr+2)+'" type="CV" route="route03" depart="'+str(i)+'" />',  file=routes)
        print('  <vehicle id="'+str(vehNr+3)+'" type="CV" route="route04" depart="'+str(i)+'" />',  file=routes)
        vehNr +=4
        CV+=4
    if random > threshold:
        #print("<!--i:"+str(i)+"ran dom number:"+ str(random)+"-----HDV-->", file=routes)
        print('  <vehicle id="'+str(vehNr)+'" type="HDV" route="route01" depart="'+str(i)+'" />',  file=routes)
        print('  <vehicle id="'+str(vehNr+1)+'" type="HDV" route="route02" depart="'+str(i)+'" />',  file=routes)
        print('  <vehicle id="'+str(vehNr+2)+'" type="HDV" route="route03" depart="'+str(i)+'" />',  file=routes)
        print('  <vehicle id="'+str(vehNr+3)+'" type="HDV" route="route04" depart="'+str(i)+'" />',  file=routes)
        vehNr +=4
        HDV+=4
print("""
</routes>""", file=routes)
routes.close()
```

Detectors are added in additional file which is shown below:



```xml

<additionals>
    <!--EW-->
    <laneAreaDetector id="D2_EW" lane="E4_0" pos="0" length="200" freq="10" file="NUL" timeThreshold="1" />
    <!--tl="<TRAFFIC_LIGHT_ID>"   #timethreshold=how much time has to pass until a ceh is recognised as halted-->
    <e1Detector id="D1_EW" lane="E4_0" pos="10" freq="1" file="demo_det_out" />

    <!--WE-->
    <laneAreaDetector id="D2_WE" lane="E3_0" pos="0" length="200" freq="10" file="NUL" timeThreshold="1" />
    <!--tl="<TRAFFIC_LIGHT_ID>"   #timethreshold=how much time has to pass until a ceh is recognised as halted-->
    <e1Detector id="D1_WE" lane="E3_0" pos="10" freq="1" file="demo_det_out" />

    <!--NS-->
    <laneAreaDetector id="D2_NS" lane="E1_0" pos="0" length="200" freq="10" file="NUL" timeThreshold="1" />
    <!--tl="<TRAFFIC_LIGHT_ID>"   #timethreshold=how much time has to pass until a ceh is recognised as halted-->
    <e1Detector id="D1_NS" lane="E1_0" pos="10" freq="1" file="demo_det_out" />

    <!--SN-->
    <laneAreaDetector id="D2_SN" lane="E2_0" pos="0" length="200" freq="10" file="NUL" timeThreshold="1" />
    <!--tl="<TRAFFIC_LIGHT_ID>"   #timethreshold=how much time has to pass until a ceh is recognised as halted-->
    <e1Detector id="D1_SN" lane="E2_0" pos="10" freq="1" file="demo_det_out" />

    <!-- Detectors at the junction -->
    <e1Detector id="D3_EW" lane="E4_0" pos="200" freq="1" file="demo_det_out" />
    <e1Detector id="D3_WE" lane="E3_0" pos="200" freq="1" file="demo_det_out" />
    <e1Detector id="D3_NS" lane="E1_0" pos="200" freq="1" file="demo_det_out" />
    <e1Detector id="D3_SN" lane="E2_0" pos="200" freq="1" file="demo_det_out" />


</additionals>

```

The main TraCI python code for bid-based signal control is mentioned below.

```python
@author: Pranik

"""



import os

import sys

import optparse

import random

import numpy as np

import pandas as pd



# we need to import some python modules from the $SUMO_HOME/tools directory

if 'SUMO_HOME' in os.environ:

    tools = os.path.join(os.environ['SUMO_HOME'], 'tools')

    sys.path.append(tools)

else:

    sys.exit("please declare environment variable 'SUMO_HOME'")




from sumolib import checkBinary  # Checks for the binary in environ vars

import traci
```

```
def get_options():
    opt_parser = optparse.OptionParser()
    opt_parser.add_option("--nogui", action="store_true",
                         default=False, help="run the commandline version of sumo")
    options, args = opt_parser.parse_args()
    return options

# contains TraCI control loop
def run():
    traci.trafficlight.setProgram("N0","1")
    #RANDOM SEED
    np.random.seed(10)
    #Recording the time step of the simulation
    total_bid_EW=float(0)
    total_bid_WE=float(0)
    total_bid_NS=float(0)
    total_bid_SN=float(0)
    veh_done=[]
    vehSN_=None #previous vehicles for D3 detectors
    vehNS_=None
    vehEW_=None
    vehWE_=None
    D2_output=pd.DataFrame(columns={"step","phase","queueNS","queueSN","queueWE","queueEW","bidEW/WE","bidNS/SN"})
    D3_EW_output=pd.DataFrame(columns={"step","veh","type","delay","bid","queue"})
    D3_WE_output=pd.DataFrame(columns={"step","veh","type","delay","bid","queue"})
    D3_NS_output=pd.DataFrame(columns={"step","veh","type","delay","bid","queue"})
    D3_SN_output=pd.DataFrame(columns={"step","veh","type","delay","bid","queue"})
    vehEW_index=0 #index for output of D3 detectors
    vehWE_index=0
    vehNS_index=0
    vehSN_index=0
    next_step=0
    previous_signal=None
    max_count=0
    step = 0


    while traci.simulation.getMinExpectedNumber() > 0:
```

```
traci.simulationStep()
print(step)
step+=1

def PosNormal(mean, sigma):
    x = int(np.random.normal(mean,sigma,1))
    return(x if x>=0 else PosNormal(mean,sigma))

def get_bid_data(veh, detector):
    bid=PosNormal(0, 5)
    queue=traci.lanearea.getJamLengthVehicle(detector)
    current_signal=traci.trafficlight.getRedYellowGreenState("N0")
    final_bid=bid + 0*queue
    traci.vehicle.setParameter(veh, "Bid", final_bid)
    traci.vehicle.setParameter(veh, "queue", queue)
    return final_bid, current_signal

def print_bid_table():
    print("---------------------")
    print("total_bid_EW:", total_bid_EW)
    print("total_bid_WE:", total_bid_WE)
    print("total_bid_NS:", total_bid_NS)
    print("total_bid_SN:", total_bid_SN)
    print("---------------------")

def get_bid_float(bid_str):
    #bid=bid_str[1:len(bid_str)-1]
    bid=float(bid_str)
    return bid
"""
Getting bid for EW
"""
veh_EW=traci.inductionloop.getLastStepVehicleIDs("D1_EW")
if veh_EW!=():
    veh_EW=list(veh_EW)[0]
    veh_type=traci.vehicle.getTypeID(veh_EW)
    if veh_type == "CV" and veh_EW not in veh_done:
        detector="D2_EW"
        final_bid, current_signal = get_bid_data(veh_EW, detector)
        total_bid_EW+=final_bid
        veh_done.append(veh_EW)
"""
Getting bid for WE
"""
veh_WE=traci.inductionloop.getLastStepVehicleIDs("D1_WE")
if veh_WE!=():
    veh_WE=list(veh_WE)[0]
    veh_type=traci.vehicle.getTypeID(veh_WE)
    if veh_type == "CV" and veh_WE not in veh_done:
        detector="D2_WE"
        final_bid, current_signal = get_bid_data(veh_WE, detector)
        total_bid_WE+=final_bid
        veh_done.append(veh_WE)
"""
Getting bid for NS
"""
veh_NS=traci.inductionloop.getLastStepVehicleIDs("D1_NS")
if veh_NS!=():
    veh_NS=list(veh_NS)[0]
    veh_type=traci.vehicle.getTypeID(veh_NS)
    if veh_type == "CV" and veh_NS not in veh_done:
        detector="D2_NS"
        final_bid, current_signal = get_bid_data(veh_NS, detector)
        total_bid_NS+=final_bid
        veh_done.append(veh_NS)
"""
Getting bid for SN
"""
veh_SN=traci.inductionloop.getLastStepVehicleIDs("D1_SN")
if veh_SN!=():
    veh_SN=list(veh_SN)[0]
    veh_type=traci.vehicle.getTypeID(veh_SN)
    if veh_type == "CV" and veh_SN not in veh_done:
        detector="D2_NS"
        final_bid, current_signal = get_bid_data(veh_SN, detector)
        total_bid_SN+=final_bid
        veh_done.append(veh_SN)

""" BID Clearance House """
Total_bid_EW=total_bid_EW + total_bid_WE
Total_bid_NS=total_bid_NS + total_bid_SN

current_signal=traci.trafficlight.getRedYellowGreenState("N0")
if Total_bid_EW > Total_bid_NS and step>=next_step:
    if current_signal=="yyyy":
        traci.trafficlight.setRedYellowGreenState("N0","rGrG")
        next_step=step+15

    if current_signal=="GrGr":
        traci.trafficlight.setRedYellowGreenState("N0","yyyy")
        next_step=step+3

elif Total_bid_NS > Total_bid_EW and step>=next_step:
    if current_signal=="yyyy":
        traci.trafficlight.setRedYellowGreenState("N0","GrGr")
        next_step=step+15

    if current_signal=="rGrG":
        traci.trafficlight.setRedYellowGreenState("N0","yyyy")
        next_step=step+3
#MAX LIMIT
if current_signal==previous_signal:
    max_count+=1
    if max_count>100 and current_signal=="rGrG":
        traci.trafficlight.setRedYellowGreenState("N0","GrGr")
    if max_count>100 and current_signal=="GrGr":
        traci.trafficlight.setRedYellowGreenState("N0","rGrG")
if current_signal!=previous_signal:
    max_count=0
previous_signal=current_signal
"""Clearning House : clearing the bid of the CVs passing through the intersection and extracting DATA of cars leaving"""
#EW
vehEW=traci.inductionloop.getLastStepVehicleIDs("D3_EW")
if vehEW!=():
    vehEW=list(vehEW)[0]
    if vehEW!=vehEW_:
        veh_type=traci.vehicle.getTypeID(vehEW)
        veh_delay=traci.vehicle.getTimeLoss(vehEW)
        #extracting data from D3 detector
        D3_EW_output.loc[vehEW_index,"step"]=step
        D3_EW_output.loc[vehEW_index,"veh"]=vehEW
        D3_EW_output.loc[vehEW_index,"type"]=veh_type
        D3_EW_output.loc[vehEW_index,"delay"]=veh_delay
        if veh_type == "CV" and vehEW!=vehEW_:
            bid=traci.vehicle.getParameter(vehEW, "Bid")
            bid=get_bid_float(bid)
            total_bid_EW-=bid
            D3_EW_output.loc[vehEW_index,"bid"]=bid
            queue=traci.vehicle.getParameter(vehEW, "queue")
            D3_EW_output.loc[vehEW_index,"queue"]=queue
        vehEW_=vehEW
        vehEW_index+=1
#WE
vehWE=traci.inductionloop.getLastStepVehicleIDs("D3_WE")
if vehWE!=():
    vehWE=list(vehWE)[0]
    if vehWE!=vehWE_:
        veh_type=traci.vehicle.getTypeID(vehWE)
        veh_delay=traci.vehicle.getTimeLoss(vehWE)
        #extracting data from D3 detector
        D3_WE_output.loc[vehWE_index,"step"]=step
        D3_WE_output.loc[vehWE_index,"veh"]=vehWE
        D3_WE_output.loc[vehWE_index,"type"]=veh_type
        D3_WE_output.loc[vehWE_index,"delay"]=veh_delay
        if veh_type == "CV" and vehWE!=vehWE_:
            bid=traci.vehicle.getParameter(vehWE, "Bid")
            bid=get_bid_float(bid)
            total_bid_WE-=bid
            D3_WE_output.loc[vehWE_index,"bid"]=bid
            queue=traci.vehicle.getParameter(vehWE, "queue")
            D3_WE_output.loc[vehWE_index,"queue"]=queue
        vehWE_=vehWE
        vehWE_index+=1
#NS
vehNS=traci.inductionloop.getLastStepVehicleIDs("D3_NS")
if vehNS!=():
    vehNS=list(vehNS)[0]
    if vehNS!=vehNS_:
        veh_type=traci.vehicle.getTypeID(vehNS)
        veh_delay=traci.vehicle.getTimeLoss(vehNS)
        #extracting data from D3 detector
        D3_NS_output.loc[vehNS_index,"step"]=step
        D3_NS_output.loc[vehNS_index,"veh"]=vehNS
        D3_NS_output.loc[vehNS_index,"type"]=veh_type
        D3_NS_output.loc[vehNS_index,"delay"]=veh_delay
        if veh_type == "CV" and vehNS!=vehNS_:
            bid=traci.vehicle.getParameter(vehNS, "Bid")
            bid=get_bid_float(bid)
             #print("total_bid_NS", total_bid_NS)
            total_bid_NS-=bid
            D3_NS_output.loc[vehNS_index,"bid"]=bid
            queue=traci.vehicle.getParameter(vehNS, "queue")
            D3_NS_output.loc[vehNS_index,"queue"]=queue
        vehNS_=vehNS
        vehNS_index+=1
#SN
vehSN=traci.inductionloop.getLastStepVehicleIDs("D3_SN")
if vehSN!=():
    vehSN=list(vehSN)[0]
    if vehSN!=vehSN_:
        veh_type=traci.vehicle.getTypeID(vehSN)
        veh_delay=traci.vehicle.getTimeLoss(vehSN)
        #extracting data from D3 detector
        D3_SN_output.loc[vehSN_index,"step"]=step
        D3_SN_output.loc[vehSN_index,"veh"]=vehSN
        D3_SN_output.loc[vehSN_index,"type"]=veh_type
        D3_SN_output.loc[vehSN_index,"delay"]=veh_delay
        if veh_type == "CV" and vehSN!=vehSN_:
            bid=traci.vehicle.getParameter(vehSN, "Bid")
            bid=get_bid_float(bid)
```

```
                    total_bid_SN-=bid
                    D3_SN_output.loc[vehSN_index,"bid"]=bid
                    queue=traci.vehicle.getParameter(vehSN, "queue")
                    D3_SN_output.loc[vehSN_index,"queue"]=queue
                vehSN_=vehSN
                vehSN_index+=1

        """D2 output"""
        current_signal=traci.trafficlight.getRedYellowGreenState("N0")
        queueEW=traci.lanearea.getJamLengthVehicle("D2_EW")
        queueWE=traci.lanearea.getJamLengthVehicle("D2_WE")
        queueNS=traci.lanearea.getJamLengthVehicle("D2_NS")
        queueSN=traci.lanearea.getJamLengthVehicle("D2_SN")
        D2_output.loc[step,"step"]=step
        D2_output.loc[step,"phase"]=current_signal
        D2_output.loc[step,"queueEW"]=queueEW
        D2_output.loc[step,"queueWE"]=queueWE
        D2_output.loc[step,"queueNS"]=queueNS
        D2_output.loc[step,"queueSN"]=queueSN
        print(Total_bid_EW)
        D2_output.loc[step,"bidEW/WE"]=Total_bid_EW
        D2_output.loc[step,"bidNS/SN"]=Total_bid_NS

        if step == 1800:
            D2_output.to_csv("data\D2_75_0q.csv")
            D3_WE_output.to_csv("data\D3_WE_75_0q.csv")
            D3_EW_output.to_csv("data\D3_EW_75_0q.csv")
            D3_NS_output.to_csv("data\D3_NS_75_0q.csv")
            D3_SN_output.to_csv("data\D3_SN_75_0q.csv")
    traci.close()
    sys.stdout.flush()
# main entry point
if __name__ == "__main__":
    options = get_options()

    # check binary
    if options.nogui:
        sumoBinary = checkBinary('sumo')
```

```
else:
    sumoBinary = checkBinary('sumo-gui')

# traci starts sumo as a subprocess and then this script connects and runs
traci.start([sumoBinary, "-c", "main.sumo.cfg",
                         "--tripinfo-output", "tripinfo.xml"]) #<--------------
run()
```