Team Control Number

22091106

Problem Chosen

A

2022

IMMC

Summary Sheet

Nowadays, with the rapid development of autonomous driving technology, the establishment of an autonomous driving system will no longer be an absurd idea. In this system, connections can be built between the autonomous driving cars, the smart lampposts installed with sensors and communication units and the cloud server. Therefore we need to carry out a plan which will make the transportation system easier and create an indicator system to evaluate the smart lamppost modification plans.

In the first model, we find out the factors that are related to evaluating a lamppost modification plan. We mainly discuss how to calculate the index which is affected by cost, coverage rate of LiDAR and the importance of each road. Afterwards, we make some plans about the measure of coverage rate in different situations.

In the second model, our goal is to establish an overall optimal configuration of all lampposts within a certain area. Instead of simply simulating all measures, we took a more natural Q-learning approach under the thought of greedy algorithms. The machine-learning process, which relied merely on the mathematical index of a given distribution as shown in the first model, gradually narrowed the result till a fixed distribution. In order to confirm the accuracy of the algorithm, we randomly altered certain sets of lampposts and withdrew the consequent changes in variables. After that, we chose the uniformity of lamppost points as the independent variable and conducted a sensitivity analysis, which, after eliminating certain anomalous data, proved that the results were accurate enough. Out of considerations of dual confirmation, we also checked that the results turned out to be optimal.We did a sensitivity analysis on the model, and found some issues where it cannot evaluate correctly.

Last, we evaluated the plan using the model built in the first part. With the help of Python packages *Shapely* and *Matplotlib*, we managed to calculate the coverage area and coverage rate of our plan in the second part. Probable explanations are raised, and we developed a dynamic model to solve the problem. We also made a dynamic model to simulate the process of WiFi connection using the Hungarian Algorithm.

Smart Lamppost Deployment

Team#22091106

November 22, 2021

Contents

1 Introduction **1**

1.1 Background 1

1.2 Problem Restatement 1

1.3 General Assumptions 2

2 Model 1: Evaluation Model **3**

2.1 Model Overview 3

2.2 Assumptions 3

2.3 Variables and Constants 4

2.4 Determining the Factors 4

3 Model 2: Modification Plan **5**

3.1 Model Overview 5

3.2 Definition and Assumptions 5

3.3 Model Calculation 6

3.4 Variables and Calculations 6

3.5 Result 8

3.6 Sensitivity Analysis 9

4 Evaluation of Our Own Plan **11**

4.1 Problem Overview 11

4.2 Calculation of the Evaluation Index 11

4.3 Verify Our Model 11

4.4 Result Analysis 13

5 Model 3: Improved Model **13**

5.1 Model Overview 13

5.2 Assumptions 13

5.3 Notation 14

5.4 Materializing the Traffic 14

5.5 Strength and Weaknesses 18

Appendix **19**

References **20**

1 Introduction

1.1 Background

Nowadays, with the rapid development of autonomous driving technology, building an automatic driving system will no longer be a ridiculous thought. Based on recent technology, autonomous driving technology has begun to move from partial driving at the L2 level to conditional driving automation at the L3 level.

According to the degree of automation of the vehicle[1], when the system cannot withstand the working conditions, the driver needs to take over the malfunctioning vehicle. After activating the automatic driving system, the vehicle itself can complete tasks such as steering, acceleration, deceleration, and reaction. The status detection and reaction are carried out under the operating conditions specified by the automatic driving system.

In order to facilitate vehicles which will better implement the L3 level autonomous driving tasks, using smart lampposts as the main representative of the smart roadside infrastructure. By refitting existing ordinary lampposts into smart lampposts equipped with sensors and communication units. They can collect road data through sensors and upload them onto the cloud server, and then download to the original lamppost or share it with other ones after the server completes the calculations. The autonomous driving cars can obtain the road data by communicating with neighboring smart lampposts.

1.2 Problem Restatement

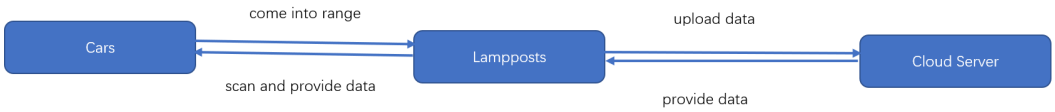

Fig. 1 Relationship between the basic factors

In this model, we actually need to have some detailed information about the whole transportation system in the area. In addition, we need to know about the driving pattern of the cars to make further discoveries about the autonomous driving system. In Fig. 1, we can clearly discover the relationship between the car, the lamppost and the cloud server.

So the problem is divided into 3 main parts:

- Build a framework for evaluating the smart lamppost modification plans based on cost and coverage of the road by the sensor and the WiFi communication.
- Give a plan for smart lamppost modification in an area of any city according to the data given.[2]
- Evaluate the lamppost modification plan based on our model and find the strengths and weaknesses of our model.

1.3 General Assumptions

- **All the cars have the same scale.**

  We assume that all the cars can be seen as a car which is 4.8m in length and 1.6m in width. This can flatten the neighborhood which we take into consideration.

- **The impact of buildings is not taken into consideration.**

  Since the distance between two roads is far from the detection zone, the buildings will not affect the detection of the road.

- **Vehicles drive on the left side according to the standard of Hong Kong.**

- **Neighborhood intersection is not taken into consideration.**

- **Most of the calculation is completed on the client side.**

  There are different types of car on the road. Since the cloud side can't afford such large amount of calculation, we assume that the most of it is completed on the client side.

- **The information that the cars need is of the same number at any time of the process.**

  In our dynamic model, the amount of information that each car requires is assumed to be a constant, which, in real life, varies according to the inherent attributions of the car.

2 Model 1: Evaluation Model

2.1 Model Overview

In this model, we will establish an indicator framework for the purpose of evaluating smart lamppost modification plans. By comparing the weight of different road types according to the cars' speed, acceleration, deceleration and reaction, we can clearly get the relationships of these areas. Basic road types include normal roads and crossroads. In this model, crossroads is defined as a circle with a radius of 50m, centering at the middle of the crossing.

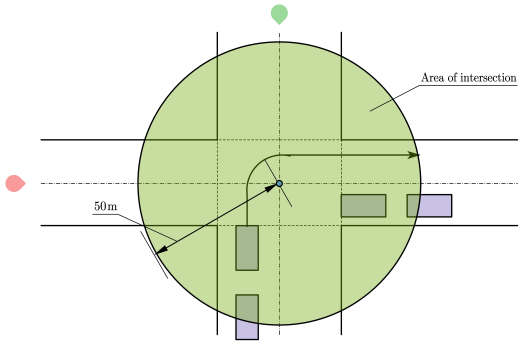

Fig. 2 Driving Pattern in Crossroads

2.2 Assumptions

- **All the destinations can be detected and can be abstracted as a dot.**

  To flatten the driving routine, we assume that all the destinations are only dots. When the car which has the shape of a car reach its destination, it will disappear immediately.

- **The car will update its newest driving data at any time of the driving process.**

  To make sure that we can supervise the driving process and the location of cars connected to LiDAR and WiFi, we assume that the data will be updated at any time of the day.

2.3 Variables and Constants

Symbol	Definition
S	Evaluating Index
P	Overall Cost
a_1	The Coverage Rate of Sensors on Crossroads
a_2	The Coverage Rate of Sensors on Normal Roads
b_1	The Coverage Rate of WiFi Communication on Crossroads
b_2	The Coverage Rate of WiFi Communication on Normal Roads
k_1	Weight of Crossroads
k_2	Weight of Normal Roads
b	The rate of the most cars that can be connected to WiFi
k_3	Weight of b

2.4 Determining the Factors

In this model, we focus on the coverage rate of LiDAR and WiFi on different road type. For example, we describe a_1 as $\frac{S_{\text{Coverage of Sensors on Crossroads}}}{S_{\text{all}}}$. In the same way can we define a_2, b_1, b_2.

When an autonomous driving cars are driving, the passengers inside usually thinks more about their safety when passing crossroads. This shows that the crossroads are more important than normal roads in our framework, so we can infer that: $k_1 \gg k_2$

Therefore by defining the weight of coverage rate in different areas, we find out a simple formula describing the effectiveness of the system:

$$S = \frac{k_1(a_1 + b_1)^3 + k_2(a_2 + b_2) + k_3 b}{P}$$

$$(k_1 > k_3 > k_2)$$

3 Model 2: Modification Plan

3.1 Model Overview

Now that we have established a basic framework of evaluating modification plans, we need to make sure that a plan can indeed be carried out in any selected city which have the economic ability to realize autonomous driving systems. In this model, we will focus on the smart lamppost modification plan based on the transportation system in Wanzai Area,iHong Kong.[2] (See Appendix) We can abstract this map into a rectangular coordinate system. In Fig. 3, the red line refers to the road in this area and the blue dots refers to the lampposts that can be installed with LiDAR sensors and WiFi Access Points. Having this map, we can easily demonstrate the effect of smart lampposts on nearby cars.

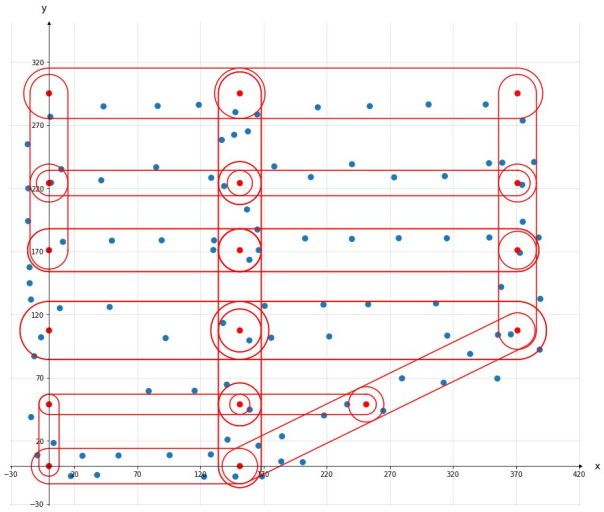

Fig. 3 Map of Wanzai Area

3.2 Definition and Assumptions

Based on the traffic rules and the geographic conditions in Wanzai Area, we make the following assumptions.

1, We assume that an autonomous driving need no time to turn left and 5 seconds to turn right. The number comes from the calculation according to the average width of roads and the highest speed to change direction safely in normal conditions. This means that when a car turn right, it will disappear immediately and appear at the end of the line in this road 5 seconds later. In this process, the car will exert no impact neither on other cars nor on the queuing line.

2, After searching the traffic rules in this area, we find that Gaoshida Avenue has a speed limit of 70km/h (19.5m/s) and other roads have a speed limit of 50km/h (14.0m/s). To make sure that our model is safe and can fit the traffic rules, we assume that all the cars

drive on normal roads at a constant speed of 50km/h (14.0m/s).

3, For we mainly focus on the driving condition of cars and their relationship with the smart lampposts, pedestrians are ignored in this model.

4, Through our calculation, the stopping distance turns out to be at least 16m. To ensure that our plan can make whole use of the technology, we need to make sure that LiDAR can cover all the crossroads.

5, Because a single WiFi Access Point can only get connected to 4 cars at a time, we need to consider the cars which are not connected to WiFi technology. Now we assume that the cars drive a constant speed and will get connected to WiFi as soon as the last car leaves the area.

So now let we assume that we are in an autonomous driving car. If the car is running normally, it will drive at a constant speed when on normal roads. When it receives the information provided by WiFi technology, then it will estimate whether it's turning or driving, whether it's going to meet red lights and when it's going to change its direction.

3.3 Model Calculation

To get further understanding of the transportation system in this area, we use Q-learning to summarize our way of distributing and find out the best modification plan.

Q-learning[4] is an off policy reinforcement learning algorithm that seeks to find the best action to take given the current state. It doesn't need any policies because the Q-learning function learns from actions that are outside the current policy by taking different actions randomly. More specifically, Q-learning seeks to learn a policy that maximizes the total reward. Therefore, we can use Q-learning to find the location of smart lampposts installed with WiFi and LiDAR technology that best fits the condition in the selected area.

3.4 Variables and Calculations

Symbol	Definition
r	Road type
d	device
L_1	Lampposts with only LiDAR
L_2	Lampposts with only WiFi
L_3	Lampposts with both

Algorithm 1 Find the best modification plan.

def Get_State(Distribution)
Cost = $5000 \cdot L_1 + 3000 \cdot L_2 + 10000 \cdot L_3$
# Formula of calculating coverage: $\frac{S_{\text{road covered}}}{S_{\text{roads}}}$

$$\sum_{\substack{\text{roadtype} \overset{\text{def}}{=\!=} \mathrm{r} \in \{\text{Intersection}, \text{Normal}\} \\ \text{device} \overset{\text{def}}{=\!=} \mathrm{d} \in \{\text{LiDAR}, \text{WiFi}\}}} \frac{\mathrm{r} \cdot \mathrm{d}}{\text{Cost}} = \frac{k_1 (a_1 + b_1) + k_2 (a_2 + b_2)}{P}$$

return State
end def
L = Set of all SmartLamps
done = **False**
initialize q_table, LEARNING_RATE and DISCOUNT
while not done **do**
  update q_table
end while
for lamp in L **do**
  while in attempt to change the configuration of lamp **do**
    if Get_State(new_distribution) > Get_State(current_distribution) **then**
      change distribution
    else
      remain
    end if
  end while
end for
previous loop → find q_table[max_distribution] and corresponding action
current_distribution = q_table[current_state + action]
new_distribution = (1 - LEARNING_RATE) * current_distribution +
LEARNING_RATE× (Δ Get_State + DISCOUNT * max_distribution)
q_table[current_state + action] = new_distribution
conduct the loop continuously until the previous loop doesn't alter anything:
done = **True**

3.5 Result

According to these rules and the calculation ideas we have put foward, we finally drew a complete figure of our plan, which is shown in the Fig. 4 below.

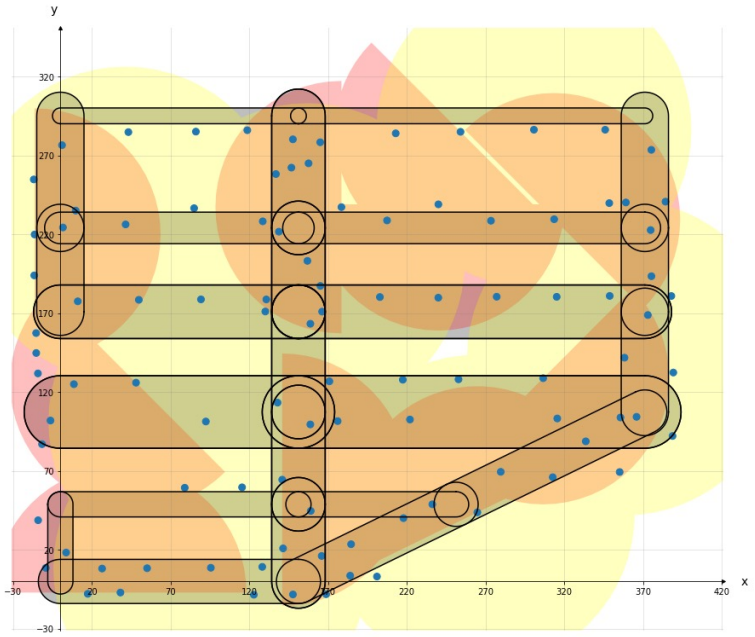

Fig. 4 Final Result of Distribution

In Fig. 4, the yellow part refers to the WiFi-covered area while the red part refers to the LiDAR-covered part. We can see that most of the space in this area is now covered by Wifi, so cars driving in this area can get connected to the cloud server and get the data provided to make better driving plans. Because WiFi technology has a wider range of coverage than LiDAR technology, there are parts which are WiFi-covered but not LiDAR-covered. But since we have assumed that cars drive at a constant speed when they're not LiDAR-covered part, the impact can be ignored.

In all we need to distribute 16 smart lampposts. There are 9 lampposts which only need to be installed with LiDAR sensors, 6 lampposts which only need to be installed with WiFi access points and only 1 lamppost to be installed with both of them. There position coordinates are shown in the two charts below.

Table 1: LiDAR Lamppost Coordinates

	xLiDAR	yLiDAR	Angle
1	48.0341	126.007	135
2	-16.3686	219.959	270
3	140.853	64.5057	270
4	178.457	237.312	90
5	240.017	239.083	180
6	313.536	229.637	315
7	306.539	128.857	225
8	38.1314	-7.04044	0
9	254.081	285.061	135
10	264.77	43.785	26

Table 2: WiFi Lamppost Coordinates

	xWiFi	yWiFi
1	41.427	226.357
2	92.3732	101.392
3	95.542	8.66328
4	156.851	203.248
5	264.77	43.785
6	300.676	286.392
7	358.18	141.86

After calculating the cost, we find that we only need to spend $73000 on our smart lamppost modification plan.

3.6 Sensitivity Analysis

In this part, we will discuss the sensitivity of our plan and find out whether our plan is useful enough for the city and everyday transportation.

After calculation, we get the formula describing the sensitivity of our model.

$$I_{sensitivity} = \frac{\frac{\Delta U}{U}}{\frac{\Delta S}{S}}$$

As mentioned in the modification plan, we get a detailed picture of the area we study and the position of each lamppost we need to distribute. By changing the position coordinates of LiDAR sensors and WiFi Access Point randomly, we can see the change of variables mentioned in Evaluation Model. Later we get rid of some abnormal data and calculate the variance of our data: 16.7357874.

In this formula, U refers to the uniformity of this model.To express the distribution of smart lampposts in a mathematical way, we used the "distribution index"U which is defined as following:[3]

$$U = \frac{\overline{d_{\min}}}{\mathrm{R}} = \frac{\left(\text{the n minimums of the } \frac{n(n-1)}{2} \text{ distances between n lampposts } /n\right)}{2 \cdot \sqrt{area \cdot n^{-1} \cdot \pi^{-1}}}$$

(In this model, n refers to the number of the lampposts.)

In theory, U is a real between 0 and 1. When the U is bigger, the distribution is more average.

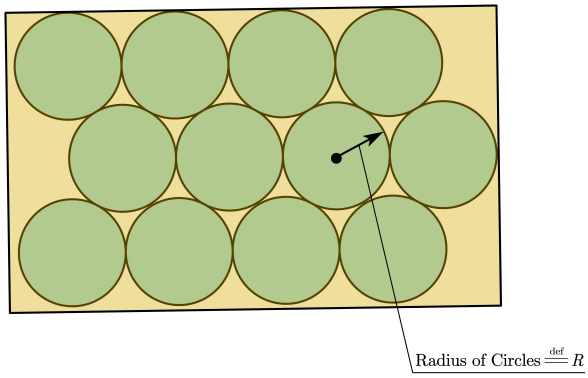

Fig. 5 Describing Uniformity

Finally we get the number of U in our model: 4.0065. This is a reasonable number and it shows that our model is robust and reasonable. It can apply to various changes and can make better distribution plans.

4 Evaluation of Our Own Plan

4.1 Problem Overview

In the last section, we have analyzed the sensitivity of our plan. In this part, we will calculate the coverage area and the coverage rate of our plan accurately. Later, we use our index to evaluate our plan. By getting the number and comparing it with other plans in which some variables are changed randomly, we will find the strengths and weaknesses of our plan. Later we will show what improvements we decide to make to our model and what we can do with the development of the automatic driving system. A dynamic model is developed to solve the problem of inaccurate evaluation.

4.2 Calculation of the Evaluation Index

We calculate the evaluation index of different conditions as we change the location coordinates of lampposts and the angle of LiDAR sensors to find the rise and fall of its effectiveness which is shown by the evaluation index.

After calculation, we find out that the evaluation index of our plan is 0.7233. However, after changing the factors, the number ranges from 0.54 to 0.71. This shows that our model has great effectiveness and can best fit the traffic system in Hong Kong now. Also, we discover that the change in WiFi location will let the number have a sharper decrease, so we can infer that WiFi location will pose a greater impact on the modification system. If we want to make further improvement, we can focus on the change in WiFi distribution.

4.3 Verify Our Model

To make sure that our model can apply to everyday traffic routine, we use C++ to simulate the driving process and calculate the connection rate of WiFi.

In the simulation process, we consider each road block as an array of cars. In each round of the update process, we begin the operation of each road from the first car in its array till the last. Notice that in our dynamic updating process, each car, except from the first in the array, possesses a unique action dependent on its predecessor. In short, the car evaluates its capability of reaching the next intersection before the red light and performs correspondent deceleration or acceleration while controlling the safe distance between the two cars at the same time, the mathematical formulas of which are shown in the code below. As for the first car in the array, it similarly decides its action, and randomly decides to turn to the next road if it ultimately reaches the intersection with the green light on.

Secondly, we use bipartite graphs to simulate the connection process of WiFi.When calculating the rate of cars connected into the WiFi system (b), we can use an algorithm which matches WiFi lamps with cars to find how many cars can be connected into the WiFi system.

Algorithm 2 Simulation of the driving condition.

```
def update(test):
time += 0.5 # updating once every half a second
for k in cars # run through all the cars do
  if car No. k is going to start turning at this time then
    pick a random direction at its crossroad for the car to go to
    decide the time the car spends on this turning (depending on the direction of the
    turning)
  end if
  if car No.k is already turning at this time then
    it continues turning regardless of traffic lights
  else
    # car No.k is in the middle of a road
    calculate the location of this car at this time
    find the coming traffic light
    if this car cannot pass the traffic light then
      calculate the minimum deceleration distance according to the original speed of
      the car and its maximum deceleration rate
      if the distance between the car and the light is no longer than this distance then
        it decelerates in its maximum decelerate rate
      else
        calculates its maximum speed and then accelerate to this speed in the maximum
        acceleration rate and then decelerate its speed to 0 (and just stops at the red
        light)
      end if
    else
      if the distance between the car and the traffic light is no longer than the distance
      the car goes in half a second then
        # it will turn in the coming half second
        label it as it would turn in the next half a second
      end if
    end if
  end if
end for
end def
```

4.4 Result Analysis

Strength

- **Our model has strong flexibility.**

  Our models are generic.Except from the necessary road map data, our model completely depends on the internal algorithms of the system, thus it can be applied to various conditions.

- **Efficiency**

  In model 2 we modified the highly-efficient Q-learning algorithm, which quickly narrows the range until it gets the optimal result.

Weaknesses and Expectation

- **We have limitation when facing discretization of variables.**

  The standard Q-learning algorithm applies only to discrete action and state processes. When there are less than 30 lampposts, it can no longer find the best data largely due to the cause of dimenssionality.[5]

- **There is difference between our model and reality.**

  Nowadays, the WiFi technology is not so efficient and the calculation on the client side doesn't have such a speed as assumed in our model. Our model should make improvements. In addition, we should always pay attention to new technology as the technology is developing fast.

5 Model 3: Improved Model

5.1 Model Overview

When analyzing the strengths and weakness of our model, we mentioned that we do not have a dynamic model to make our calculations more accurate. Now, we try to make it more accurate by using C++.

5.2 Assumptions

- $\forall L, V_{M,L} \overset{\text{always}}{=\!=} 50\text{km/h}$

  By searching for data on websites[6], we found out that all roads of such scale in our model have a speed limit of 50km/h*, for given the fact that only the bottom of Gloucester Road is connected with the rest of the map.*

- **The car inflow and outflow of the selected area is always at an equilibrium.**

  Or else, $N_{C,L}$ is either consistently decreasing or increasing, which, given the isolation and the areal limitations of the model, is apparently unreasonable.

5.3 Notation

Symbol	Definition
$\tau_{L_1 \to L_2, R}$	the red-light interval of road L_1 at its intersection with road L_2
$\tau_{L_1 \to L_2, G}$	the green-light interval of road L_1 at its intersection with road L_2
W_L	the width of road L
T_L	the duration of a total light period
$V_{M,L}$	the maximum velocity of cars on road L
$N_{C,L}$	the number of cars on road L

5.4 Materializing the Traffic

Firstly, we set up several functions concerning cars, roads, and traffic lights using C++. As for the dynamic process, the real-time performance of cars is simulated Because all the roads are of approximately the same width, deriving the equation above the first assumption, we can get the following relationship:

$$\forall I, T_{I,turn} = \begin{cases} 0.5, turn = \text{left} \\ 1.5, turn = \text{straight} \\ 5, turn = \text{right or turn around} \end{cases}$$

$$\frac{\tau_{L_1 \to L_2, G}}{\tau_{L_2 \to L_1, G}} \propto \frac{W_{L_1}}{W_{L_2}} \tag{5.1}$$

$$\forall L_1, L_2 \text{ as lights}, \tau_{L_1 \to L_2, G} + \tau_{L_1 \to L_2, R} \overset{\text{always}}{=\!=} T_L \tag{5.2}$$

In this relationship, (5.2) apparently maintains the numerical proportion of traffic light intervals as shown in (5.1), and it is also beneficial to our discrete model, since the timelines of different traffic lights can be easily synchronized.

In our model, we assume that the time of cycle of red lights and green lights is 100 seconds in total. In addition, we introduced the concept of 'yellow lights' so as to offset the inaccuracy of updating the model discretely: a certain time is given for each vehicle to either decelerate or accelerate at its maximum.

When it comes to deceleration, we have the constants and variables below

Symbol	Definition
v_0	Initial Velocity
a	Acceleration
x	Minimum Braking Displacement
t	Time
θ	the Angle of the Direction of the Car's Displacement and Horizontal Direction
l	Displacement
v	Velocity

So we can get the minimum breaking displacement is:

$$x = \frac{v^2}{2a}$$

If a refers to the largest acceleration, the displacement in that period is:

$$\mathrm{d}v = a\mathrm{d}t$$
$$\mathrm{d}l = v\mathrm{d}t$$

So we can infer that:

$$\Delta x = l \sin\theta$$
$$\Delta y = l \cos\theta.$$

Secondly, for different states of the output of the dynamic model, we use the Hungarian algorithm to get the optimal connection scheme of cars and tests whether it is optimal. The process is shown in the algorithm below.

We construct a bipartite graph G($V_1$4 ,V_2;E), where V_1 refers to the set of lampposts, and V_2 refers to the set of all on-road vehicles.

$$\forall u \in V_1, v \in V_2, uv \in E \Leftrightarrow \text{the distance between u and v} < 100\text{m}$$

Algorithm 3 Match lampposts with WiFi

```
# We consider that each of WiFi lampposts has four WiFi spots, with each one connecting
to one car.
lines[] # lines[i] refers to the spots that can be linked to the car No. i
used ← [] # judging whether a WiFi spot can be used
match ← [] # record the matching car of each WiFi spot
for i in lines[car] # all the spots that can connect with this car do
  if used[car][i] is 0 then
    used[car][i] ← 1
    if match[line[car][i]] is 0 or find(match[line[car][i]]) is 1:# this spot is not connected
    or this position can be spared then
      match[line[car][i]] ← car # record the pair
      return 1# this car can be connected
    end if
  end if
end for
return 0 # all the available spots cannot connect the car
while updating the information of the cars do
  set 4 spots for each WiFi lamp (each spot can link at most one car)
  for k in cars do
    for i in WiFi lamps do
      if the distance between WiFi lamp i and car k is no more than 100m then
        link an edge between car k and the 4 spots of WiFi lamp i
      end if
    end for
  end for
end while
for k in cars do
  ans←ans + find(k) # whether car No.k can be connected
end for
```

By getting the moving pattern of each car, we get the complete figure of this autonomous transportation system. The dots in different colors refer to moving patterns of different cars.

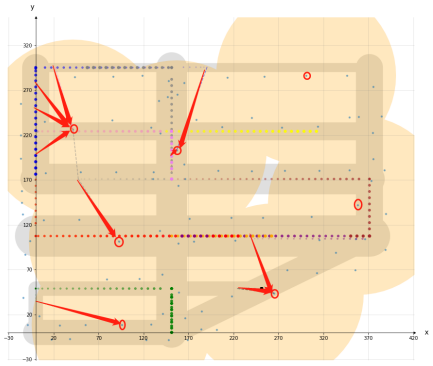

Fig. 6 $t = 0$

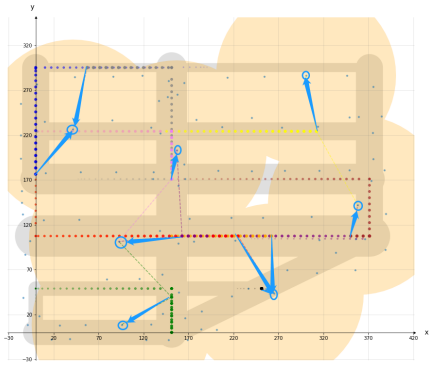

Fig. 7 $t = 60$

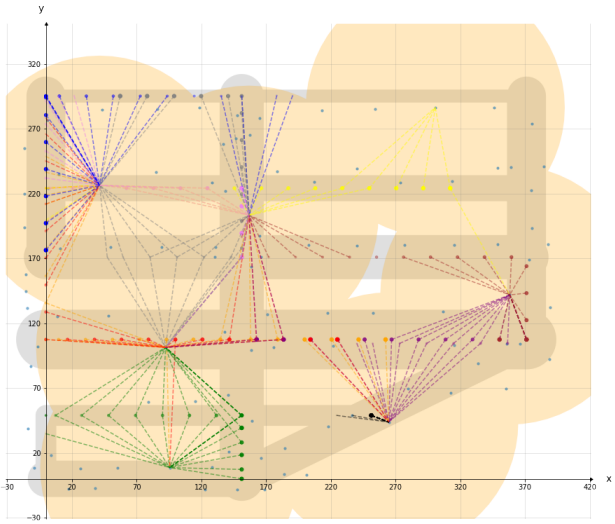

Fig. 8 Describing the dynamic pattern.

Finally, we manually optimize the installation under the conditions that meet the requirements which install the smallest set of all WiFi street light lines.

Based on the Hungarian Algorithm, we can also make some improvements. The car which comes near to the crossroad will have a greater chance to get connected to the WiFi Access Point.

5.5 Strength and Weaknesses

Strengths

- **Accuracy**

  Compared to Model 2, establishing a dynamic model adds realistic factors into the modification plan, such as red and green light. This will make our model more practical and more accurate. This is a great improvement based on the weaknesses of the last model.

- **Efficiency**

  In model 3, the intricate mechanism of the Hungarian algorithm helps us get the result within $\Theta(m+n)$, where m and n respectively refers to the number of vertexes and edges in the diagram.

Weaknesses

- **Complexity**

  Because our algorithm is too complicated, it usually takes a lot of time to receive the correct data. Also, the algorithm is sometimes unstable.

- **Limitation in WiFi Technology**

  The maximum throughput of a WiFi Access Point is 800 Mbps.In our dynamic model, the amount of information that each car requires is assumed to be a constant, which, in real life, varies according to the inherent attributions of the car. In this conception, the Hungarian algorithm will simply become a weighed two-dimensional optimization, optimizing car connection and information transferring at the same time. However, the model complexity and processing time will be greatly increased.

Appendix

Attachment 1

References

[1] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor J3016_201806, SAE International, https://www.sae.org/standards/content/j3016_201806/

[2] Hong Kong Geographical Map, GEOINFO MAP, https://www.map.gov.hk/gm/map/

[3] Check if 2D points are evenly distributed, MathWorks MATLAB Answers John D'Errico, https://www.mathworks.com/matlabcentral/answers/180860-check-if-2d-points-are-evenly-distributed

[4] Simple Reinforcement Learning: Q-learning, Andre Violante, https://towardsdatascience.com/simple-reinforcement-learning-Q-learning-fcddc4b6fe56

[5] Q-learning, Wikipedia, https://en.wikipedia.org/wiki/Q-learning

[6] Speed Limits in Hong Kong, Wikipedia, https://en.wikipedia.org/wiki/Speed_limits_in_Hong_Kong