Team Control Number

For office use only
T1 ________________
T2 ________________
T3 ________________
T4 ________________

1999759

Problem Chosen

C

For office use only
F1 ________________
F2 ________________
F3 ________________
F4 ________________

2019 Design for Taxi Driver Making Decisions at Airports

Abstract

Airport taxi scheduling is a classic and practical problem. Reasonable decision-making scheme is conducive to maximize the driver's profit, and also help the airport to allocate driving resources efficiently. In this paper, the decision-making mechanism of whether the driver leaving or staying at the airport is designed, and the scheme of airport's passenger-taxi boarding is planned.

For questions 1 and 2, considering the driver aims to make more profits, we take the taxi's net income in a certain time interval as the decision standard. we take the particular time period from the taxi entering the "storage pool" to the completion of the passenger delivery. In this way, various factors can be comprehensively analyzed. By cluster analysis of timestamp, taxi density, flight density, length of stay, we can predict the queueing time of taxi at a certain timestamp. Factors can be quantified by airport data, approximation, probability statistics and other methods. After substituting the model into the one-day data of Pudong airport, we come to the conclusion that the net income of taxi waiting in line during 22:00-24:00 and rush hour will be higher than that of returning to the city, and the decision-making model has the greatest dependence on flight density.

For the third problem, the setting of boarding points should consider the number and the interval of boarding points. Therefore, we set up a dynamic solution, which uses genetic algorithm. And in order to solve the problem of taxi and passenger arrangement, A queuing service model about how many taxis correspond to one boarding point and how many passengers correspond to one boarding point is setting up. After substitute the data into the model we get the setting schemes: before six o'clock in the evening, the pedestrian volume is small, three boarding points are set on one side and two parking points are set on the other side, the time interval of each parking point is 22m, three taxis are arranged for each batch to go to one boarding point. After 6 p.m., there is a large pedestrian volume. So there will be 28 boarding points on both sides of the road. Each boarding point is 122 meters apart, and for each batch two taxis are arranged to go to same boarding point.

For the fourth question, our priority scheme is to add a short distance Lane in the queuing area for the latest taxis after delivering short distance passengers. As to whether the scheme can make the taxis' revenue more balanced, we build a model to verify. In the verification, we find that when there is no priority, the short-distance vehicle revenue is indeed inferior to that of the long-distance vehicle. And the average revenue of the short-distance vehicle can be improved with our solution.

Contents

1 Restatement of the problems

1.1 Background

Taxi is an important way for passengers to go to the destination after getting off the plane. Passengers who want to take a taxi will go to the loading area for queuing, and then take a taxi. The airport's managers will allocate the number of passengers and taxis allowed. Domestic taxis often need to make decisions after delivering guests to the airport:

(A) go to the "car storage pool" and wait in line to take passengers and then return to the city. There is a time cost of queuing.

(B) return to the urban area without load, save the cost of queuing time, but pay the cost of no-load, and lose the potential passenger source of the airport.

For how to make decisions, drivers often judge empirically by the number of flights and passengers arriving in a certain period of time. In the real scene, there are many determinate and non determinate factors that affect the choice of drivers.

1.2 Tasks

In this paper, the mathematical model is established and the following problems are solved:

(1) find out the factors that affect the driver's judgment, including taxi revenue, number of airport passengers, etc. This paper analyzes the influence mechanism of various factors on taxi drivers, establishes the decision-making model of taxi drivers, and designs the selection strategy.

(2) collect the data of a domestic airport and related taxis, give the taxi driver's decision-making scheme in this situation, and analyze the rationality of the decision-making model built in the previous question, as well as the dependence on various factors.

(3) in the loading area, it is usually the taxi queuing to carry passengers or the passenger queuing for taxis. Now, there are two parallel lanes in the loading area. It is necessary to design the scheduling scheme of the taxi and the passenger and set the loading point so as to make the loading efficiency of the loading area the highest.

(4) there are short-distance passengers and long-distance passengers in the airport. Taxi drivers can't refuse to take them, but they can take them back and forth. In order to balance the income of all taxis, it is necessary to design a priority scheme for short-distance vehicles and verify the scheme works.

2 Model Analysis

2.1 Analysis of task 1

First, we have to determine the driver's decision criteria. Here it's the net income of the queuing car and the returning car in the time interval of the queuing taxi from start queuing to finish passengers delivering , so that important factors such as queuing time, airport passenger source, no-load cost can be taken into account. Here, the net income is calculated by subtracting the total fuel cost from the total fare income. For queuing vehicles, the net income is related to mileage, single ride fare and oil price. For return vehicles, the net income is also related to the saved queuing time, vehicle speed, probability of carrying urban passenger, weather, road conditions and other factors. Vehicle speed and mileage can be obtained by approximation; pricing rules and oil prices can be inquired; urban passenger carrying probability can be calculated by statistics; weather influence are related to visibility; road conditions can be estimated by the traffic flow obeying Poisson distribution; queuing time cannot be obtained directly, which is affected by timestamps, number of passengers and number of taxis, while drivers cannot determine the number of passengers directly, and what can be determined is the number of flights arrived in this period. Therefore, cluster analysis is carried out for (time point, flight density, taxi density, vehicle dwell time) to predict the taxi queuing time in the new situation.

2.2 Analysis of task 2

First of all, we need to determine an airport, take the relevant samples from the taxi and flight data, and set some parameters. Then bring the samples (time point, flight density, taxi density, vehicle dwell time) into the cluster model, adjust the cluster number k, label variable distance weight and other relevant parameters, find the best cluster number, and then bring the new samples without queuing time into the decision-making model to obtain the corresponding decision. This paper gives the selection scheme of taxi drivers in different situations.

As for the decision model's dependence on various factors. We combine the decision-making model with time period, flight density and taxi density. Observe and analyze their relations by drawing statistics.

Finally, tested by the actual situation, such as public transportation, urban roads and some official statistical data, the decision-making model is reasonable. Then, the sensitivity matrix is obtained by using sensitivity analysis on the decision-making model. And we can obtain the variables with the the model's highest dependence.

2.3 Analysis of task 3

For this question, we play the role of management department, need to set up boarding points, and arrange taxis and passengers to the corresponding boarding points, the goal is to make the total boarding efficiency the highest. There are only two lanes. In order to increase efficiency, boarding points are set on both sides of the road, and the number of passengers at each side of the boarding point should be as average as possible. It is also necessary to allocate the same number of taxis to the boarding point, as well as to find out how many taxis are dispatched to the same boarding point at a time.

In order to make the total boarding efficiency the highest, that is, the total waiting time of passengers is the least. The waiting time of passengers is divided into two parts, one is the time when the passengers arrive at the corresponding boarding point, the other is the time when the passengers wait in line at the boarding point until they become the first person of the line. When the passenger flow is large, and a certain flight arrives at the airport, the passengers of the previous flight are possibly still waiting for the taxis, so it is unnecessary to calculate when they arrive at the corresponding boarding point, because they are already in the waiting line. Therefore, the model needs to be classified into two types, which are respectively used in the scenarios with large and small passenger flow. Moreover, when the passenger flow is large, in order to meet the needs of passengers, the number of boarding points will inevitably increase, but not as much as possible, because there will be "ghost traffic jam" phenomenon affecting the average speed of taxis, so we use the logarithmic model of speed and traffic density to simulate this phenomenon when the traffic flow is large, and the setting of boarding points needs to consider the interval to ensure safety and convenience for passengers. Finally, we establish the relationship between the number of boarding points, the interval, the scheduling scheme and the total waiting time of passengers. We use genetic algorithm to find the number of boarding points, the interval and the scheduling scheme with the minimum waiting time of passengers, and use the data of Pudong Airport as an example to analyze.

2.4 Analysis of task 4

The purpose of establishing the short-distance vehicle priority scheme is to reduce the queuing time of short-distance vehicles. The effective method is to set up the short-distance vehicle priority channel, other vehicles can't enter the queue until the short-distance vehicle reaches the passengers.

The equilibrium of revenue is reflected in the variance of revenue of all taxis. Therefore, the average revenue model of taxis is established. It is assumed that the short-distance vehicle will return to the airport every time until it reaches the long-distance passenger, and the long-distance vehicle will not return to the airport after completing the passenger journey. The appropriate time period for the short-distance vehicle is from start queuing to finish delivering the long-distance passenger, and the appropriate time interval for the long-distance vehicle is from start queuing to completion of passenger delivering. According to this, we can calculate the hourly income of short-distance vehicles and long-distance vehicles in their respective time intervals. Finally, we can get the variance of hourly income of all taxis by considering the proportion of long-distance vehicles and short-distance vehicles. If the variance is reduced than before, it is proved that the scheme is feasible and effective.

3 Model Assumptions

- Assumption 1: The taxi returning to the city has the same time interval, revenue and mileage.
- Assumption 2: The passengers in task 1 and task 2 are all long-distance passengers returning to the city.
- Assumption 3: The taxi will return to the airport after the short-distance passengers are loaded, and it would not leave until the long-distance passengers are loaded.
- Assumption 4: The taxi will not return after the long-distance passengers are loaded.
- Assumption 5: The mileage and income of short distance passengers received by short distance vehicles are the same every time.
- Assumption 6: The time for the short distance bus to return to the airport is fixed.
- Assumption 7: The revenue and mileage of the last long-distance passenger carried by the short-distance taxi is the same as that of the long-distance passenger carried by the long-distance taxi.
- Assumption 8: The time from the car pool to the nearest boarding point is not counted.
- Assumption 9: The distance from the nearest boarding point to the entrance of the loading area is not counted and distance from the nearest boarding point to the overpass exit is not counted.

- Assumption 10: The intervals between arrival time of flight and the time when passengers arrive at the loading area are not counted
- Assumption 11: In the same period of time, passengers' need for a ride is the same.
- Assumption 12: In the same time period, the arrival time interval between flights is the same
- Assumption 13: There has always been an unlimited number of taxis available for dispatch.
- Assumption 14: The weather condition of task 2 is good and the road traffic is smooth
- Assumption 15: The arrival time of the flight in task 2 is expected as landing time of the flight.

4 Parameter Table

Important symbols that in this article are listed in table 1. Some of the symbols in the following contents are not listed here.

Symbol	**Defination**
W_1	Net income of queue drivers in a given time window
I_1	Total revenue of drivers in queue in a given time window
O_1	Fuel consumption of queuing drivers carrying passengers
W_2	The net income of no-load drivers in the specified time window
I_2	The total income of no-load drivers in the specified time window
O_e	Fuel consumption of no-load driver
L_1	Distance between airport and Downtown
f_1	The starting price of three-stage gradient charging for taxis in the city
f_2	The second gradient average fee of the three-stage gradient charge of the city's taxi
f_3	Third gradient average fee of three-stage gradient charge of taxi in the city
a	Starting mileage of three-stage gradient charging for taxis in the city
b	Three gradients of taxis in the city
o	Fuel consumption per unit mileage
k	Unit volume oil price
α	Weather influence factors of driving speed
β	Traffic impact factors
M_o	Critical number of vehicles in a road section
V_1	Average speed of taxi queue
t_s	Waiting time for taxi
N_a	Flight density at a certain time
N_c	Taxi density at a certain time
T	Time tab
t_q	Average queuing time of each vehicle in a certain period of time
N_y	Number of passengers carried by return taxis in the saved queuing time
i_2	Return taxi charges for each time of carrying passengers in urban area
L_2	The mileage of each seeing off in return taxi Downtown
O_2	Total fuel cost for returning taxi
O_{e1}	Fuel consumption of return taxi from airport to urban area

Table 1: The definition of the symbols.

5 Problem 1: Design for Decision Model

Taxi drivers always aim to make more profits, therefore, we use the net income of taxi drivers in a certain time interval as the standard of decision-making. Assuming that both taxis D_a and D_b arrive at the airport at time A, driver D_a decides to enter the "car storage pool" to queue for passengers and complete the delivery at time B, then time A to time B is the time period considered by our decision-making model, because this time period takes into account the main factors such as queuing time, no-load time and so on. In this period, the net income W_1 of D_a is the total income I_1 minus the fuel consumption O_1 when delivering passengers, i.e:

$$W_1 = I_1 - O_1 \tag{1}$$

Set $t(Y - X)$ as the time spent from X to Y, calculated in hours. If D_b decides to return to the urban area to carry passengers, then during the period of $T(B - A)$, D_b's net income W_2 is the total income I_2 minus the total fuel cost O_2 consumed in delivering passengers, and then minus the fuel cost O_e consumed in no-load time:

$$W_2 = I_2 - O_2 - O_e \tag{2}$$

If $W_1 > W_2$, it is a better choice for drivers who arrive at the airport at time A to enter the "car storage pool" to carry passengers. On the contrary, if $W_1 < W_2$, it is more cost-effective to return to the city without load. If the revenue is the same, both options are available.

5.1 Net Income for Queueing Taxis

Further analysis, for D_a: specifically speaking, I_1 is the fare paid by passengers. I_1 is related to the mileage L_1 for delivering passengers. In the real scene, most passengers from the airport will go to the city, which is also mentioned in the problem statement. Therefore, we set the distance from the airport to the city center as L_1. I_1 is also related to the charging standard of taxis. Generally speaking, L_1 exceeds the gradient charging mileage a and b of taxis, so there are:

$$I_1 = f_1 + f_2 \cdot (b - a) + f_3 \cdot (L_1 - b) \tag{3}$$

where $f_1, f_2$and f_3 are the gradient pricing fees for three sections of taxi mileage respectively. O_1 is L_1 multiplied by fuel consumption per kilometer o multiplied by oil price per volume k. Namely:

$$O_1 = L_1 \cdot o \cdot k \tag{4}$$

Therefore, for driver D_a, the net income can be specified as:

$$W_1 = f_1 + f_2 \cdot (b - a) + f_3 \cdot (L_1 - b) - L_1 \cdot o \cdot k \tag{5}$$

The time period $t(B - A)$ can be divided into "storage pool" queuing time t_q and delivery time t_s. t_s can be estimated by dividing average speed per mileage. The average speed v_1 here is approximated by the airport high speed limit minus $20km/h$. In real life, the speed will also be affected by the weather conditions, and the bad weather will reduce the safe driving speed. The impact factor is set as $\alpha(0 < \alpha \leq 1)$, and α is 1 when the weather is good. In other weather conditions, α is related to visibility. The lower the visibility is, the smaller α is and the slower the speed is. In addition to the weather, the traffic conditions in this period will also affect the driving speed, so that the influencing factor is $\beta, (0 < \beta \leq 1)$. Set the number of vehicles within the mileage L_1 section at time t as M, when $M \leq M_0$, the driving is smooth, β is 1. M_0 can be approximated by the average daily traffic flow of this section, M obeys the Poisson distribution $P(\lambda)$, so that:

$$\beta = \begin{cases} 1, if M \leq M_0 \\ P(M > M_0), if M > M_0 \end{cases} \tag{6}$$

And we can conclude that:

$$t_s = \frac{L_1}{\alpha \cdot \beta \cdot V_1} \tag{7}$$

But calculation of t_q is more complex, which is related to the number of passengers in this period and the vehicle density N_c in this period. Here, the number of passengers can be estimated by the flight density N_a. In order to better predict the t_q of a certain period, we divide 0:00-24:00 into 12 periods, each of which has a corresponding period label T. Using K prototype clustering[1], a set of sample points (T, N_a, N_c, t_q) is established, where T is the time period label. Therefore, we can find the corresponding classification for the specific (T, N_a, N_c), and then get t_q. The clustering steps are as follows:

$X = \{X_1, X_2, \ldots, X_n\}$ represents n such sample point sets. For each point set, there are four attributes (T, N_a, N_c, t_q) corresponding to $X_i = \{x_{iT}, x_{ia}, x_{ic}, x_{iq}\}$. For numerical attributes N_a, N_c, t_q, firstly normalization processing is carried out:

$$x_{ik}{}' = \frac{x_{ik} - \bar{x_k}}{s_k} \tag{8}$$

where:

$$\bar{x_k} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_{ik} \tag{9}$$

$$s_k = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_{ik} - \bar{x_k})^2} \tag{10}$$

we use Euclidean distance to define the different distance of numerical attribute:

$$D(X_i, X_j) = \sqrt{(x_{ia} - x_{ja})^2 + (x_{ic} - x_{jc})^2 + (x_{iq} - x_{jq})^2} \tag{11}$$

For classification attribute T, Hamming distance is used:

$$d(x_{ik}, x_{jk}) = \begin{cases} 1, if x_{ik} \neq x_{jk} \\ 0, if x_{ik} = x_{jk} \end{cases} \tag{12}$$

So the distance between the sample point set X_i and the cluster Q_l is:

$$D_s(X_i, Q_l) = D(X_i, X_l) + \mu \cdot d(x_{iT}, x_{lT}) \tag{13}$$

where X_l is the prototype (center) of the cluster Q_l. Then the total loss function of this classification is:

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{il} D_s(X_i, X_l) \tag{14}$$

5.2 Net Income for Returning Taxis

For driver D_b, he takes time t_s to return to the urban area. Within the time of driver D_a's queuing time t_q, driver D_b can receive passengers many times, set it to be N_y times. The total income I_2 of driver D_b is the fare paid by a passenger per time (name it i_2) times N_y (assume the income of each passenger delivering in the urban area is the same). i_2 is related to the distance L_2. In reality, most of the passengers in the urban area are of short distance trip. Here, we use the data we collected to count the average mileage of each passenger delivering downtown as L_2, and then substitute it into the taxi's gradient charging standard to get I_2:

$$i_2 = \begin{cases} f_1, if L_2 \leq a \\ f_1 + (L_2 - a) \cdot f_2, if a < L_2 \leq b \\ f_1 + f_2 \cdot (b - a) + f_3 \cdot (L_2 - b), if L_2 > b \end{cases} \tag{15}$$

The calculation of O_2 is similar to that of O_1:

$$O_2 = L_2 \cdot o \cdot k \cdot N_y \tag{16}$$

The oil cost of O_e is divided into two parts, one is the no-load oil cost O_{e1} from the airport to the urban area, the other is the oil cost O_{e2} waiting for passengers in the urban area for N_y times. And the distance from the airport to the city is L_1, so we have:

$$O_{e1} = L_1 \cdot o \cdot k \tag{17}$$

On the other hand, we can use the collected data to count the average time taken by taxis to pick up a passenger in the urban area, at different time labels, and then take the urban speed limit minus $10km/h$ as the average speed of the taxis in the urban area, viz:

$$O_{e2} = t_w \cdot V_2 \cdot o \cdot k \cdot N_y \tag{18}$$

If the time of delivering passengers in urban area is t_p, then:

$$t_p = \frac{L_2}{\alpha \cdot \beta \cdot V_2} \tag{19}$$

In addition to the impact of weather on taxi speed, it will also affect t_w. For example, heavy rain or hot weather will make urban residents more inclined to take a taxi, and t_w will decrease, while in typhoon weather, people are not inclined to take a taxi, and t_w will increase. Set the influencing factor as $\gamma, (\gamma > 0)$. From this, we can get the approximate value of N_y:

$$N_y = \frac{t_q}{\gamma \cdot t_w + t_p} \tag{20}$$

Therefore, for driver D_b, the net income can be specified as:

$$W_2 = i_2 \cdot N_y - L_2 \cdot o \cdot k \cdot N_y - (L_1 \cdot o \cdot k + t_w \cdot V_2 \cdot o \cdot k \cdot N_y) \tag{21}$$

6 Problem 2: Decisions for Drivers and Analysis for the Decision Model

6.1 Data Preparing

1. **Raw Data**
   The overview of taxi data and flight data is shown in the form of data distribution statistics chart as followed.And the detailed datas are placed in appendix.

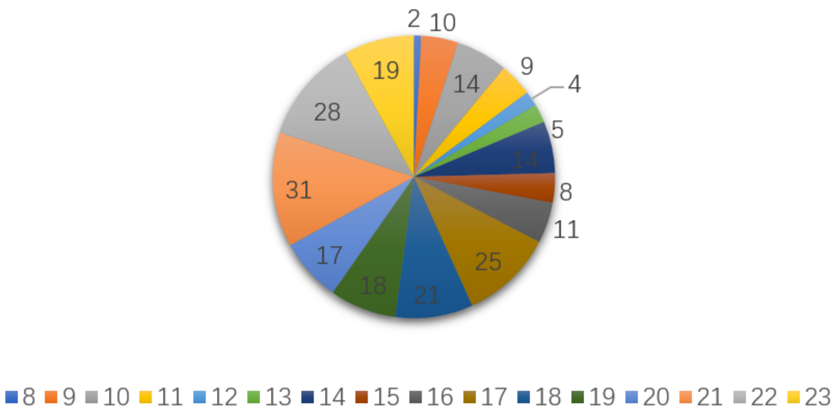
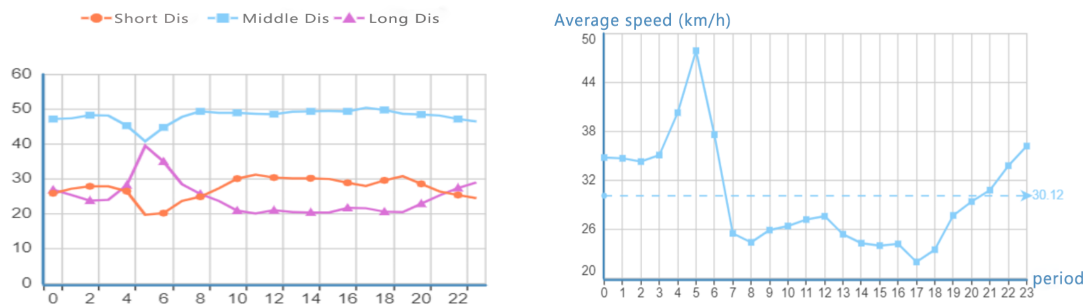


Figure 1: Flight data of Pudong Airport 20180602 obtained by crawling



(a) Passenger distance distribution of taxi in Shanghai in 2015

(b) Average speed distribution of taxis in Shanghai in 2015

Figure 2: Taxi Data

2. **Processed Data**

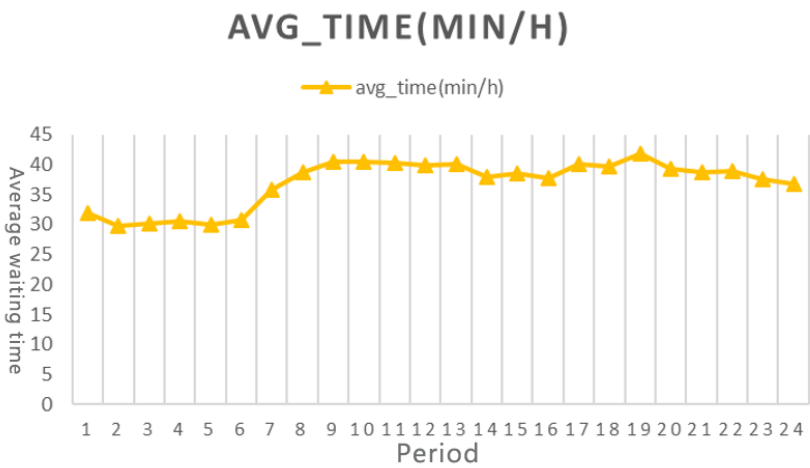


Figure 3: Average waiting time of receiving orders in urban areas in different time periods

From the raw data, we do a lot of processing, and finally extract 2077 pieces of data as clustering candidate data.And the field is shown in below table.And we put forward some assumptions: If the taxi enters Pudong Airport in empty state at t_{empty} and becomes loaded at t_{busy}, the starting time of queuing is assumed to be t_{empty}. If the taxi enters the airport in loaded state and becomes empty at t_{empty} and becomes loaded at t_{busy} again, the starting time of queuing is assumed to be t_{empty}.

Table 2: Candidate data for clustering

Field	**Description**	**Calculate Method**
Id	taxi number	from origin data
$Time$	the time begin queuing at Pudong Airport	from origin data
$Wait\_time$	taxi queuing time in airport	$t_{busy} - t_{empty}$
$taxi\_density$	Accumulated number of taxis in Pudong airport within 2 hours ahead of $Time$	Use a set to record all the unique taxi number in these two hours, and the size of the set is the taxi density.
$flight\_density$	Total number of passengers landing at Pudong airport within 2 hours ahead of $Time$	Sum of the total number of passengers of the flight with the planned landing time in these two hours

6.2 Parameter Calibration

Table 3: Parameter Setting for Model

Symbol	Value	Meaning
L_1	46.4	Unit: km, the approximate distance from Pudong Airport To Downtown
t_DE	0.576014	Unit: H, the average waiting time of passengers in the queue in this period, selected from the clustering result set, time point is 19:17
α	1	Weather factor, for the convenience of calculation, it is assumed that the weather is in good condition
V_2	$f(t)$	Unit: km/h, average speed in urban area,$f(t)$ is the function to calculate V_2
V_1	64.7	Unit: km/h, average speed from Pudong to downtown
β	1	Road condition factor, also assuming good road condition
o	0.088	Unit: L/km, fuel consumption per mileage
k	7.13	Unit: ¥/L, unit oil price
f_1	14	Unit: ¥, starting price of taxi in Shanghai
a	3	Unit: km, first gradient mileage limit
f_2	2.5	Unit: ¥, more than 2.5 ¥/km over a
b	15	Unit: km, first gradient mileage limit
f_3	3.6	Unit: ¥, over part B, 3.6 ¥/km
μ	0.15	Weighting factors for time periods

6.3 Model Calculation

After standardizing the candidate data, 1000 pieces of 2077 pieces of data are randomly selected as samples. Each sample is $(time, flight\_density, taxi\_density, wait\_time)$, which corresponds to the parameter (T, N_a, N_c, t_q) in Task 1, we set $k = 4, 5, 6, 7$ for K prototype clustering, and bring k into the following formula:

$$E = \sum_{i=1}^{k} \sum_{i=1}^{n} D_s(X_i, X_{center})$$

Calculating the mean square error of each cluster set under each k value:

Table 4: Mean Square Deviation

K	E
4	100.56
5	35.94
6	165.67
7	189.86

It can be concluded from above table that the clustering effect is best when $k = 5$, and the clustering diagram after clustering is as follows:

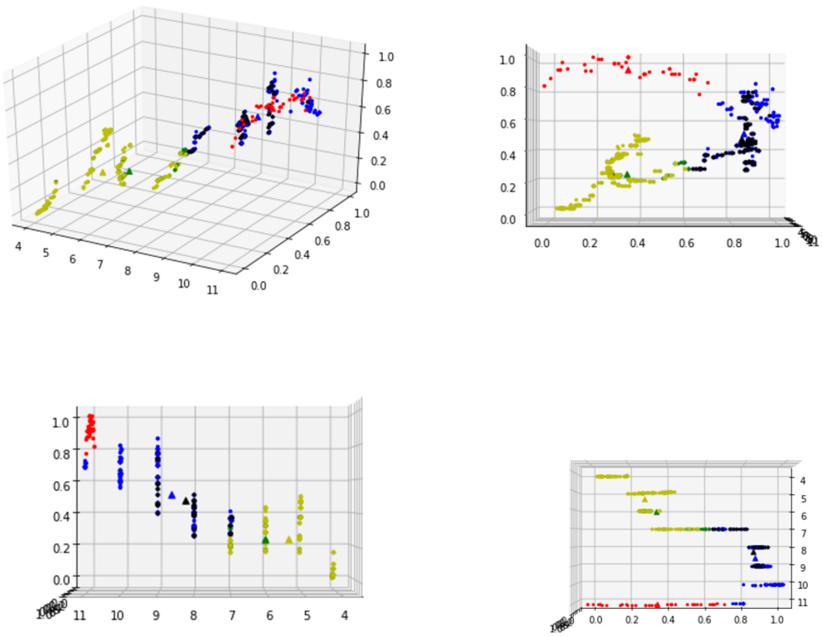
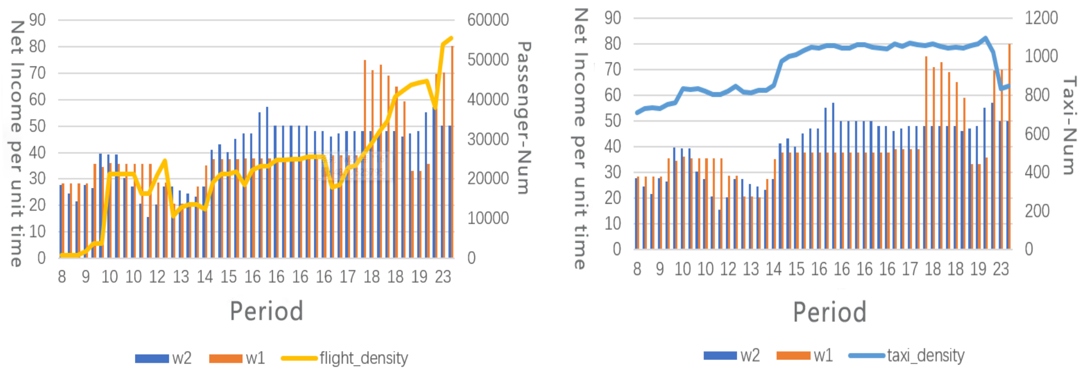


Figure 4: Distribution of Clustering Results

When $k = 5$, cluster sets and cluster centers are saved for the following calculation. Then, 50 pieces of data are randomly selected from 2077 pieces of data as decision samples (excluding the $wait\_time$ field, and the data is in the appendix), and brought into model. The net profit of each sample in the queue and the net profit returned to the urban area are calculated. The overall comparison chart is as follows:



(a) Net income per unit time - Number of passengers (b) Net income per unit time -Number of taxis

Figure 5: Comparison of overall results

According to the above figures, we can see the decision scheme made by the model:

Table 5: Decision Plan

Decision	Period
Queue up	$8:00-9:00$ & $11:00-12:30$ & $18:00-19:00$ & $22:00-24:00$
Return to urban	$9:00-11:00$ & $12:30-18:00$ & $19:00-22:00$

This plan is the choice we give to taxi drivers in this airport.

6.4 Rationality Analysis

The decision calculated by our model has the following characteristics:

1. Tend to choose a later night (such as 22:00-24:00), which is consistent with the actual situation of life. Because the choice of public transport at night becomes scarce, the probability of flight passengers choosing to take taxis will increase. Just as the proportion of night flight passengers choosing taxis in Pudong Airport calculated by China Daily is as high as 45%, in this case, the demand for taxis will be large, and the waiting time of drivers will be shortened. In addition, the charge of night taxi is higher, especially in the airport where the order receiving rate is probably long-distance, the driver's income will increase significantly. Therefore, drivers will naturally prefer to wait in line for passengers in this period of time.

2. Tend to make the choice of waiting in line at the peak of urban traffic (18:00-19:00 & 8:00-9:00). In fact, it is also easy to understand that when the urban area is in the rush hour of going to and from work, the road is blocked and the average traffic jam time is long, which will lead to the decrease of the driver pick-up frequency, the increase of the air run time and the increase of the fuel consumption, so the taxi drivers are more inclined to avoid such peak periods.

3. Tend to make the choice of waiting in line when the passenger demand is small (11:00-12:30). In the period of noon, according to people's normal life, they are at the time of lunch. At this time, the demand for taxis in the urban area is bound to be small.

4. In other time periods (mostly during the day), the proportion of airport passengers who choose to take taxis is small (only 15% of Pudong Airport passengers choose to take taxis

during the day according to China Daily net). If they queue up, they will waste a lot of time. On the contrary, in the urban area, the road traffic condition is normal, which is more conducive to business.

And about rationality analysis

1. The parameters such as taxi time interval, average speed distribution, average waiting time for single receipt and taxi pricing are all open real data. At the same time, the extracted airport taxi sample data is obtained through a large number of screening statistics, which is consistent with the actual situation.

2. The clustering model used to calculate the waiting time of the queue has a better clustering effect. After the evaluation of the silhouette coefficient combining the two factors of cohesion and separation, that is, for each vector in K clusters, the silhouette coefficient[2] of each vector is calculated by the following formula:

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{22}$$

6.5 Dependency Analysis

Our decision model consists of six variables: N_a,N_c,L_2,V_2,t_w,T,and we calculate the average and the maximum, minimum values of them respectively so as to conduct sensitivity analysis. And the values are shown in below table.

Table 6: Parameter sensitivity analysis

Parameter	Max	Min	Avg	Description
N_a	55923	740	23187	flight density
N_c	1107	703	943	taxi density
L_2	15.5647	12.0199	12.9387	taxi carrying distance per order
V_2	47.9	22.0	30.1375	taxi average speed in various time
t_w	0.982	0.4974	0.6155	average waiting time for receiving orders
T	23	8	11	time(in 24 hours)

Set the i parameter C_i of every parameter in $(N_a.N_c, L_2, V_2, t_w)$from its minimum value to the max value with stride s:

$$s = \frac{C_{i\_max} - C_{i\_min}}{10} \tag{23}$$

That is $\{C_{i\_min}, C\_i_min + s, C_{i\_min} + 2s, \cdots, C_{i\_max}\}$,and the remaining five parameters are set to average value.Then bring these variables into the decision model, so that we can get $\{d_{w\_i\_1}, d_{w\_i\_2}, \cdots, d_{w\_i\_10}\}$ where $d_{w\_i\_k} = w_{1\_i\_k} - w_{2\_i\_k}$.,So that the ratio of the increment of each d_w relative to the former to the increment of C_i can be obtained, that is:

$$r_{w\_i\_k+1} = \frac{|\frac{d_{w\_i\_k+1} - d_{w\_i\_k}}{d_{w\_i\_k}}|}{\frac{s}{C_{i\_min}+ks}} \tag{24}$$

we can get $r_{w\_i} = [0, r_{w\_i\_2}, r_{w\_i_3}, r_{w\_i\_4}, r_{w\_i\_5}]$.By calculating each of the six parameters, a sensitivity matrix of the six parameters can be obtained:

$$r_w = \begin{bmatrix} r_{w\_1\_1} & \cdots & r_{w\_1\_10} \\ \vdots & \ddots & \vdots \\ r_{w\_6\_1} & \cdots & r_{w\_6\_10} \end{bmatrix} \tag{25}$$

Finally, the average value of each line is obtained:

$$r_{w\_mean} = \begin{bmatrix} r_{w\_1\_mean} \\ \vdots \\ r_{w\_6\_mean} \end{bmatrix} \tag{26}$$

Bring the value of the previous table into the calculation to get r_{w_mean}

Table 7: Sensitivity mean value

Variable	N_a	N_c	L_2	V_2	t_w	T
value	25.26	14.37	1.59	6.78	-9.43	7.12

It can be seen from the above table that: the sensitivity of N_a is the highest, and the sensitivity of $N_c$is the second, that is to say, the decision model has the greatest dependence on flight density N_a, and the second is the dependence on taxi density N_c.

7 Problem 3: Design for Schedule Model

In this question, we play the role of management department. There are two parallel lanes in the “car zone” of the airport. We are required to set up a reasonable “boarding point”. Also, we need to arrange taxis and passengers and ensure the safety of vehicles and passengers. Under the conditions, the total ride efficiency is the highest. When there is no dispatch, it is easy for passengers to give priority to the vehicle on the side closer to themselves. There is no load on the side, or when driving to the opposite side of the road, the taxi in the passing lane causes a safety hazard. If you do not stipulate that you can only get on the train again, it will also cause the passengers behind you to choose the taxi behind, which is inefficient and dangerous. In addition, if the car is released too much, it will also lead to confusion. To summarize the above issues, we need to make provisions to facilitate scheduling.

7.1 Assumptions

When a new flight arrives at the airport, passengers who need a taxi are required to register with the management department, and the management department also assigns the pick-up point that each passenger should go to. The taxi waits in the storage pool. According to the physical location of the relative storage pool at each loading point and the number of existing passengers, the management department allocates the loading point to the taxi in the storage pool, and the storage pool reaches the nearest one. The time of the vehicle is not counted. Since the many-to-many model has better efficiency, we also plan to use the multiple-to-multiple model.Therefore, for each boarding point, you can specify k consecutive vehicles in the storage pool at a time. The car goes to the same pick-up point, and the first k passengers in the team can carry the baggage at the same time to get on the bus. Considering the efficiency and passenger safety, the upper vehicle point is set on the outside of the two parallel lanes, and only the overpass can be reached to the opposite vehicle, which takes t_t seconds. The initial entrance position of the “carriage zone” is on one side of the road, that is, the initial position of all passengers is on the same side, named L side, the other side of the road is named R side, and the number of passengers on each boarding point on the same side As balanced as possible.

7.2 Model establishment

In terms of the setting of the boarding point, we have n loading points on the L side, and are lined up at the entrance of the "riding area". The distance from the first boarding point to the boarding area is not counted. m pick-up points, and at the exit of the flyover, line up, the distance from the first pick-up point to the exit of the bridge is not counted. Set the distance between the vehicles to be $(d + k * \Delta d)$m. Considering the passenger's walking speed, it is necessary to $(d + k * \Delta d)$s to reach the next boarding point. After the passenger waits for the taxi, it takes c seconds to carry the luggage, and the taxi can enter the driving state. Due to the topic, the total ride efficiency is the highest. Among them, the passenger's total riding efficiency E:

$$E = \frac{P}{T_s} \tag{27}$$

P is the total number of passengers in a round of flights, and T_S is the sum of the time when P passengers are waiting to board. It can be seen that if E is as large as possible, the TS should be as small as possible, that is, all passengers should wait the least. The time TS in which the passenger waits is composed of two parts: the total time T_a taken by all the passengers to the boarding point, and the total time T_w of all the passengers waiting at the boarding point to the current taxi. which is

$$T_s = T_a + T_w \tag{28}$$

considering

$$T_a = T_{aL} + T_{aR} \tag{29}$$

It is divided into two parts: the total time T_{aL} used by the L side passengers to get to the boarding point and the total time T_{aR} used by the R side passengers to get to the boarding point. Let the number of passengers on the L side be x, and the number of passengers on the R side be $P - x$.

Considering T_{aL}, the time required for the passenger to go to the nearest first pick-up point is 0, and the second required time to go to the nearest $(d + k * \Delta d) * s$,..., to the nearest i The boarding time is $(d+k*\Delta d)*(i-1)$. i is a natural number. The above assumes that the number of passengers at each boarding point is average, so the number of passengers at each boarding point on the L side is $\frac{x}{n}$. Use the Gaussian summation formula, where n is the base

$$T_{aL} = \frac{(d + k * \Delta d) * (n - 1) * x}{2} \tag{30}$$

Consider T_{aR} again, compared to T_{aL}, plus the passengers crossing the bridge T_s

$$T_{aR} = \frac{(d + k * \Delta d) * (m - 1) * (p - x)}{2} + t_t * (P - x) \tag{31}$$

Consider T_w, that is, the sum of the time that all passengers can wait until the current taxi can drive at the boarding point, which is the time $t_w f$ of each passenger waiting at the boarding point to wait for the current taxi to drive. $T_w f$ also has two parts: each passenger needs to wait for the taxi to come time t_1 and each passenger needs to carry luggage, sit in the seat, wait for the taxi to enter the driving state time t_2 Consider t_1 first. According to the previous assumption, the taxi waits in the storage pool. According to the physical location of the relative storage pool and the number of existing passengers at each loading point, the management department allocates the waiting point to the waiting taxi. The time from the car pool to the nearest boarding point is not counted. Each time a batch of cars is issued, the quantity is $k(m + n)$, k is assigned to the same boarding point, k cars, m+n is all the boarding points. On average, each vehicle needs to drive all the entry points at a speed of vm/s, and because the number of points on the L side and the R side is different, they are:

$$t_{1L} = \frac{(d + k * \Delta d) * (n - 1)}{v * k} \tag{32}$$

$$t_{1R} = \frac{(d + k * \Delta d) * (n - 1)}{v * k} \tag{33}$$

After the Gaussian summation of the number of people

$$T_{1L} = \frac{(d + k * \Delta d) * (n - 1) * x}{v * k} \tag{34}$$

$$T_{1R} = \frac{(d + k * \Delta d) * (n - 1) * (p - x)}{v * k} \tag{35}$$

Considering t_2, each passenger needs to wait for all the passengers in front of him to carry the luggage into the car. Each person needs about c seconds, and the passenger is ranked in the team's i-th.

$$t_2 = \frac{c * (i - 1)}{k} \tag{36}$$

After the Gaussian summation of the number of people

$$T_{2L} = \frac{c}{2} * \frac{x}{n} * \frac{x}{n * k} \tag{37}$$

$$T_{2R} = \frac{c}{2} * \frac{P - x}{m} * \frac{P - x}{m * k} \tag{38}$$

According to the Definition of T_w

$$T_w = T_{1L} + T_{1R} + T_{2L} + T_{2R} \tag{39}$$

According to the Definition of T_s

$$\begin{aligned} T_s = {} & \frac{(d + k * \Delta d) * (n - 1) * x}{2} + \frac{(d + k * \Delta d) * (m - 1) * (p - x)}{2} + t_t * (P - x) \\ & + \frac{(d + k * \Delta d) * (n - 1) * x}{v * k} + \frac{(d + k * \Delta d) * (n - 1) * (p - x)}{v * k} + \frac{c}{2} * \frac{x}{n} * \frac{x}{n * k} \\ & + \frac{c}{2} * \frac{P - x}{m} * + \frac{P - x}{m * k} \end{aligned} \tag{40}$$

Finding n, m, and x that minimizes the value of (3-14), you can get how many parking points are needed on the L side, how many parking points are needed on the R side, and how many people need to be dispatched to cross the road. Pay special attention to this style, and do not consider the people waiting for the flight at the boarding point, so the usage is: when the new flight arrives, the boarding point has no passengers who are still waiting for the previous flight. Scope of application: Airport when passenger traffic is small.

But more generally, when the passengers of the flight reach the parking spot, they find that the passengers who arrived before the flight are waiting for the bus. The number of teams at each boarding point is B, so the length of all the boarding points in the dispatch should be The same is true for all passengers to wait the least. And the time to walk to the pick-up point should not be counted, because the length of the team is also reduced during the past time, and it has been counted in the waiting time. The number of R-side and R-side ride points should both be n. At this time, the number of passengers at the boarding point is (B*2*n', +p), of which

2*n′ is the number of teams before the flight arrives, and after the flight arrives, the number of each team is (B*n′ ,+p)/(2*n) Use the Gaussian summation formula

$$T_s = \frac{(d + k * \Delta d) * (n - 1) * (B * 2 * n' + p)}{v * k} + c * \frac{2 * B * n' + p}{*} 4 * n * (2 * B * n' + p) \quad (41)$$

In order to obtain B, we assume that the flight is uniform over a period of time, and can calculate the time difference of the arrival of the flight. This data is equivalent to the time difference for every two passengers arriving at the passenger point. At the same time, the number of passengers that can be sent out per minute on the pick-up point is equal to the number of passengers that are driven out per minute. Considering that a batch of vehicles is 2*n, the passengers must wait for the passengers for c seconds, plus vm/s to drive through all the parking spots $\frac{(n-1)*(d+k*\Delta d)*s}{v}$, so

$$E_m = (\Delta)t * \frac{2 * n}{c + \frac{(n-1)*(d+k*(\Delta)d)}{v}} * k \quad (42)$$

Within the time difference between the arrival of two passengers and the passenger point, the total number of people who board the bus is Δ t*E_m Considering that B is the arrival of this flight, the person on the previous flight that is waiting for each boarding point of the bus, when the next flight f′ arrives, the number of passengers on the previous flight is still waiting for each passenger. For B′, the new flight brought P/2n individuals to each boarding point, but each boarding point in the Δ t where the new flight did not come $(\Delta t * E_m)/2n$ person. and so

$$B = B' + \frac{p}{2 * n} - \frac{\Delta t * E_m}{2 * n} \quad (43)$$

It should also be noted that when n is small, v can be approximated as the safe speed limit of the vehicle, but as n increases, the density of the vehicles on the road will be large, and each taxi must brake at least once on the road. Because they have to stop and pick up passengers, they will have a "ghost traffic jam" phenomenon - because the brakes of the first car, all the drivers behind must also brake, all the drivers behind must also brake, one car passes down, with The “fluctuation effect” will lead to a general slowdown in large-scale road traffic.
We model this urban traffic flow problem[3]

Model assumptions: Assume that the vehicle speed v is a function of the traffic density k′, and when k′,=k_j/e is set, k_j is the jam density (density is as high as the vehicle speed is 0). In the steady state, the vehicle speed v and the head spacing d of the adjacent two vehicles are the same, and thus the traffic density k is equal to 1/d, which is a constant. When the $n - 1th$ car decelerates or accelerates and the steady state is destroyed, the braking force or driving force applied by the nth car is proportional to the speed difference between the two cars, inversely proportional to the interval between the two cars, and stabilizes after braking or driving. status.

According to Newton’s second law and the third hypothesis, differential equations can be written.

$$\frac{\partial V_n}{\partial t} = \lambda * \frac{v_n - v_{n-1}}{x_n - x_{n-1}} \quad (44)$$

Where λ is the proportionality factor. Notice the derivative relationship between v_n(t) and x_n(t)

$$\frac{\partial V_n}{\partial t} = \lambda * \frac{\partial In[x_n - x_{n-1}]}{\partial t} \quad (45)$$

integrate on both side

$$v_n(t) = \lambda * In[x_n(t) - x_{n-1}(t)] + c \quad (46)$$

Where c is the undetermined constant. According to Hypothesis 2, after the steady state is restored,

$$v_n(t) = v \tag{47}$$

$$v_n(t) - x_{n-1}(t) = d = \frac{1}{k'} \tag{48}$$

$$v = \lambda * In[k'] + c \tag{49}$$

According to conditions of Hypothesis 1

$$\lambda = v_1 \tag{50}$$

$$c = v_1 * In[k_j] \tag{51}$$

The logarithmic model of vehicle speed and traffic density under high traffic density is obtained.

$$v = v_1 * In[\frac{k_j}{k}] \tag{52}$$

Where v_1 is the smoothing speed, k′ is the vehicle density, and k_j is the blocking density.

7.3 Model verification:

7.3.1 preparation of data

The following is an example of setting the boarding point with Pudong Airport as an example. The data we collected is as follows:

id	flight_num	time	total_passenger_num
0	0	0- -2	0
1	0	2- -4	0
2	0	4- -6	0
3	0	6- -8	0
4	12	8- -10	9270
5	23	10- -12	24690
6	9	12- -14	9810
7	22	14- -16	21940
8	36	16- -18	28853
9	39	18- -20	39016
10	48	20- -22	35896
11	47	22- -24	36013

Figure 6: the data that we collected

(The third column is the time period, the second column is the number of flights in the time period, and the fourth column is the total number of passengers in the time period)

For the number of people on each flight, P will choose a taxi. We check the airport traffic every 2 hours, and then based on the big data of Pudong Airport - during the day, about 15Assuming that the number of taxi passengers selected on each flight is the same, you can get the number of taxis P that will be required for each flight, and the direct time interval Δt.

id	flight_num	time	total_passenger_num	taxi_num	taxi_num/flight_num	interval
0	0	0- -2	0	0	0	0
1	0	2- -4	0	0	0	0
2	0	4- -6	0	0	0	0
3	0	6- -8	0	0	0	0
4	12	8- -10	9270	927	77	600
5	23	10- -12	24690	2469	107	313
6	9	12- -14	9810	981	109	800
7	22	14- -16	21940	2194	99	327
8	36	16- -18	28853	2885	80	200
9	39	18- -20	39016	11704	300	184
10	48	20- -22	35896	10768	224	150
11	47	22- -24	36013	10803	229	153

Figure 7: Flight schedule

7.3.2 Model calculation

The initial length of the team is 0, and the airport traffic is small in the morning. considering:

$$\begin{aligned} T_s = & \frac{(d+k*\Delta d)*(n-1)*x}{2} + \frac{(d+k*\Delta d)*(m-1)*(p-x)}{2} + t_t*(P-x) \\ & + \frac{(d+k*\Delta d)*(n-1)*x}{v*k} + \frac{(d+k*\Delta d)*(n-1)*(p-x)}{v*k} + \frac{c}{2}*\frac{x}{n}*\frac{x}{n*k} \\ & + \frac{c}{2}*\frac{P-x}{m}*+\frac{P-x}{m*k} \end{aligned} \tag{53}$$

Where d is the minimum distance of the boarding point, considering that the boarding point is relatively crowded, the taxi speed limit is 20km/h, the safety distance is 10m, take d=10, Δ d=4. There are k vehicles in the same train at the time of each batch of vehicles. Considering safety and order, they should be as small as possible. T_t is the time of crossing the bridge, which is roughly set to 30s. The average speed of the taxi is considered to be 5m/s and the baggage time is 20s.

In order to find the loading point n, m and the passenger arrangement x and the taxi arrangement k which minimize the waiting time of T_s, the genetic algorithm[4] is used to solve the problem. When the termination condition is reached for a certain flight, the passengers of the previous flight are still found. queue.

Substituting into the model calculation, the results are as follows. (The first column: the first flight on that day, the second column: the average number of taxis required for passengers on each flight, the third column: flight interval (seconds), the fourth column: the number of points on the L side (n), the fifth column: the number of boarding points on the R side (m), the sixth column: the number of passengers who need to cross the bridge account for all passengers (x/p), the seventh column: the same boarding point in a group of taxis Number of cars (k), eighth column: number of queues at the pick-up point when the flight arrives (B), ninth column: model maximum number of passengers per second (E_m)

1	77	600	3	2	0.786953	3	0	0.435926
2	77	600	3	1	0.808251	3	0	0.438343
3	77	600	3	2	0.798914	3	0	0.438349
4	77	600	3	2	0.781756	3	0	0.431217
5	77	600	3	2	0.795016	3	0	0.434433
6	77	600	3	2	0.830522	3	0	0.473172
7	77	600	3	1	0.823751	3	0	0.437696
8	77	600	3	2	0.793194	3	0	0.440564
9	77	600	3	2	0.828482	3	0	0.438299
10	77	600	3	2	0.789525	3	0	0.440092
...								
95	80	200	3	1	0.824771	3	0	0.447373
96	80	200	3	2	0.801545	3	0	0.438209
97	80	200	3	1	0.817225	3	0	0.46187
98	80	200	3	1	0.829837	3	0	0.444896
99	80	200	3	1	0.814318	3	0	0.441267
100	80	200	3	2	0.804689	3	0	0.440008
101	80	200	3	1	0.80738	3	0	0.456504
102	80	200	3	1	0.79714	3	0	0.42743
103	300	184	4	3	0.715598	3	41	0.534311
104	300	184	4	3	0.716426	3	81	0.537627
105	300	184	5	3	0.746946	3	116	0.562513
106	300	184	4	3	0.677954	3	158	0.52671

Figure 8: Calculation result table 1

It was found that until 103 flights arrived at 6 pm, when the current flight arrived, the person on the pick-up point was waiting in line. That is, before 6 pm, all the time when the current flight arrived, it was found that there was no queue at the pick-up point. In order to minimize the total waiting time T_s of all passengers, the boarding point should be arranged with 3 on the L side, 2 on the R side, and the distance between the boarding points is 18m; the vehicle arrangement is k=3, that is, each batch is rented. There are 3 consecutive vehicles to the same boarding point; passengers arrange about 1/5 passengers to pass the bridge, go to the R side to wait for the car, and let the top two of the team get on the train at the same time.

After 6 pm, the traffic volume continues to increase and reaches the peak at 22-24, using the model at this time.

$$T_s = \frac{(d + k * \Delta d) * (n-1) * (B * 2 * n' + p)}{v * k} + c * \frac{2 * B * n' + p}{*} 4 * n * (2 * B * n' + p) \tag{54}$$

$$B = B' + \frac{p}{2 * n} - \frac{\Delta t * E_m}{2 * n} \tag{55}$$

$$v = v_1 * In[\frac{k_j}{k}] \tag{56}$$

The minimum distance between the vehicles on the d, considering that the boarding point is relatively crowded, the taxi speed limit is 20km / h, the safety distance is 10m, take d = 10, Δ d = 4. There are k vehicles in the same train at the time of each batch of vehicles. Considering safety and order, they should be as small as possible. The taxi speed v_1 considers the speed limit to be 5m/s, k_j is 1/3, k=k/(10+4*k), because the passenger flow is huge at this time, so the waiter is added to serve the passengers, carry the luggage and sit on the seat. The time c is shortened to 5s.

In order to find the boarding point arrangement n and the taxi arrangement k which minimize T_s, that is, the waiting time, a genetic algorithm is used for solving.

Substitute the model calculation and get. (The first column: the first flight on that day, the second column: the average number of taxis required for passengers on each flight, the third column: flight interval (seconds), the fourth column: the number of passengers on one side (n), the fifth column: the number of cars in the same boarding point in a batch of taxis (k), the sixth column: the number of people queued at the arrival point when the flight arrives (B), the seventh column: the maximum number of passengers per second in the model (E_m))

103	600	184	12	2	1	1.400548
104	600	184	12	2	2	1.399008
105	600	184	13	2	3	1.396627
106	600	184	13	2	4	1.396817
107	600	184	14	2	5	1. 39521
108	600	184	14	2	6	1. 39334
109	600	184	15	2	7	1. 39209
110	600	184	15	2	8	1.391731
111	600	184	16	2	9	1.39005
112	600	184	16	2	10	1.389761
113	600	184	17	2	11	1.388543
......						
221	458	153	28	2	17	1.379149
222	458	153	28	2	17	1.378787
223	458	153	28	2	17	1.378787
224	458	153	28	2	17	1.378834
225	458	153	28	2	17	1.378775
226	458	153	28	2	17	1.378943
227	458	153	28	2	17	1.379
228	458	153	28	2	17	1.378842
229	458	153	28	2	17	1.378929
230	458	153	28	2	17	1.379196
231	458	153	28	2	17	1.378852
232	458	153	28	2	17	1.378963
233	458	153	28	2	17	1.378946
234	458	153	28	2	17	1.379102
235	458	153	28	2	17	1.378803
236	458	153	28	2	17	1.378785

Figure 9: Calculation result table 2

It can be seen that after 6 pm, 28 parking spots need to be arranged on the side of the road. The distance between each parking point is 122 meters. If this arrangement is arranged, the waiting team will not increase any more when it is increased to a certain length. The length is 17, because the total number is 17*33*2, and the maximum number of passengers per second is 1.358. Therefore, the last passenger can take about 15 minutes to wait until the car, and the model service capability is acceptable.

By sorting the number of points printed on the above into a picture, you can see the number of boarding points that should be arranged for each time period.

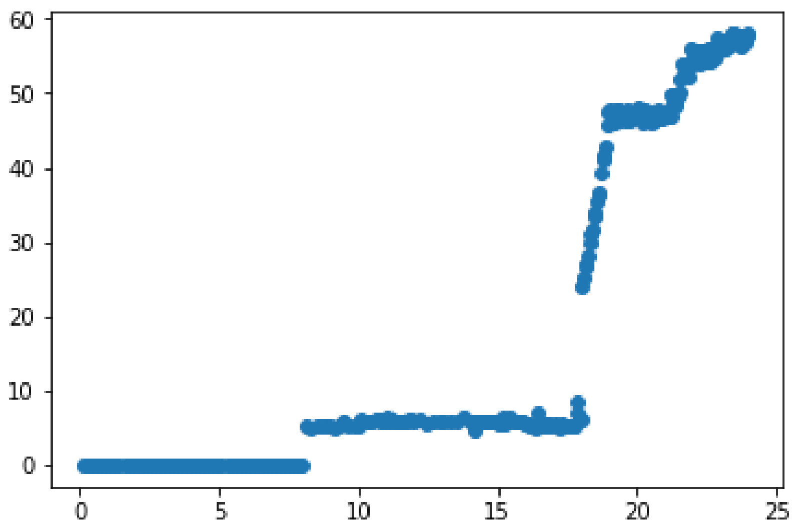


Figure 10: Boarding time period chart

0 to 8 o'clock, the total number of boarding points is 0; From 8:00 to 18:00, the total number of boarding points is 5, 3 on one side and 2 on the other side; From 18 o'clock to 22:00, the total number of boarding points is 46, 23 on one side and 23 on the other side; From 22 o'clock to 24:00, the total number of boarding points is 56, 28 on one side and 28 on the other.

7.3.3 Model rationality analysis

We found information about Pudong International Airport:

1. Passengers can get on the bus within an average of 20 minutes during peak hours, which is basically consistent with the model;

2. Pudong Airport arranged a "baggage ambassador" in uniform to help when the passenger flow was large, which was consistent with the model assumptions;

3. Pudong Airport will issue a card to the taxi driver to specify the driver's pick-up point to be driven, which is consistent with our model assumptions;

4. Pudong Airport arranged a corresponding pick-up point for each taxi, which is consistent with the model assumptions. Pudong International Airport has a maximum of 12 pick-up points, and we have calculated 56. The difference is so great. The reasons we analyzed are: the road at the pick-up point of Pudong International Airport can accommodate more than two vehicles, and the pick-up point Can be scattered in all directions of the airport; and the problem requires that the boarding point road can only accommodate two cars, and all the loading points are concentrated on one road, so our model has less space in the scheduling, in order to meet the same The magnitude of the passenger flow leads to deviations in the results.

8 Problem 4: Design for Priority Model

8.1 Construction of Priority Model

Generally speaking, the average income of long-distance taxis is higher than that of short-distance taxis. Therefore, the following priority schemes are formulated for short-distance taxis:

A short Lane will be added in the queuing area for taxis with short-distance passengers delivering last time. The other is the general queuing lane. The vehicles in the short Lane have priority to enter the loading area for reaching passengers, and the vehicles in the general lane can only enter the loading area for boarding passengers after all the vehicles in the short Lane have been arranged. Every time the taxi finishes loading passengers from the airport, there will be mileage record. When the airport staff determines that the mileage record is a short distance, the taxi can enter the short distance lane for priority.

The following analysis shows the change of taxi revenue after the implementation of the plan:

Assume the short-distance taxis returns to the airport every time after delivering short-distance passengers. They start to line up at time D, board at time E, finish delivering at time F, return to the airport at time G. The short-distance taxis continuously delivering the short-distance passengers i times in a cycle until get and deliver long-distance passenger, then the total revenue of the short-distance taxis is:

$$W_{st} = I_s \cdot i + I_l, i = 1, 2, 3, .., N_c \tag{57}$$

where I_s and I_l are the short-distance and long-distance taxi fares respectively, which are determined by the taxi charging standard and the mileage L_s and L_l, and the net income is:

$$W_s = W_{st} - 2 \cdot o \cdot i + O_l \tag{58}$$

O_s and O_l are respectively the fuel charge for short-distance and long-distance mileage:

$$O_s = o \cdot k + L_s \tag{59}$$

$$O_l = o \cdot k + L_l \tag{60}$$

The probability that the taxi will carry the short distance passengers for $i - 1$ times after the first time is:

$$P(x = i) = (1 - P_l)^{i-1} \cdot P_l \tag{61}$$

Obviously, the random variable i obeys the geometric distribution. Its expectation is:

$$E(X) = \frac{1}{P_l} \tag{62}$$

The total cycle time is the time of i times queues, plus the time of i round trips, plus the time of the last long-distance delivery:

$$t_{s2} = t_{DE} \cdot i + 2 \cdot i \cdot t_{EF} + t_l \tag{63}$$

t_{EF} and t_L are also related to mileage L_s and L_L, respectively:

$$t_{EF} = \frac{L_s}{\alpha \cdot \beta \cdot V_1} \tag{64}$$

$$t_l = \frac{L_l}{\alpha \cdot \beta \cdot V_1} \tag{65}$$

t_{DE} is obtained by clustering analysis of flight density and taxi density in this time period. So the average hourly revenue of short distance vehicles in the cycle becomes:

$$\bar{W_s} = \frac{I_s \cdot E(X) + I_l - 2 \cdot O_s \cdot E(X) - O_l}{t_{DE} \cdot E(X) + 2 \cdot E(X) \cdot t_{EF} + t_l} \tag{66}$$

It is set that the long-distance taxis will not return to the airport after delivering passengers. For the long-distance bus, one cycle is queuing, carrying and finishing delivering. Assume the long-distance taxis start to queue at time D, carry passengers at time E (the average queuing time of all the taxis is the same), and carry passengers at time H, then the average hourly income $\bar{W}_l$ of the long-distance taxi meets the requirements:

$$\bar{W}_l = \frac{I_l - O_l}{t_{DE} + t_{EH}} \tag{67}$$

where

$$O_l = o \cdot k + L_l \tag{68}$$

$$t_{EH} = t_l \tag{69}$$

Set the total number of taxis is N_C, and the proportion of long-distance vehicles is P_l. then, the average hourly income W_b for every taxi before the implementation of the plan is:

$$W_b = \bar{W}_l \cdot P_l + \bar{W}_s \cdot (1 - P_l) \tag{70}$$

The variance of the average revenue is:

$$S_b = N_c \cdot P_l \cdot (\bar{W}_l - W_b)^2 + N_c \cdot (1 - P_l) \cdot (\bar{W}_s - W_b)^2 \tag{71}$$

After the implementation of the plan, the cycle of the short distance vehicle is still i consecutive short distance cycle plus the last long distance cycle, and its net income is the same as before the implementation of the plan:

$$W_{s2} = W_s \tag{72}$$

After the first queue, the time of each queue is decreased, changes to:

$$t_{DE2} = (1 - P_l) \cdot t_{DE} \tag{73}$$

The total cycle time is the time of the first queue plus the time of the next $i - 1$ queues plus the time of i round trips plus the time of the last long-distance delivery:

$$t_{s2} = t_{DE} + t_{DE2} \cdot (i - 1) + 2 \cdot i \cdot t_{EF} + t_{EH} \tag{74}$$

So the average hourly revenue of short distance vehicles in the cycle becomes:

$$\bar{W}_{s2} = \frac{I_s \cdot E(X) + I_l - 2 \cdot O_s \cdot E(X) - O_l}{t_{DE} + t_{DE2} \cdot (E(X) - 1) + 2 \cdot E(X) \cdot t_{EF} + t_{EH}} \tag{75}$$

For a long-distance taxi, still assume the long-distance taxi does not turn back to the airport, the average revenue of one cycle of the long-distance taxi is:

$$\bar{W}_{l2} = \bar{W}_l \tag{76}$$

The average hourly revenue per taxi becomes:

$$W_a = \bar{W}_{l2} \cdot P_l + \bar{W}_{s2} \cdot (1 - P_l) \tag{77}$$

The variance of revenue becomes:

$$S_a = N_c \cdot P_l \cdot (\bar{W}_{l2} - W_a)^2 + N_c \cdot (1 - P_l) \cdot (\bar{W}_{s2} - W_a)^2 \tag{78}$$

If $S_b > S_a$, it shows that our priority scheme does make the revenue of all taxis in this time period more balanced.

8.2 Verification of Priority Model

Similarly, Pudong Airport is selected as the actual example. The scenario is the same as the previous example with the same parameter setting of table3.

For the time being, the value of P_l cannot be obtained. Therefore, we have set multiple candidate values for P_l : $0.6, 0.7, 0.8$, (set $P_l > 0.5$, because most airport passengers are long-distance passengers) and we observed multiple experimental results. And from these results, we can draw some conclusions:

```
Average profit for short-distance taxi before adopting our plan 69.34274065712705
Average profit for short-distance taxi after adopting our plan 76.540957400783
Average profit for long-distance taxi 98.92494691567406
Average profit for all taxis before adopting our plan 87.09206441225525
Average profit for all taxis after adopting our plan 89.97135110971763
Square difference before adopting our plan:  210.02566250957221
Square difference after adopting our plan:  120.25031678466065
```

Figure 11: Results when $P_l = 0.6$

```
Average profit for short-distance taxi before adopting our plan 74.41487683227767
Average profit for short-distance taxi after adopting our plan 80.75062409690918
Average profit for long-distance taxi 98.92494691567406
Average profit for all taxis before adopting our plan 91.57192589065514
Average profit for all taxis after adopting our plan 93.4726500700446
Square difference before adopting our plan:  126.15614245353053
Square difference after adopting our plan:  69.36426208334234
```

Figure 12: Results when $P_l = 0.7$

```
Average profit for short-distance taxi before adopting our plan 79.03610508698397
Average profit for short-distance taxi after adopting our plan 83.82435144810836
Average profit for long-distance taxi 98.92494691567406
Average profit for all taxis before adopting our plan 94.94717854993604
Average profit for all taxis after adopting our plan 95.90482782216093
Square difference before adopting our plan:  63.2905646858644
Square difference after adopting our plan:  36.4844773560105
```

Figure 13: Results when $P_l = 0.8$

1. This scheme can significantly reduce the variance of average revenue and make the revenue of all taxis as balanced as possible.
2. It is true that airport long-distance passenger transport is more profitable than short-distance passenger transport, so it is a scientific and reasonable measure to set up a priority scheme for short-distance vehicles.
3. This scheme improves the average revenue of short distance vehicles and all taxis to a certain extent.
4. With the increase of P_l, almost all taxis take long-distance passengers, so the variance of the overall income decreases.

9 Model Appraisal

9.1 The strengths of the model

1. The data reliability is high, and the generalization ability of the trained model is stronger because all kinds of factors are considered as much as possible.
2. From the verification results of the example, the selection of time window is reasonable, taking into account various main factors
3. Problem 3 model has two forms, which can be applied to different passenger flow.It does not have the situation that the passenger flow is large and the team length is infinite, and it has strong carrying capacity for large passenger flow.What's more, the model considers various security issues

9.2 The weaknesses of the model

1. Due to the limited amount of data collected, there is no analysis of holidays, seasons and other factors, and the analysis of weather factors is not detailed enough
2. The airport flight data of problem 2 has no international flight, and the data is missing between 00:00-8:00, which makes the decision model unable to make decisions for some time periods.
3. The problem 3 model does not consider the differences of each passenger, different passengers may have different boarding times, and finally all passengers are the slowest one to wait. And this model needs a lot of boarding points when solving large passenger flow, and there may be a better model scheme.

9.3 The improvement of the model

1. By adding sample data and analyzing the data of airport and taxi for one month or even one year, we can consider more factors and improve the robustness of the model
2. Problem 3 model can further consider the difference of passengers and calculate the expectation of the maximum boarding time of all passengers from the perspective of probability.

9.4 The Extension of the model

1. In addition to the airport, the taxi decision-making model and the plan of the loading area of each high-speed railway station and railway station can refer to this model for rapid transplantation.
2. In addition to the decision-making model of airport boarding point, the problem 3 model can also be applied to the maximum utility required by many to many service model, such as the cash register of large shopping malls, etc.

10 Conclusion

1. In this paper, We have established a total of three mathematical models.
2. For question 1, 2. We consider the driver's profit for the purpose, so use the airport and taxi data, through the gathering

   Class analysis, approximation, probability statistics and other methods can quantify each influencing factor and obtain a decision model.
3. For question 3, we have to consider that the setting of the boarding point should make passengers ride the most efficient. Therefore, we will get the number and interval of boarding points and how many taxis. And how many passengers correspond to a boarding point to establish a dynamic queuing service model for time and passenger traffic Type, the solution method uses genetic algorithm. The reason is different from the actual plan.
4. For question 4, we provide a priority plan: add a short-distance lane in the queuing area for the upper The passengers in the short-distance lanes are queued for the passengers in the short-distance lanes. For the program Validity, we conducted modeling validation.

References

[1] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[2] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[3] L. Lingjiang, "Urban highway traffic and traffic flow models [j]," *Mechanics and Engineering*, vol. 1, 2005.

[4] C. Dombry, "A weighted random walk model, with application to a genetic algorithm," *Advances in applied probability*, vol. 39, no. 2, pp. 550–568, 2007.

Appendices

Appendix A Source Codes

Here are programmes we used in our model as follow.

K-means source code:

```
import numpy as np
import random
import re
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D
def show_fig():
    dataSet = loadDataSet()
    fig = plt.figure()
    ax = fig.add_subplot(111)
```

```
    ax.scatter(dataSet[:, 0], dataSet[:, 1])
    plt.show()

def calcuDistance(vec1, vec2):
    if vec1[0] == vec2[0]:
        d = 0
    else:
        d = 1
    dis = np.sqrt(np.sum(np.square(vec1[1:] - vec2[1:]))) + 0.15*d
    return dis

def loadDataSet():
    dataSet = pd.read_csv("final_data_set.csv").values
    dataSet = random.sample(list(dataSet), 1000)
    return dataSet

def initCentroids(dataSet, k):
    dataSet = list(dataSet)
    return random.sample(dataSet, k)

def minDistance(dataSet, centroidList):
    clusterDict = dict()
    k = len(centroidList)
    for item in dataSet:
        vec1 = item
        flag = -1
        minDis = float("inf")
        for i in range(k):
            vec2 = centroidList[i]
            distance = calcuDistance(vec1, vec2)
            if distance < minDis:
                minDis = distance
                flag = i
        if flag not in clusterDict.keys():
            clusterDict.setdefault(flag, [])
        clusterDict[flag].append(item)
    return clusterDict

def getCentroids(clusterDict):
    centroidList = []
    for key in clusterDict.keys():
        centroid = np.mean(clusterDict[key], axis=0)
        centroidList.append(centroid)
    return centroidList


def getVar(centroidList, clusterDict):
    sum = 0.0
    for key in clusterDict.keys():
        vec1 = centroidList[key]
        distance = 0.0
        for item in clusterDict[key]:
            vec2 = item
            distance += calcuDistance(vec1, vec2)
        sum += distance
    return sum

def showCluster(centroidList, clusterDict,ang1,ang2):
    colorMark = ['r', 'b', 'g', 'k', 'y', 'w','m']
    fig = plt.figure()
    ax = Axes3D(fig)

    for key in clusterDict.keys():
```

```
        ax.scatter(centroidList[key][0], centroidList[key][1], centroidList[key][2],
            marker = '^', s=30, c=colorMark[key])
        for item in clusterDict[key]:
            ax.scatter(item[0], item[1],item[2], c=colorMark[key],s=6)
    ax.view_init(ang1,ang2)

def test_k_means(k):
    dataSet = loadDataSet()
    centroidList = initCentroids(dataSet, k)
    clusterDict = minDistance(dataSet, centroidList)
    newVar = getVar(centroidList, clusterDict)
    oldVar = 1
    times = 2
    while abs(newVar - oldVar) >= 0.00001:
        centroidList = getCentroids(clusterDict)
        clusterDict = minDistance(dataSet, centroidList)
        oldVar = newVar
        newVar = getVar(centroidList, clusterDict)
        times += 1
        #showCluster(centroidList, clusterDict)
    return centroidList,clusterDict
```

Model2 source code:

```
import pandas as pd
import numpy as np
import os
from k_means import calcuDistance,test_k_means,showCluster
L1 = 46.4
L2 = pd.read_csv('dis_city.csv')
V1 = 64.7
V2 = pd.read_csv('velocity.csv')
o = 8.8/100.0
k = 7.13
tCity = pd.read_csv('./city_avg_waiting_time_new.csv')

def cal_extra_Income(te,tw,avgD,v):
    return (te/(tw+avgD/v))*cal_I(avgD,L1,'urban') - (te/(tw+avgD/v))*avgD*o*k

def cal_I(L2,L1,t_type):
    a = 3
    b = 15
    f1 = 14
    f2 = 2.5
    f3 = 3.6
    if t_type == 'airport':
        return  f1 + f2*(b-a) + f3*(L1-b)
    if L2 <= a:
        return f1
    elif a<L2 and L2 <=b:
        return f1+(L2-a)*f2
    else:
        return f1 + f2*(b-a) + f3*(L2-b)

def cal_O2(L1):
    return  L1*o*k

def cal_time_c_a(L1,L2,V2,tw):
    a = 1
    b = 1
    y = 1
    return a*b*(L1/V1 + L2/V2) + y*tw
```

```
def cal_W2(tw,V2,avgD,t1):
    L2 = avgD
    L1 = 46.4*V2/50
    I2 = cal_I(L2,L1,'urban')
    Ie = cal_extra_Income(t1-L1/V1,tw,avgD,V2)
    O = cal_O2(L1)
    time = t1#cal_time_c_a(L1,L2,V2,tw)
    print("I2 %f O %f Ie %f time %f"%(I2,O,Ie,time))
    return (I2 - O + Ie)/time

def cal_W1(T,Na,Nc,centroidList):
    a = 1
    b = 1
    I1 = cal_I(L2,L1,'airport')
    O1 = L1*o*k
    ts = a*b*L1/V1
    tq = estimate_waiting_time(T,Na,Nc,centroidList)
    print("I1 %f O1 %f ts %f tq %f"%(I1,O1,ts,tq))
    return (I1 - O1)/(ts + tq),ts + tq
def stander(Na,Nc,data_set):
    x = data_set['flight_density']
    Na = (Na-np.min(x))/(np.max(x)-np.min(x))
    x = data_set['taxi_density']
    Nc = (Nc-np.min(x))/(np.max(x)-np.min(x))
    return Na,Nc
def get_origin_waiting_time(t,data_set):
    x = data_set['wait_time']
    t = t*(np.max(x)-np.min(x))+np.min(x)
    print("estimate T %f"%(t))
    return t*13.0/60.0/60.0
def estimate_waiting_time(T,Na,Nc,centroidList):
    data_set = pd.read_csv('data_set_simplify.csv')
    Na,Nc =  stander(Na,Nc,data_set)
    vec1 = np.array([T,Na,Nc])
    judgeList = []
    for i in range(0,len(centroidList)):
        dis = calcuDistance(vec1, centroidList[i][0:-1])
        judgeList.append(dis)
    min_index = judgeList.index(min(judgeList))
    return get_origin_waiting_time(centroidList[min_index][-1],data_set)

def get_decision(centroidList):
    #data_set =  pd.read_csv('data_set_simplify.csv')
    #data = data_set.sample(n=50)
    data = pd.read_csv('input.csv')
    count = 0
    w1List = []
    w2List = []
    for i in range(0,50):
        temp = data.iloc[i]
        print("time %s T %d Na %f Nc %f"%(temp['time'],int(temp['t_label']/2),temp['
            flight_density'],temp['taxi_density']),end=" ")
        w1,t1 = cal_W1(int(temp['t_label']/2),temp['flight_density'],temp['
            taxi_density'],centroidList)

        v = V2.iloc[int(temp['t_label'])]['velocity']
        tw = tCity.iloc[int(temp['t_label'])]['avg_time(min/h)']/60.0
        avgD = L2.iloc[int(temp['t_label'])]['avg_dis']
        print("tw: %f v:%f avgD %f"%(tw,v,avgD),end = " ")
        w2 = cal_W2(tw,v,avgD,t1)
        print("w1 %f w2 %f"%(w1,w2))
        print("")
        if w1 < w2:
```

```
            count+=1
        w1List.append(w1)
        w2List.append(w2)
    data.insert(6,'w1',w1List)
    data.insert(7,'w2',w2List)
    return count,data
def get_best_desision():
    total = []
    cen = {}
    clu = {}
    for k in range(4,8):
        centroidList,clusterDic = test_k_means(k)
        total.append(get_decision(centroidList))
        cen[k] = centroidList
        clu[k] = clusterDic
    return total,cen,clu
if __name__=='__main__':
    total,cen,clu= get_best_desision()
    print(total)
```

Model4 source code:

```
Ll = 46.4
Ls = 15
tde = 0.576014
pl = 0.9
Ex = 1/pl
tde2 = tde * (1 - pl)

def Income(L):
    f1 = 14
    f2 = 2.5
    f3 = 3.6
    a = 3
    b = 15
    if L <= a:
        return f1
    elif L <= b:
        return f1 + (L - a) * f2
    else:
        return f1 + f2 * (b - a) + f3 * (L - b)


def OilCost(L):
    o = 0.088
    k = 7.13
    return L * o * k


def TimeOnTrip(L):
    alpha = 1
    beta = 1
    v = 64.7
    return L/(alpha * beta * v)

def AverageOnShort1(Ll,Ls):
    return (Income(Ls) * Ex + Income(Ll) - 2 * OilCost(Ls) * Ex - OilCost(Ll)) / (
        tde * Ex + 2 * Ex * TimeOnTrip(Ls) + TimeOnTrip(Ll))

def AverageOnLong(L):
    return(Income(L) - OilCost(L))/(tde + TimeOnTrip(L))

def AverageBefore(Ll,Ls):
```

```
    return (AverageOnLong(Ll) * pl + AverageOnShort1(Ll,Ls) * (1 - pl))


def SquareDiffBefore(Ll,Ls):
    return (pl * (AverageOnLong(Ll) - AverageBefore(Ll,Ls)) ** 2 + (1 - pl) * ((
        AverageOnShort1(Ll,Ls) - AverageBefore(Ll,Ls))) ** 2)

def AverageOnShort2(Ll,Ls):
    return (Income(Ls) * Ex + Income(Ll) - 2 * OilCost(Ls) * Ex - OilCost(Ll))/(tde
        + tde2 * (Ex - 1) + 2 * Ex * TimeOnTrip(Ls) + TimeOnTrip(Ll))

def AverageAfter(Ll,Ls):
    return (pl * AverageOnLong(Ll) + (1 - pl) * AverageOnShort2(Ll,Ls))

def SquareDiffAfter(Ll, Ls):
    return (pl * (AverageOnLong(Ll) - AverageAfter(Ll,Ls)) ** 2 + (1 - pl) * ((
        AverageOnShort2(Ll,Ls) - AverageAfter(Ll,Ls))) ** 2)

print("Average profit for short-distance taxi before adopting our plan",
    AverageOnShort1(Ll,Ls))
print("Average profit for short-distance taxi after adopting our plan",
    AverageOnShort2(Ll,Ls))

print("Average profit for long-distance taxi",AverageOnLong(Ll))

print("Average profit for all taxis before adopting our plan",AverageBefore(Ll,Ls))
print("Average profit for all taxis after adopting our plan",AverageAfter(Ll,Ls))

print("Square difference before adopting our plan: ",SquareDiffBefore(Ll,Ls))
print("Square difference after adopting our plan: ",SquareDiffAfter(Ll,Ls))
```