

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 queueing 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 queueing 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	4
1.1	Background	4
1.2	Tasks	4
2	Model Analysis	4
2.1	Analysis of task 1	4
2.2	Analysis of task 2	5
2.3	Analysis of task 3	5
2.4	Analysis of task 4	6
3	Model Assumptions	6
4	Parameter Table	7
5	Problem 1: Design for Decision Model	8
5.1	Net Income for Queueing Taxis	8
5.2	Net Income for Returning Taxis	9
6	Problem 2: Decisions for Drivers and Analysis for the Decision Model	10
6.1	Data Preparing	10
6.2	Parameter Calibration	12
6.3	Model Calculation	12
6.4	Rationality Analysis	14
6.5	Dependency Analysis	15
7	Problem 3: Design for Schedule Model	16
8	Problem 4: Design for Priority Model	16
8.1	Construction of Priority Model	16
8.2	Verification of Priority Model	18
9	Model Appraisal	19
9.1	The strengths of the model	19
9.2	The weaknesses of the model	19
9.3	The improvement of the model	20
9.4	The Extension of the model	20

10 Conclusion	20
Appendices	21
Appendix A Source Codes	21

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	Definition
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:

$$W_1 = I_1 - O_1 \quad (1)$$

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:

$$W_2 = I_2 - O_2 - O_e \quad (2)$$

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:

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

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:

$$O_1 = L_1 \cdot o \cdot k \quad (4)$$

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

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

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 α ($0 < \alpha \leq 1$), 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 β , ($0 < \beta \leq 1$). 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:

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

And we can conclude that:

$$t_s = \frac{L_1}{\alpha \cdot \beta \cdot V_1} \quad (7)$$

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:

$X = \{X_1, X_2, \dots, X_n\}$ 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:

$$x_{ik}' = \frac{x_{ik} - \bar{x}_k}{s_k} \quad (8)$$

where:

$$\bar{x}_k = \frac{1}{n} \cdot \sum_{i=1}^n x_{ik} \quad (9)$$

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

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

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

For classification attribute T , Hamming distance is used:

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

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

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

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

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

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 :

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

The calculation of O_2 is similar to that of O_1 :

$$O_2 = L_2 \cdot o \cdot k \cdot N_y \quad (16)$$

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:

$$O_{e1} = L_1 \cdot o \cdot k \quad (17)$$

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:

$$O_{e2} = t_w \cdot V_2 \cdot o \cdot k \cdot N_y \quad (18)$$

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

$$t_p = \frac{L_2}{\alpha \cdot \beta \cdot V_2} \quad (19)$$

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 γ , ($\gamma > 0$). From this, we can get the approximate value of N_y :

$$N_y = \frac{t_q}{\gamma \cdot t_w + t_p} \quad (20)$$

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

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

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.

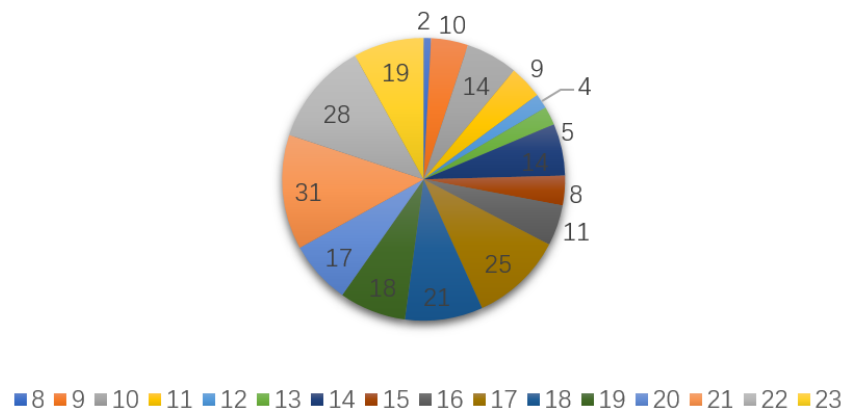


Figure 1: Flight data of Pudong Airport 20180602 obtained by crawling

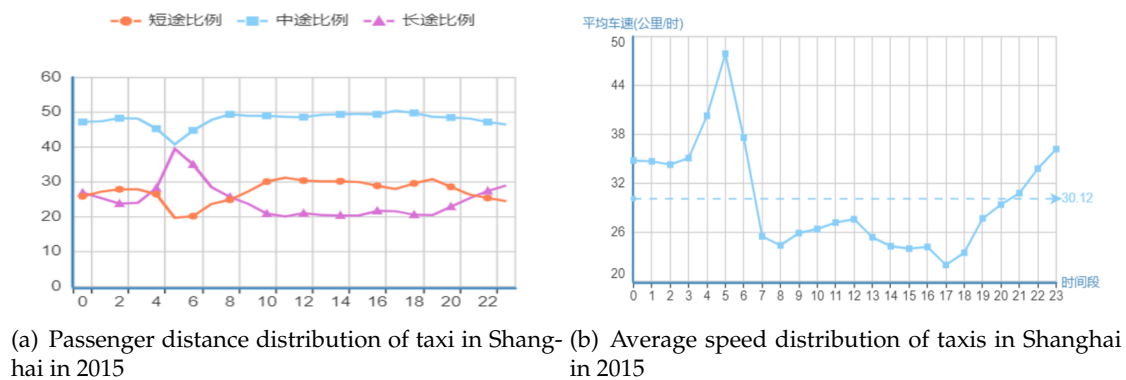


Figure 2: Taxi Data

2. Processed Data

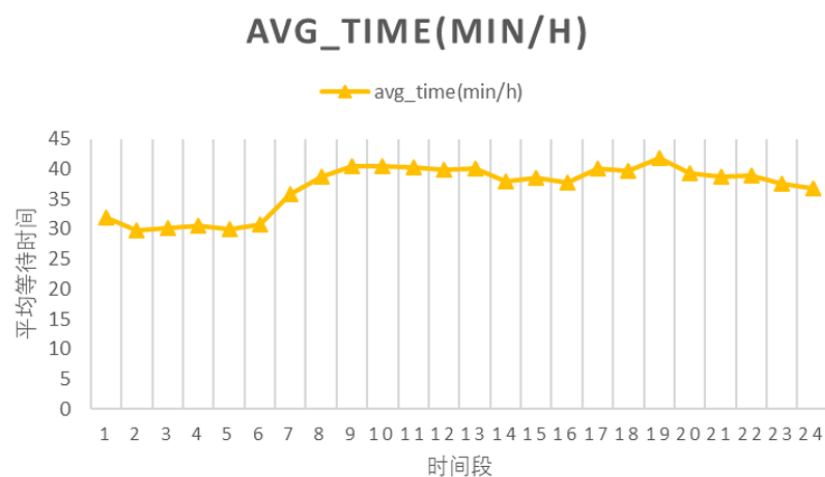


Figure 3: Average waiting time of receiving orders in urban areas in different time periods

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:

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

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:

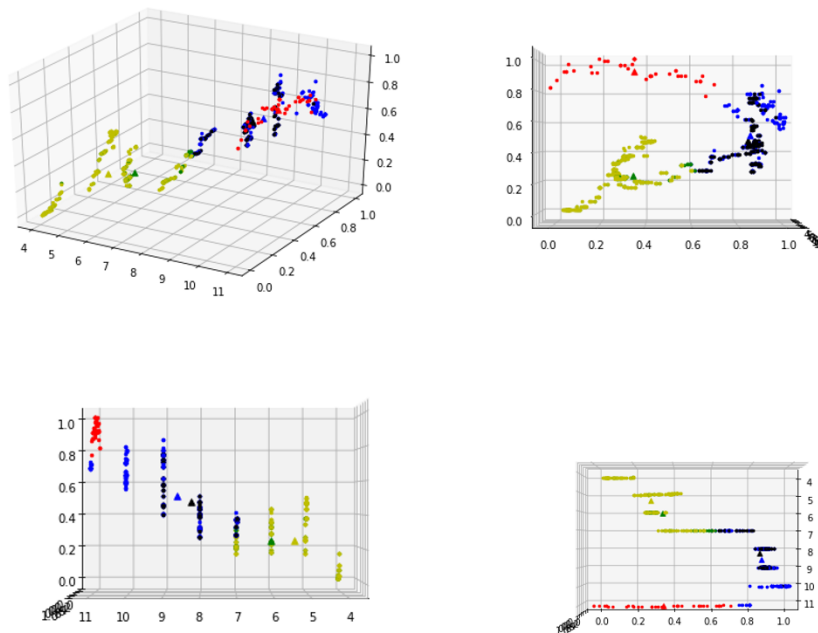
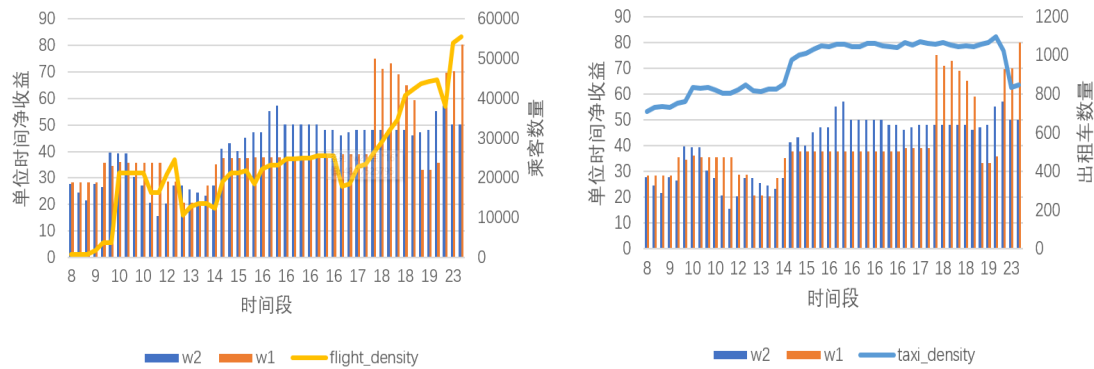


Figure 4: Distribution of Clustering Results

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 contour coefficient combining the two factors of cohesion and separation, that is, for each vector in K clusters, the contour coefficient[2] of each vector is calculated by the following formula:

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

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 :

$$s = \frac{C_{i_max} - C_{i_min}}{10} \quad (23)$$

That is $\{C_{i_min}, C_{i_min} + s, C_{i_min} + 2s, \dots, C_{i_max}\}$, and the remaining five parameters are set to average value. Then bring these variables into the decision model, so that we can get $\{d_{w_i_1}, d_{w_i_2}, \dots, d_{w_i_10}\}$ 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:

$$r_{w_i_k+1} = \frac{\frac{|d_{w_i_k+1} - d_{w_i_k}|}{d_{w_i_k}}}{\frac{s}{C_{i_min} + ks}} \quad (24)$$

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:

$$r_w = \begin{bmatrix} r_{w_1_1} & \cdots & r_{w_1_10} \\ \vdots & \ddots & \vdots \\ r_{w_6_1} & \cdots & r_{w_6_10} \end{bmatrix} \quad (25)$$

Finally, the average value of each line is obtained:

$$r_{w_mean} = \begin{bmatrix} r_{w_1_mean} \\ \vdots \\ r_{w_6_mean} \end{bmatrix} \quad (26)$$

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

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:

$$W_{st} = I_s \cdot i + I_l, i = 1, 2, 3, \dots, N_c \quad (27)$$

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:

$$W_s = W_{st} - 2 \cdot o \cdot i + O_l \quad (28)$$

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

$$O_s = o \cdot k + L_s \quad (29)$$

$$O_l = o \cdot k + L_l \quad (30)$$

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

$$P(x = i) = (1 - P_l)^{i-1} \cdot P_l \quad (31)$$

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

$$E(X) = \frac{1}{P_l} \quad (32)$$

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:

$$t_{s2} = t_{DE} \cdot i + 2 \cdot i \cdot t_{EF} + t_l \quad (33)$$

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

$$t_{EF} = \frac{L_s}{\alpha \cdot \beta \cdot V_1} \quad (34)$$

$$t_l = \frac{L_l}{\alpha \cdot \beta \cdot V_1} \quad (35)$$

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:

$$\bar{W}_s = \frac{I_s \cdot E(X) + I_l - 2 \cdot O_s \cdot E(X) - O_l}{t_{DE} \cdot E(X) + 2 \cdot E(X) \cdot t_{EF} + t_l} \quad (36)$$

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:

$$\bar{W}_l = \frac{I_l - O_l}{t_{DE} + t_{EH}} \quad (37)$$

where

$$O_l = o \cdot k + L_l \quad (38)$$

$$t_{EH} = t_l \quad (39)$$

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:

$$W_b = \bar{W}_l \cdot P_l + \bar{W}_s \cdot (1 - P_l) \quad (40)$$

The variance of the average revenue is:

$$S_b = N_c \cdot P_l \cdot (\bar{W}_l - W_b)^2 + N_c \cdot (1 - P_l) \cdot (\bar{W}_s - W_b)^2 \quad (41)$$

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:

$$W_{s2} = W_s \quad (42)$$

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

$$t_{DE2} = (1 - P_l) \cdot t_{DE} \quad (43)$$

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:

$$t_{s2} = t_{DE} + t_{DE2} \cdot (i - 1) + 2 \cdot i \cdot t_{EF} + t_{EH} \quad (44)$$

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

$$\bar{W}_{s2} = \frac{I_s \cdot E(X) + I_l - 2 \cdot O_s \cdot E(X) - O_l}{t_{DE} + t_{DE2} \cdot (E(X) - 1) + 2 \cdot E(X) \cdot t_{EF} + t_{EH}} \quad (45)$$

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:

$$\bar{W}_{l2} = \bar{W}_l \quad (46)$$

The average hourly revenue per taxi becomes:

$$W_a = \bar{W}_{l2} \cdot P_l + \bar{W}_{s2} \cdot (1 - P_l) \quad (47)$$

The variance of revenue becomes:

$$S_a = N_c \cdot P_l \cdot (\bar{W}_{l2} - W_a)^2 + N_c \cdot (1 - P_l) \cdot (\bar{W}_{s2} - W_a)^2 \quad (48)$$

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.

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 6: Results when $P_l = 0.6$

1. This scheme can significantly reduce the variance of average revenue and make the revenue of all taxis as balanced as possible.

```

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 7: 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 8: Results when $P_l = 0.8$

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

In this paper, we have constructed our models based on the biological features of EVD, social features of human society and several reasonable assumptions. Our models consist of two parts: one is considering the the spread of disease within a single city and serves as the base of the other; the other takes the people flow among the cities into account, the application of which gives an optimized plan regarding how should we allocate the resources of medication such as vaccine.

Both of the models are applied to specific cases separately, and the results of computation which are carefully studied justified our model. Through our analysis of the model, we explored and explained the complex relationship among numerous variables and parameters.

The effectiveness of medical treatment (including segregation, vaccination and pharmacotherapy) is verified by our model and the strategy to allocate vaccine and drug is revealed by our investigation.

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] Fan Miao, Li Chao. "Python machine learning and practice:the road from scratch to the kaggle competition", 2019

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, angl, 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(angl, 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:

```

L1 = 46.4
Ls = 15
tde = 0.576014
p1 = 0.9
Ex = 1/p1
tde2 = tde * (1 - p1)

```

```

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))

```
