Study on Taxi Driver Decision-Making of Beijing Capital International Airport Based on M / M / S model

Baiyan Xu¹, Jiawei Liu², Long Chen², Jiabin Deng²

¹ Department of science, Xinjiang Institute of Technology, Aksu 843100, China
baiyanxu23@126.com
² School of Electrical Information, Changchun Guanghua University, Changchun 130033, China
ccliujiawei@163.com

Abstract. Based on the ideas of queuing theory and data fitting, this paper uses Dev-cpp software to write C++ programs and establishes a driver decision model. Firstly, the selection scheme of taxi drivers for Beijing Capital International Airport is given, and the rationality of the model and its dependence on related factors are analyzed using Dev-cpp software programming. Secondly, through the M / M / S model in the queuing theory model, it is verified that the "boarding point" arrangement scheme with the highest total boarding efficiency can be achieved under the premise of ensuring safety.

1. Introduction
In recent years, taxi has become the main tool for passengers to travel in large and medium-sized cities in China. Most of the passengers in the airport will choose to leave the airport by taxi. Therefore, it is necessary to effectively improve the timeliness and accuracy of the airport management department's response to the continuous transfer demand of arriving passengers.

Most domestic airports are separated from the pick-up (arrival) channel. Taxi drivers who see people off to the airport will face two choices:
(A) It takes a certain time cost to go to the arrival area and wait for passengers to return to the city.
(B) The taxi driver will pay the no-load cost and may lose the potential passenger income.

2. Model assumptions and symbol descriptions
(1) The working time of taxi operation is 24 hours.
(2) There is no traffic jam on the taxi.
(3) The driver's income in the city is continuous and there is no idle situation.
(4) Take a taxi by yourself.

| Symbol | Variable names in the program | Type | Meaning |
|--------|------------------------------|------|---------|
| R_a    | airportSolicitEarning        | variable | Airport solicitation revenue (yuan / time) |
| S_u    | solicitUnit                 | constant | Unit price in the city (yuan / half hour) |
| T_o    | getOnTaxiTime               | constant | Boarding time (minutes) |
| T_i    | intervalTime                | constant | Interval time (minutes) |
Symbol | Variable names in the program | Type | Meaning |
--- | --- | --- | --- |
$pN$ | passengerFlowVolume | variable | Passenger flow (waiting area) |
$tN$ | airportTaxiNum | variable | Number of taxis (waiting area) |
$cR$ | curRank | variable | Current rank of this taxi |
$wT$ | waitTime | variable | Waiting time (minutes) |
$cT$ | waitTimeCost | variable | Cost of waiting time (yuan) |
$pN$ | citySolicitEarning | variable | Back to the city soliciting income (yuan) |
$eA$ | AEarning | variable | Benefit A |
$eB$ | BEarning | variable | Benefit B |
$tc$ | curTime | variable | Current time (minutes) |
$C_T$ | calcWaitTime() | function | Calculating wait time |

3. Model one establishment and solution

Taking Beijing Capital International Airport as an example, by collecting the passenger flow number of the waiting area and the number of taxis in the parking pool at different times within a few days of the Capital Airport and performing a second-fit operation, the passenger flow number and storage of the waiting area at different times of the day in the capital airport are obtained. The number of taxis in the car pool. The line chart of the number of passengers in the waiting area after fitting is shown in Fig.1.

![Fig.1 Line chart of passenger flow in the waiting area](image)

The distance between the capital airport and the center of Beijing (Tian'anmen Square) is measured by the Baidu map software. This distance is regarded as the distance that the taxi driver returns to the city after pulling from the airport to the passengers. According to the Beijing taxi billing method, the taxi income is $R_v=92$ yuan.

According to the collected data, the average monthly turnover of taxis in Beijing is 17,280 yuan, so the revenue of taxis per half hour is $S_v=12$ yuan.

Now suppose the current time: $cT=60$, the current passenger flow: $pN=28$, and the current number of taxis in the storage pool: $tN=34$. Since $N_r<N_v$, the waiting time can be derived from the model:

$$T_v=N_r * T_e + C_T = 28 + 37 = 37$$

Among them, the function $C_T$ calculates the time required to wait 9 minutes after updating the current number of passengers. So wait for the cost: $C_w = T_w * S_v = 37 * 12 / 30 = 14.8$ yuan.

Final benefit of scheme A: $R_A = R_v - C_w = 92 - 14.8 = 77.2$ yuan.

Return to soliciting income in the city: $R_v = T_v * S_v = 37 * 12 / 30 = 14.8$ yuan.

Final benefit of scheme B: $R_B = R_v = 14.8$ yuan.

Because of $R_A > R_B$, it is recommended that the taxi driver choose Option A.
Based on this model, a set of C++ program is written by Dev-cpp software, including a source program file and a configuration file, as shown in Fig. 2.

```c++
#include "calcABEarning.h"

/* Function: Calculate waiting time *
   taxiProblem::calcWaitTime()
{
    double res = 0;
    if (curRank != 1)
      if (passengerFlowVolume>=
curRank)
        {res = curRank * 
          getOnTaxiTime;
        }
      else
      if (curRank ==
pasengerFlowVolume; res += 
  intervalTime;
  curTime = (curTime + 30) % 1440;
  passengerFlowVolume = 
pasFlowVolForecast[curTime / 30];
  res += 
calcWaitTime();
  }
  else res = getOnTaxiTime;
  return res;
}

/* Function: Calculate the return of *
   choice A *
   taxiProblem::calcAEarning()
{
  double res;
  waitTimeCost = waitTime / 30.0
  * solicitUnit;
  res = airportSolicitEarning - 
  waitTimeCost;
  return res;
}

/* Function: Calculate the return of *
   choice B *
   taxiProblem::calcBEarning()
{
  double res;
  citySolicitEarning = waitTime / 
  30.0 * solicitUnit;
  
  res = citySolicitEarning;
  return res;
}

void taxiProblem::getConfig()
{
  FILE* fp = 
    fopen("calcABEarning/config.ini", "r");
  if (fp == nullptr)
    {
    printf("config.ini: Reading data failed! \n");
    return;
    }
  fscanf(fp, "curTime:\n");
  fscanf(fp, "%d\n", 
    &curTime);
  fscanf(fp, "tempPasFlowVol:\n");
  fscanf(fp, "%d\n", 
    &tempPasFlowVol);
  fscanf(fp, 
    "tempAirTaxiNum:\n");
  fscanf(fp, "%d\n", 
    &tempAirTaxiNum);
  fscanf(fp, 
    "pasFlowVolForecast:\n");
  fscanf(fp, "%d\n", 
    &pasFlowVolForecast);
  for (int i = 0; i< 48; i++)
    fscanf(fp, 
      ",%d\n", 
      &pasFlowVolForecast[i]);
  fscanf(fp, "\n");
  fclose(fp);
}

void taxiProblem::start()
{
  doubleAEarning, BEarning;
  passengerFlowVolume = 
  airportTaxiNum = 0;
  getConfig();
  printf("Current time:%ld\n", curTime / 60,
  curTime % 60);
  printf("Current passenger flow in waiting 
  areas:%ld\n", passengerFlowVolume);
  printf("Number of 
  taxis in the current storage 
  pool:%d\n", airportTaxiNum);
  curRank = airportTaxiNum + 
  1;
  waitTime = calcWaitTime();
  Aearning = calcAEarning();
  Bearning = calcBEarning();
  printf("If\n using plan A, passengers can be 
pulled to the airport after about\n minutes \n, waitTime); 
  printf("(nA) head to 
  the arrival area and wait for 
  the passengers to return to the 
  city: \n");
  printf("Cost of
 waiting time:%lf yuan\n," 
  waitTimeCost);
  printf("Solicitation 
  revenue:%lf yuan\n"," 
  airportSolicitEarning);
  printf("Final 
  revenue:%lf yuan\n," 
  Bearning);
  printf("(nB) directly return to the 
  city to 
  solicit: \n");
  printf("Earnings in 
  the city within %lf 
  minutes:%lf yuan\n," 
  waitTime, citySolicitEarning);
  printf("Final 
  revenue:%lf yuan\n," 
  Bearning);
  printf("(n So: you 
  should choose)\n");
  if (AEarning>BEarning)
    printf("Option A: 
    Line up to the arrival area and 
    wait for the passengers to 
    return to the city \n");
  else
    printf("Option B: 
    Directly empty and return to 
    the city \n");
}
```

Fig. 2  C++ program and configuration file

The configuration file is as follows:
The current storage
pool:%d
, airportTaxiNum;
curRank = airportTaxiNum + 
1;
waitTime = calcWaitTime();
Aearning = calcAEarning();
Bearning = calcBEarning();
printf("\n using plan A, passengers can be pulled 
to the airport after about\n minutes \n, waitTime); 
printf("(nA) head to 
the arrival area and wait for 
the passengers to return to the 
city: \n");
printf("Cost of
 waiting time:%lf yuan\n," 
waitTimeCost);
printf("Solicitation 
revenue:%lf yuan\n"," 
airportSolicitEarning);
printf("Final 
revenue:%lf yuan\n," 
Bearning);
printf("(nB) directly return to the 
 city to 
solicit: \n");
printf("Earnings in 
the city within %lf 
minutes:%lf yuan\n," 
waitTime, citySolicitEarning);
printf("Final 
revenue:%lf yuan\n," 
Bearning);
printf("(n So: you 
should choose)\n");
if (AEarning>BEarning)
printf("Option A: 
Line up to the arrival area and 
wait for the passengers to 
return to the city \n");
else
printf("Option B: 
Directly empty and return to 
the city \n");

Cur Time: 60;
Temp Pas Flow Vol: 28;
Temp Air Taxi Num: 34;
Pas Flow Vol Forecast: 11, 32, 28, 25, 8, 6, 3, 0, 3, 2, 0, 7, 21, 3, 4, 8, 6, 8, 22, 8, 9, 17, 3, 48, 12, 22, 
19, 95, 38, 20, 8, 7, 14, 30, 25, 17, 18, 29, 138, 29, 18, 15, 10, 5, 72, 0, 27, 53.
The program operation results are shown in Fig.3.

![Fig.3 program running results](image)

4. Model two ‘riding point’ settings
Most of China's airports, railway stations, bus stations and other transportation hubs will set up taxi pick-up points accordingly. In solving the problem of taxi boarding points, we can consider quoting the "queuing theory model". The queuing theory model was first proposed by Danish electrical engineer Ireland (A.K.Erlang) and applied to solve the problem of automatic telephone design. The queuing theory is now widely used in berth design, machinery manufacturing, transportation systems and other systems. This question is based on the standard model model in queuing model and solving. There are the following formulas for the various parameters of the model:

- **Service intensity**: \( \rho = \frac{\lambda}{\mu} \);
- **Waiting area idle probability**: \( P_0 = 1 - \rho \);
- **Average number of passengers in the waiting area**: \( L_s = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho} \);
- **Average number of people queuing outside the waiting area**: \( L_q = \frac{\lambda^2}{\mu (\mu - \lambda)} = \frac{\rho^2}{1 - \rho} = L_s \rho \);
- **Average time spent by passengers in the waiting area**: \( W_s = \frac{1}{\mu - \lambda} \);
- **Average waiting time of passengers**: \( W_q = \frac{\lambda}{\mu (\mu - \lambda)} = W_s \rho \).

Where \( \mu \) is the average service vehicle per hour and \( \lambda \) is the average passenger flow.

In the queuing system, the objects that require service are called "customers", and the organizations engaged in service are called "service desks". After the customers reach the service desks, they may get services immediately or they may wait until they can use the service desks. Contrary to the problem of taxis for airport passengers, customers are passengers who want to take a taxi, and the service desk is the taxi pick-up point. Once there is a taxi in the storage pool, customers who need a taxi will be immediately served.

In the input process based on the airport problem queuing system, the customer source satisfies the persistence and infinity and the probability of the customer flow satisfies the Poisson distribution. In the queuing process, each individual customer is independent of each other, and the queuing rules satisfy the first come, first served rule.
Based on the collected data of Beijing Capital Airport, the total passenger flow in the daily waiting area is 1010, so the average passenger flow per hour is $\lambda = 1010 / 24 \approx 42$ person/hour. The total daily passenger flow of taxis is 1208, so the average service vehicle per hour is $\mu = 1208 / 24 \approx 50$ vehicles per hour. Assume that the airport currently has only one pick-up point, which is a single-aisle service desk.

Service intensity: $\rho = \lambda / \mu = 42 / 50 = 0.84$.

Waiting area idle probability: $P_0 = 1 - \rho = 0.16$;

Average number of passengers in the waiting area: $L_s = \frac{\lambda}{\mu - \lambda} = \frac{42}{1 - 0.84} = 5.25$;

Average number of people queuing outside the waiting area: $L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{1764}{400} = 4.41$;

Average time spent by passengers in the waiting area: $W_s = \frac{1}{\mu - \lambda} = \frac{1}{50 - 42} = 0.125$;

Average waiting time of passengers: $W_q = \frac{\lambda}{\mu(\mu - \lambda)} = W_s / \rho = 0.105$.

It can be seen from the question that the airport boarding area is two parallel lanes, so another airport boarding point can be set up on the other side of the road to ensure the safety of passengers. The airport boarding point is shown in Fig. 4.

![Fig.4 Airport boarding point](image)

After a new riding point is added, the model in the queuing model can no longer be used, and the model is used instead. The difference between the two models is that the service desk of the former can only be 1, while the service desk of the latter can be multiple. Each service desk is independent of each other, and the service rate of each desk is equal.

Therefore, the work intensity of each service desk after adding a new service desk is:

$\rho_s = \lambda / 2\mu = 0.42$

Waiting area idle probability: $P_0 = [1 + 0.84 + \frac{(0.84)^2}{2!(1-0.42)}]^{-1} = 0.41$;

Average number of passengers in the waiting area: $L_s = \frac{(0.84)^2 \times 0.42}{2!(1-0.42)} = 1.18$;

Average number of people queuing outside the waiting area: $L_q = 1.18 + 0.84 = 2.02$;

Average time spent by passengers in the waiting area: $W_s = \frac{L_s}{\lambda} = \frac{2.02}{42} = 0.048$;

Average waiting time of passengers: $W_q = \frac{L_q}{\lambda} = 0.061$.

By comparison, in the case of ensuring safety, parameters such as waiting time for passengers in the two setting point’s scheme are significantly lower than those in the single setting point, and the two parallel lanes given in the question are not suitable for setting three. In fact, if there are three or more
boarding gates, it will cause traffic congestion and bring safety risks to passengers. From this, it can be explained that the setting of the double boarding point can allow passengers to get to the taxi faster and improve the riding efficiency.

5. Conclusion
Finally, this paper combines the real-time data of the Capital International Airport to re-examine and verify the model, corrects human error and incorrect statistical data, and eliminates invalid values caused by various abnormal conditions.

Preparation of scheme design. After reasonable data analysis and calculation, and program programming, the best "boarding point" and "priority" arrangement schemes were cleverly set, which provided a lot of convenience for the airport management department, and also solved the "car and others" "People wait for the car" headache, eliminating potential safety hazards around the airport.

The model uses two objective functions, and uses different planning algorithms for different choices of the driver, which simplifies the optimization of the space complexity and makes the optimization more in line with the facts.

Acknowledgments
This research was financially supported by the Jilin Natural Science Foundation (JJKH20171027KJ); Foundation of Jilin Development and Reform Commission (2019C054-7); Key Foundation of Youth Scientific Research of Changchun Guanghua University (QNJJZD2019005).

References
[1] Jiang Huijuan, Yu Yang. Improved DBSCAN algorithm for fine extraction of hot spots for taxi passengers[J]. Geospatial Information, 2017,15(10):16-20 (in Chinese).
[2] Liu Yangdong, Tian Ye, Yuan Bo, Mao Wei. A Secret Recommendation Model Based on Vehicle Traffic[J]. Scientific Research Information Technology and Application, 2015,6(2):66-73 (in Chinese).
[3] Lu Jian, Wang Wei. Method for Determining Urban Taxi Ownership[J]. Transportation Engineering, 2004,20(1):92-95 (in Chinese).
[4] Sun Rongheng, Li Jianping. Basics of Queuing Theory[M]. Beijing: Science Press, 2002 (in Chinese).
[5] Hou Wen. The relationship between commonly used probability distributions[J]. Journal of Liaoning Normal University (Natural Science Edition), 2005,28(4):503-505 (in Chinese).
[6] Fan Hejun, Du Bo. Estimation of the Reasonable Number of Taxi in Megacities and the Realization Path——Taking Beijing as an Example[J]. Economic and Management Research, 2015,36(8):91-95 (in Chinese).
[7] Pan Zihua. Impact and strategy of taxi traffic on urban traffic [J]. Science and Technology Innovation Herald, 2008,100(21):167-168 (in Chinese).
[8] Yuan Changwei, Wu Qunqi, Wei Dali, Wu Dayong. Taxi market equilibrium mechanism and optimization model considering refusal of loading[J]. China Journal of Highway and Transport, 2014,27(6):91-97 (in Chinese).