

A Design of a Customized Bus-Booking Platform

Junxiang Cao (518021910574, cjjxqaq@sjtu.edu.cn), Sen Li (518021910608, 874049001@qq.com), Shiqu Wu (518021910665, 15759684259@163.com)

Department of Computer Science,
Shanghai Jiao Tong University, Shanghai, China

Abstract. The airport traffic problems caused by night flights reflect the need for a bus platform that can respond to orders. The new platform needs to be improved due to the problem of the carrying capacity and cost of taxi.

Our report is to design a static bus order planning scheme. Initially, we set up the virtual airport separately as the starting point of the order, and screen the orders to be calculated within the radius. And we use the $K - \text{means}$ clustering algorithm to get the corresponding destinations, and then use genetic algorithm to plan the bus route for the corresponding situation. After consulting the relevant data to determine all the unknown parameters, the profit is counted and compared with taxi.

Keywords: virtual airport, bus, orders, static route planning.

1 Introduction

1.1 Problem Background

With the rapid development of cities and the rapid expansion of population in recent years, the improvement and renewal of urban transportation systems have become a major problem. More and more cities are beginning to popularize the medium and long-distance transportation tools in cities such as subway and light rail. Bicycle sharing also solves many "last mile problems" during commuting, but some groups still have unmet transportation needs.

According to statistics from relevant departments, many passengers arrive at their destinations on late night flights, but due to the suspension of mainstream public transportation such as subway buses, they can only be forced to choose taxis, airport shuttle buses, or the emerging online car rental. However, the carrying capacity of this part of the transportation mode is not ideal enough to satisfy all passengers who arrive at their destination late at night. Therefore, there are areas for improvement in the transportation system.

1.2 Our Work

At first, we focus on how to reduce the amount of data used. Thus we want to consider setting up a virtual airport first, and then we use different numbers of destinations, and visualize them.

Then we consider how to design algorithm and dispatch buses to maximum the profit. Of course we do consider the passenger satisfaction factor. After numerical experiments, we have feasible solutions which can be compared with optimal solutions. Finally, we compare it with taxi.

2 Assumption

In order to simplify our design work, we make the following assumptions:

- Only those data included in virtual airport which we use a circle to build. And all the orders in the circle are considered to start in the at the center of the circle.
- For each passenger i , the bus fare is decided by the linear distance from departure to destination.

3 Nomenclature

In this project we use the nomenclature in Table 1 to describe our design.

Table 1: Nomenclature

Symbol	Definition	Unit
d_i	Destination of passenger i	—
L	Capacity limitation of each bus	—
c_b	Bus fuel cost per kilometer	yuan
c_r	Cost per route of a bus	yuan
p_i	Bus fare price to station d_i	yuan
r_i	Route length of bus i	kilometer
P	Profit	yuan

4 Design Task with Numerical Experiments

4.1 Virtual Airport Selection

First we need to set virtual airport based on the departure location of orders.

1. **Chengdu.** In Chengdu, it is easy to determine where the virtual airport should be. Chengdu is a city centered on Tianfu Square. Therefore it is very suitable to set up a virtual airport with Tianfu Square as the center. Fig.1 shows the obvious center of Chengdu both in geography and in our orders. We set virtual airport with a radius of 3.4 kilometers.
2. **Haikou.** In Haikou, it is not obvious to determine the virtual airport as shown in Fig.2. So we first pick 5 high order density landmark, which is shown in Table.2.

Table 2: 5 Landmarks With High Order Density

Landmark	Value	Column	Longitude(°E)	Latitude(°N)
Hainan Provincial Government			110.3543	20.0241
Hainan Medical College			110.3366	20.0345
Haikou Intermediate People's Court			110.3348	20.0237
Haikou Hospital of traditional Chinese Medicine			110.3336	19.9941
Haikou Coach West Station			110.2976	20.0175

Then we calculate the average longitude and latitude of these 5 landmarks and get the location of the virtual airport which is (110.3314°E,20.0188°N). And then we screen out the orders in the virtual airport with a radius of 4 kilometers.

4.2 Destination Selection

To select some hot node as our bus destinations, we need to consider the destination density among all the orders we consider. In order to get a more representative destination, we decide to adopt $k - means$ clustering algorithm. To better explain it, we use pseudocode in Alg.1.

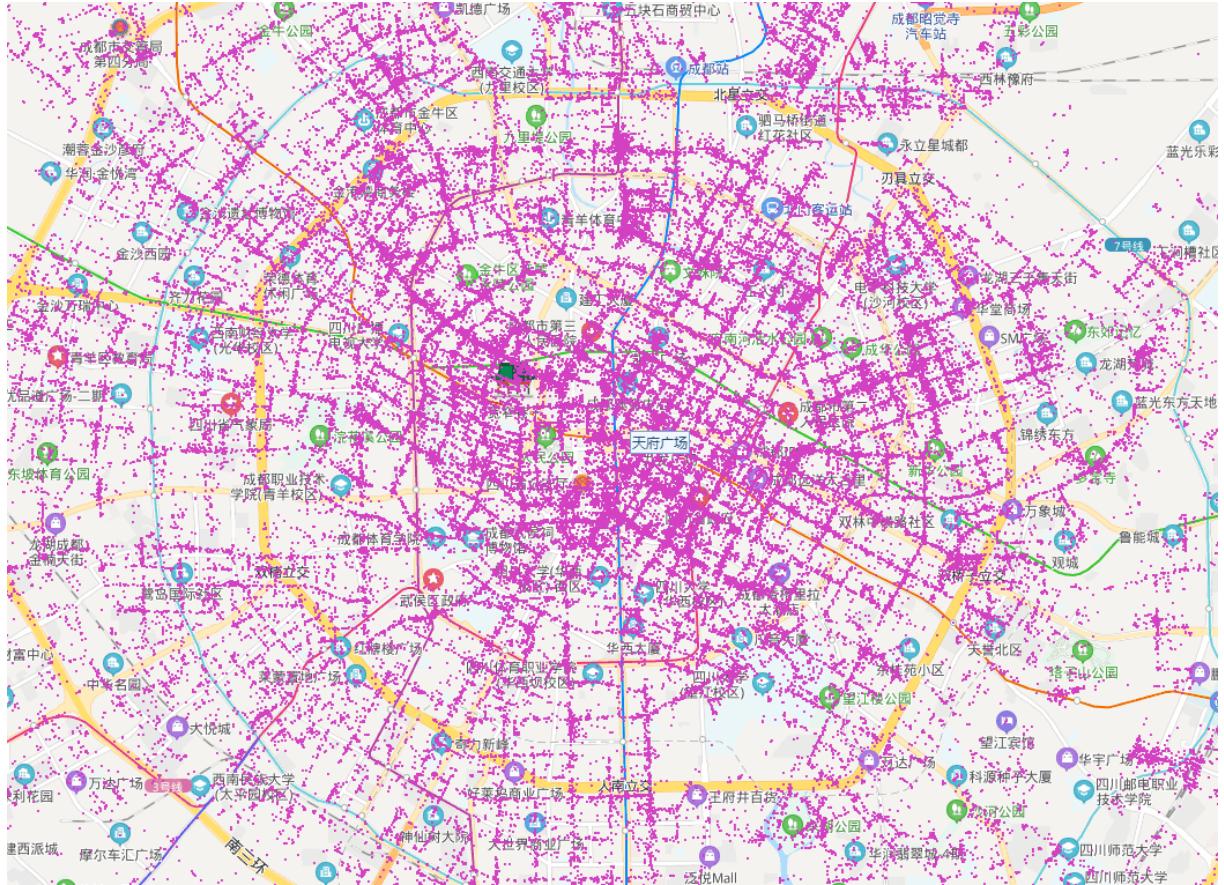


Fig. 1: Departure nodes map of Chengdu

Algorithm 1: $k - \text{means}$ clustering algorithm

Input: Number of samples n , cluster number k and initial cluster center

$$\mu_1, \mu_2, \dots, \mu_n$$

Output: Final clustering center (destinations) $\mu_1, \mu_2, \dots, \mu_n$

- ```

1 repeat
2 | assign the n samples according to the nearest μ_i ;
3 | recalculate cluster center $\mu_1, \mu_2, \dots, \mu_k$;
4 until $\mu_1, \mu_2, \dots, \mu_n$ do not change;
5 output final $\mu_1, \mu_2, \dots, \mu_k$;

```

After  $k - \text{means}$  cluster algorithm, we can get  $k$  hot destinations. There are two versions of  $k = 30$  and  $k = 50$ . And for both Chengdu and Haikou, we both get the final destinations in Fig.3.

### 4.3 Problem Formulation

Here we give the formal definitions of three ODRP problems.

- Suppose that there are  $n$  passengers and  $m$  buses at the departure station, while the destination station of passenger  $i$  is  $d_i \in D$  and the capacity limitation of each bus is  $L$ . The bus fuel cost per kilometer is  $c_b$ . The cost per route of a bus is  $c_r$ . The bus fare price for one passenger from the departure station to station  $d_j$  is  $p_j$ . Please dispatch each bus  $i$  with a route whose length is  $r_i$  kilometers to satisfy all passengers ( $n < mL$ ) and maximum the profit  $P = \sum_{i=1}^n p_i - \sum_{i=1}^m c_r - \sum_{i=1}^m (c_b \times r_i)$ .
- Suppose that there are  $n$  passengers and  $m$  buses at the departure station, while the destination station of passenger  $i$  is  $d_i \in D$  and the capacity limitation of each bus is  $L$ . The bus fuel cost per

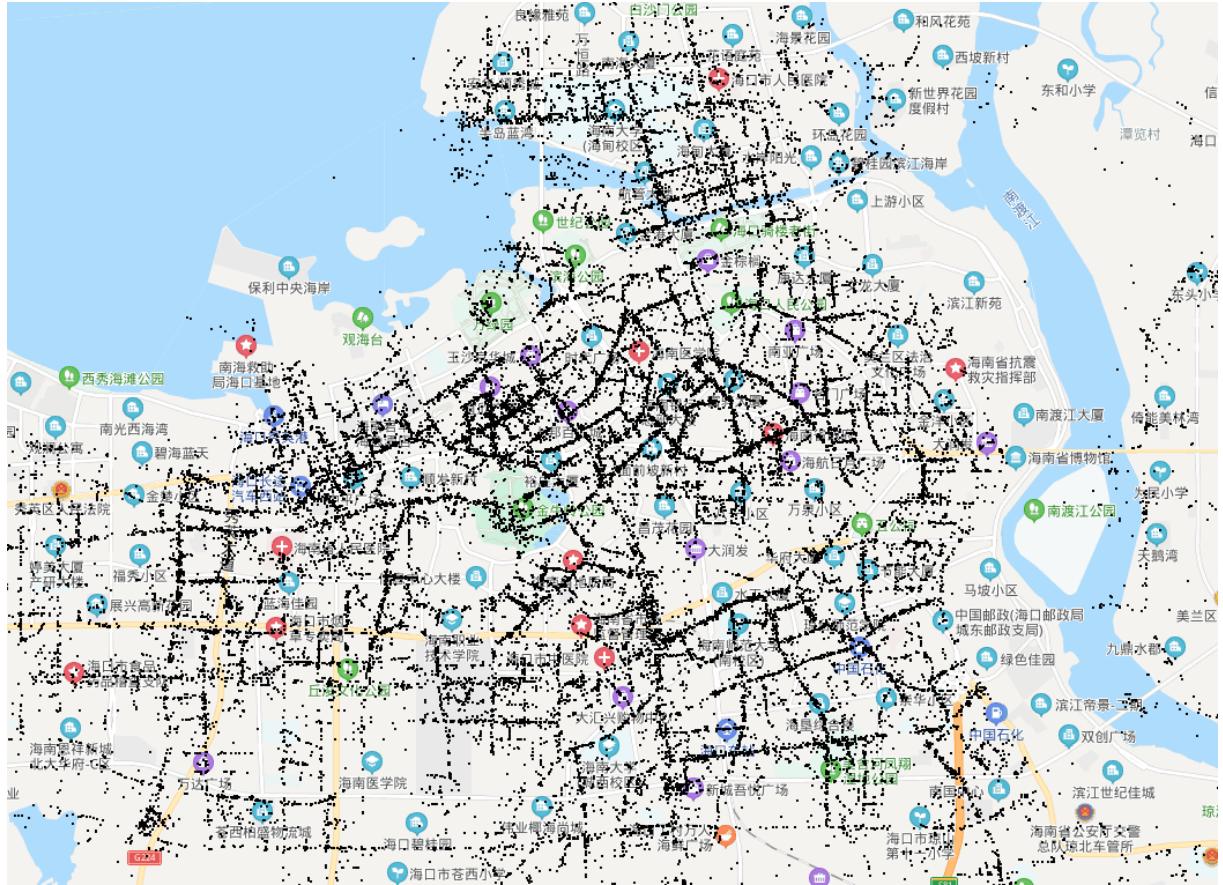


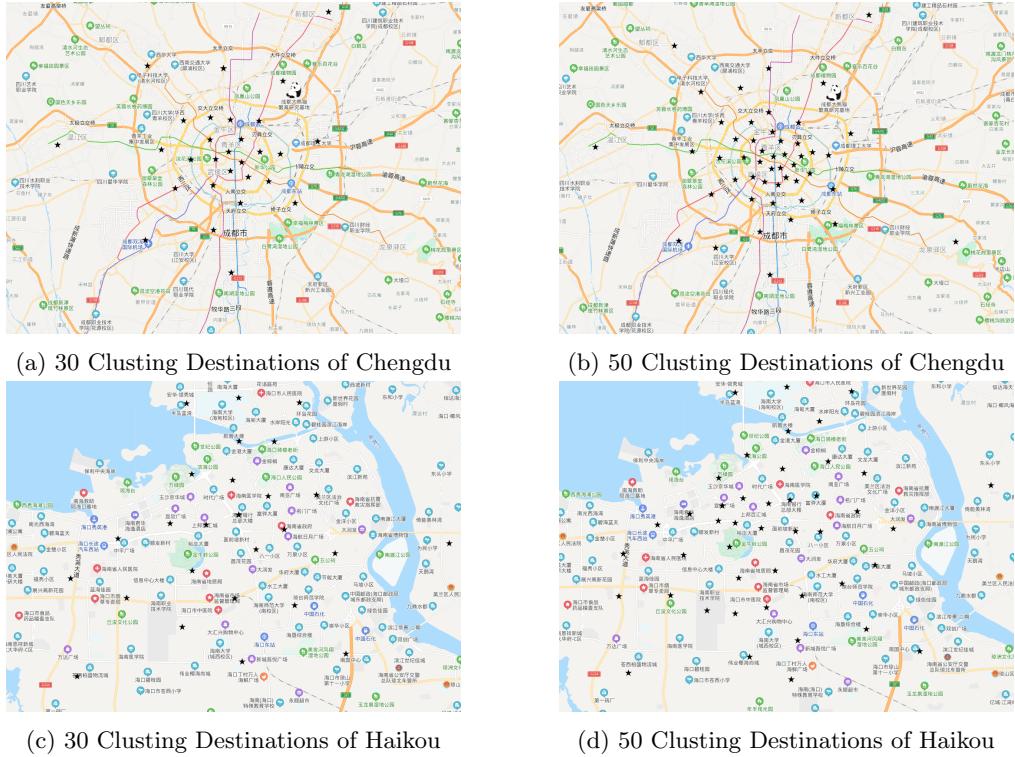
Fig. 2: Departure nodes map of Haikou

kilometer is  $c_b$ . The cost per route of a bus is  $c_r$ . The bus fare price for one passenger from the departure station to station  $d_j$  is  $p_j$ . Please dispatch each bus  $i$  with a route whose length is  $r_i$  kilometers to but do not need to satisfy all passengers (suppose satisfying  $x$  passengers) and only in order to maximum the profit  $P = \sum_{i=1}^x p_i - \sum_{i=1}^m c_r - \sum_{i=1}^m (c_b \times r_i)$ .

3. Suppose that there are  $n$  passengers and  $m$  buses at the departure station, while the destination station of passenger  $i$  is  $d_i \in D$  and the capacity limitation of each bus is  $L$ . The bus fuel cost per kilometer is  $c_b$ . The cost per route of a bus is  $c_r$ . The bus fare price for one passenger from the departure station to station  $d_j$  is  $p_j$ . Please dispatch each bus  $i$  with a route whose length is  $r_i$  kilometers to but do not need to satisfy all passengers (suppose drop off  $y$  passengers and it will lead to losing  $\alpha y$  more potential passengers) and in order to maximum the profit  $P = \sum_{i=1}^{n-y-\alpha y} p_i - \sum_{i=1}^m c_r - \sum_{i=1}^m (c_b \times r_i)$ .

#### 4.4 Algorithm Design

**Brief Explanation of Genetic Algorithm in Path Planning:** For ODRP-V2 problem, since in section 2.1 we've select the  $k$  hot destinations, we view them as points (dubbins) here. Our goal is to maximize the total profit of STMB, which is to say: minimize the total distance of routes (since the car maintanence fee per route is a constant  $c_r$ , the fule cost per car per kilometer is also a constant  $c_b$  and we don't need to concern about the customer's dissatisfaction factor here) while meet the restriction ( $m$  bus and each bus's limitation is  $L$ , so we can take at most  $m * L$  customer). In order to meet the goal, we need to let each bus takes as many customers as possible and try to dispatch routes for all the buses (since  $n > m * L$ ). It's obvious that each bus's route will not contaion the same vertex twice, after it has reached its last destination station or all the customers in the bus have been delivered properly, the shortest path is to directly go back to the departure station. These features are the same as Multiple Travelling Salesman's Problem (we use MTSP to represent it in the following part). Therefore, we consider to

Fig. 3:  $k - \text{means}$  Clustering Destinations Map

use Genetic Algorithm to solve route planning problem because it's a typical, widely used method to solve MTSP. There are five main metrics of GA: gene, chromosome, individual, population and fitness. Each of these five metrics is corresponding to separated stations, route, one bus dispatching, all  $m$  buses route dispatching, total geometry distance for each route. Since it's very convenient to use a sequence of numbers to represent a route, where each number represents a separated station, it's easy to construct chromosome, which is convenient for our later operation. And there are three main steps of GA:

- Step one: Calculate the fitness of each individual of current population and do the selection according to the selection rules (we use tournament selection here) to form a new generation .
- Step two: Perform crossover in some rate on each individuals. When two individuals are selected in crossover steps, exchange their chromosomes(exchange some stations in the route) using the cross strategy.
- Step three: Do the mutation in some rate on each individuals. When the individual is selected to do the mutation, remove some stations from the route and add new stations.

#### 4.5 Pricing Strategy

We tend to charge by mileage, that is, the distance from the Department to the passenger's destination is directly proportional to the custom fare. At the same time, we introduce the fare gap parameter, that is, the difference  $p$  between our pricing and taxi price, as the fare gap parameter. The smaller the fare gap, the lower the parameter, which means that the higher the fare, the smaller the probability of passengers willing to choose our bus system. Our goal is to find the extreme point in the middle.

At first, we need to show the data of simple bus and taxi price strategy in both Chengdu and Haikou to better determine our special bus price. Here is shown in Tab.3.

Our special bus price need to be one value between simple bus and taxi. So here we introduce a  $P$  function to express the price strategy. Fig.4 shows how the  $P$  function determines.

The key step is how to we determine  $A$  and  $B$ .

- **Node A.** Assume our bus system price (which will not earn or lose) will not bore customers at all. At this time,  $P = 1$ .

Table 3: Price strategy of taxi and simple bus system

| Category    | Chengdu        |                          |     | Haikou         |                 |     |
|-------------|----------------|--------------------------|-----|----------------|-----------------|-----|
|             | Taxi(starting) | Taxi(extra /km)          | Bus | Taxi(starting) | Taxi(extra /km) | Bus |
| Day(taxi)   | 8              | 1.9(2-10km) 2.9(10-60km) | —   | 8              | 2(2-10km)       | —   |
| Night(taxi) | 9              | 2.2(2-10km) 3.3(10-60km) | —   | —              | —               | —   |
| Day(bus)    | —              | —                        | 2   | —              | —               | 2   |
| Night(bus)  | —              | —                        | 3   | —              | —               | 2   |

- **Node B.** Assume our price goes highest at taxi price, and due to carpooling factors, most passengers are unwilling to choose our bus system. But there is still part of them will choose our bus system (assume the ratio is 10%) because of their waiting bus is ready at departure or they are a small group of more than four people (more than taxi load). Due to the S-shaped change of customer purchasing psychology index about commodity price, we set *sigmoid* function as our  $P$  – price function, which is  $a - \frac{1}{c(a+e^{-d(x-e)})}$ . After solving, we can get  $P(x) = 1 - \frac{1}{1.12(1+e^{-d(x-\frac{a+b}{2})})}$ . After the specific value is brought in, the mathematical expression is as follows:

$$P(x) = 1 - \frac{1}{1.12(1+e^{-3.53(x-1.38)})} \quad (0.068 < x < 3.3)$$

$x$  means our bus fare per kilometer, 0.068 is the lower limit to earn profit and 3.3 is the night taxi fare.  $P$  function is like the probability function which reflects the willingness of passengers to choose our bus system. We roughly divide several points to test the change of customer satisfaction. Details are shown in Tab.4.

Table 4: Satisfaction  $P(x)$  Change Table

| Satisfaction | fare( $x$ ) | 0.068 | 1    | 1.5  | 2     | 2.5   | 3    | 3.3   |
|--------------|-------------|-------|------|------|-------|-------|------|-------|
| $P(x)$       |             | 0.99  | 0.81 | 0.46 | 0.197 | 0.124 | 0.11 | 0.108 |

In reality, we can not only consider profits from business, but also customer satisfaction. In fact, when the customer satisfaction is lower than 0.8, it is very unfriendly to the reputation of our bus system. Therefore, we do not consider the case of less than 0.8, only looking for a huge profit solution in the satisfaction of more than 0.85. That is to say, we consider to find solution near  $x = 1$ , we then devide another several points to test the profit and finally the best price strategy is 0.9yuan/km. Then  $P(x) = 8 + (x - 2) \times 0.9 + 1.8$ , which 8 is the flag-down fare of first 2km.

#### 4.6 Route Planning and Solution

**Optimal Solution** For optimal solution, we use *CPlex* to help solve it. And it is much harder than we think. We change a lot of codes but the computing power is not so ideal that only can calculate up to 30 orders. Details are as Tab.5.

After testing, our optimal solution calculating power's upper limit is 30 destinations, 25 carry capacity, 170 orders, which will cost more than half an hour to get the solution.

**Feasible Solution** We implement the genetic algorithm designed above and implement it with code, and then substitute it into our order data. The iterative process in the algorithm is as Fig.5.

And the final bus route plan is as follows:

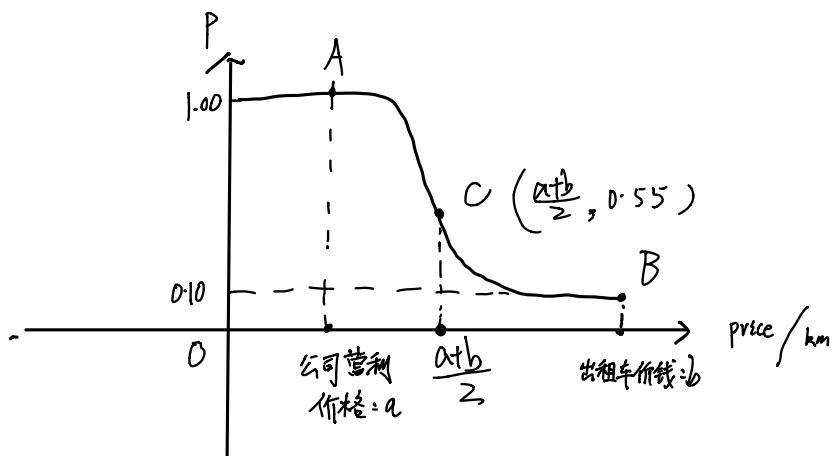
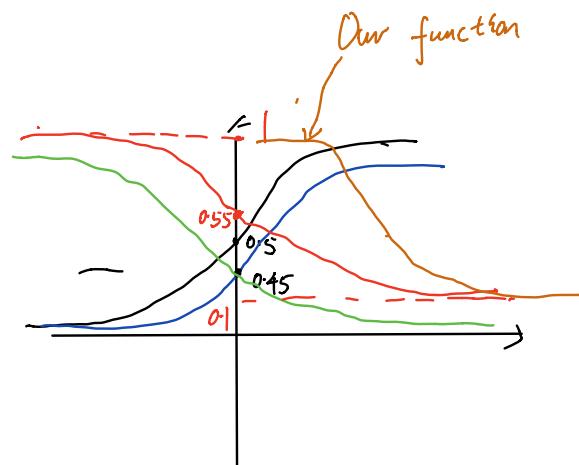


Fig. 4: P Function Formation

- **Chengdu-30.** When  $k = 30$ , there are 5 bus routes as Fig.6.
- **Chengdu-50.** When  $k = 50$ , there are 7 bus routes as Fig.7.
- **Haikou-30.** When  $k = 30$ , there are 3 bus routes as Fig.8.
- **Haikou-50.** When  $k = 50$ , there are 5 bus routes as Fig.9.

After route planning, we can get the final profit as follows:

And the visible graph it is as Fig.10.

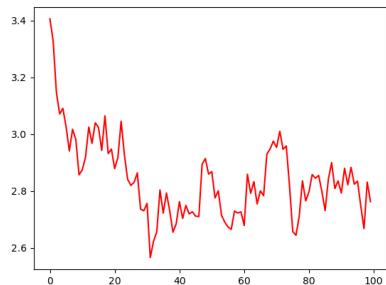
#### 4.7 Compare with Taxi

We can compare our bus with taxi as Tab.7.

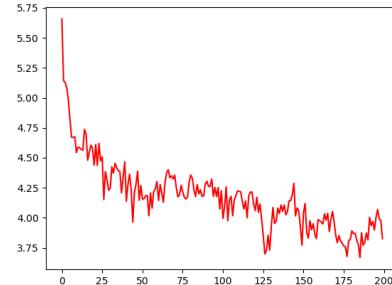
It is obvious that on price column, our bus system has significant advantages. And although for each route, we have higher cost than taxi, our one bus has much more carry capacity than one taxi does. What's more, we can attract much more orders. On the whole, our public transportation system creates more profits.

Table 5: Optimal Solution of 30 Orders

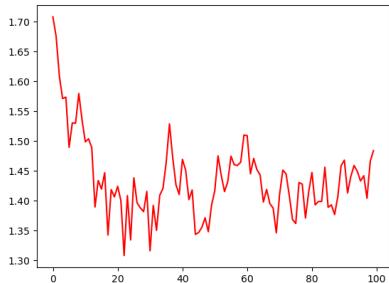
| Category               | Value      |
|------------------------|------------|
| Bus Number             | 2          |
| Orders Number          | 30         |
| $c_b$                  | 2yuan      |
| $c_r$                  | 5.2yuan    |
| $L$                    | 12         |
| Total Fuel Fee         | 4yuan      |
| Total Maintenance Cost | 19.217yuan |
| Total Income           | 212.48yuan |
| Profit                 | 189.2yuan  |



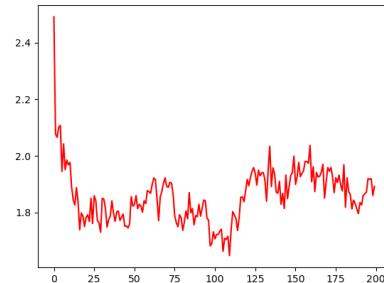
(a) 30 Destinations iterating of Chengdu



(b) 50 Destinations iterating of Chengdu



(c) 30 Destinations iterating of Haikou



(d) 50 Destinations iterating of Haikou

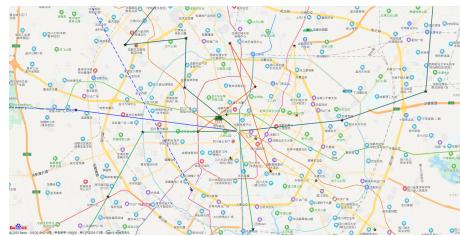
Fig. 5: Genetic Algorithm Iterative Graph

Table 6: Satisfaction  $P(x)$  Change Table

| City \ Bus Capacity | 25        | 30        | 35        | 45        |
|---------------------|-----------|-----------|-----------|-----------|
| Chengdu(30)         | 969927.94 | 978391.28 | 984389.47 | 992320.52 |
| Chengdu(50)         | 973702.95 | 982240.96 | 988093.55 | 996233.34 |
| Haikou(30)          | 714509.57 | 720148.41 | 724208.64 | 729559.07 |
| Haikou(50)          | 715535.04 | 721165.73 | 725214.58 | 730651.19 |



(a) Routes Map 1



(b) Routes Map 2



(c) Routes Map 3



(d) Routes Map 4

Fig. 6: 30 Destinations Routes Maps with Different Scaling of Chengdu



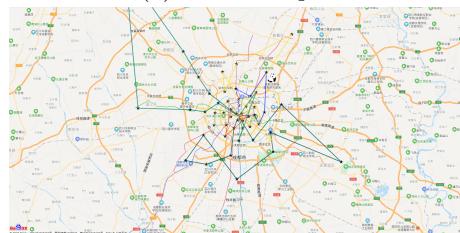
(a) Routes Map 1



(b) Routes Map 2



(c) Routes Map 3

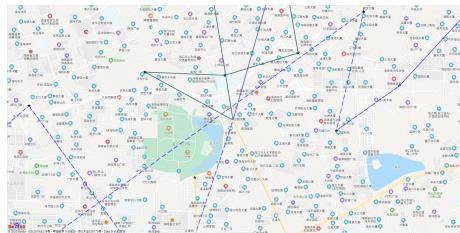


(d) Routes Map 4

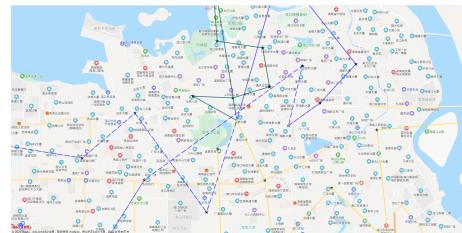
Fig. 7: 50 Destinations Routes Maps with Different Scaling of Chengdu

Table 7: Bus &amp; Taxi Comparing Table

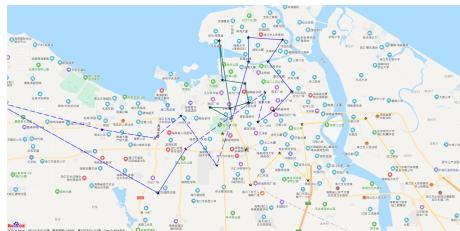
| Cost Column              | Car Column | Our Bus System                 | Taxi                   |
|--------------------------|------------|--------------------------------|------------------------|
| $c_f$ (fuel cost per km) |            | 2.211                          | 0.599                  |
| $c_r$                    |            | 5.2                            | 6.5                    |
| Price( $P(x)$ )          |            | $8 + 0.9 \times (x - 2) + 1.8$ | $8 + 3.3(x - 2) + 6.6$ |



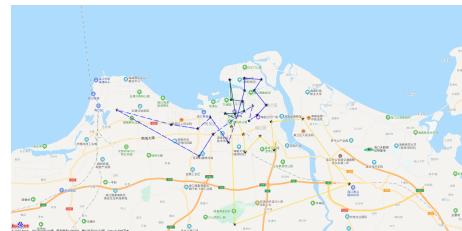
(a) Routes Map 1



(b) Routes Map 2

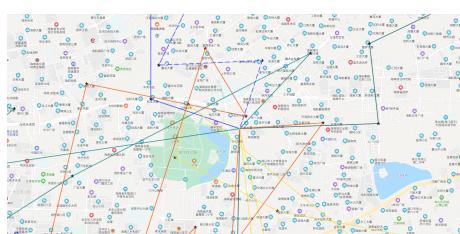


(c) Routes Map 3

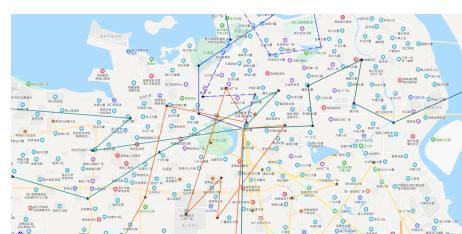


(d) Routes Map 4

Fig. 8: 30 Destinations Routes Maps with Different Scaling of Haikou



(a) Routes Map 1



(b) Routes Map 2



(c) Routes Map 3



(d) Routes Map 4

Fig. 9: 50 Destinations Routes Maps with Different Scaling of Haikou

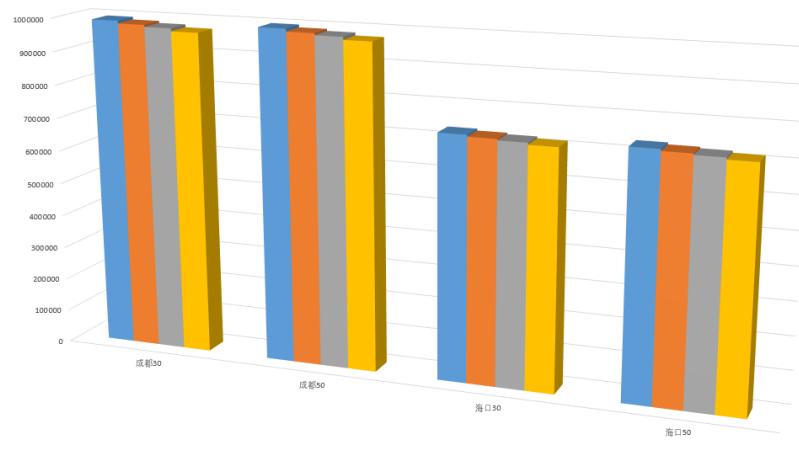


Fig. 10: Visible Profit

## Acknowledgements

- **Junxiang Cao:** In this project, we designed the model and algorithms to solve the bus booking problem and then analysed the performance of our algorithm. We made use of  $k - \text{means}$  algorithms, graph( $GA - TSP$ ) algorithms and other algorithms that we learned from the lecture. We met a lot of difficulties during these days, and we tried our best to get over them. For example, we used k-means algorithm to construct a destination stations graph to apply graph algorithms on it. After that, we designed a  $GA - TSP$  algorithm to search for the feasible solution of the selections of bus routes. And also, in this project, we have learned a lot from each other in our team when communicating and arguing on problems. In the end, I would like to thank Professor Gao for teaching us so many useful algorithms and analyzing skills. We think we will benefit a lot from them in our future study.
- **Sen Li:** Before doing this project, I thought it was a programming-oriented code to implement the project. But after the beginning, I gradually realized that the difficulty lies more in the design and ideas of the algorithm, and many details that need to be finalized. We first discussed the implementation idea of this project together, and then divided the work. Throughout the process of completion, although we have separate divisions of labor, we have always maintained a positive attitude to discuss and solve problems, which has also allowed us to move forward steadily. This project has greatly exercised our ability to cooperate in completing tasks and is a very good exercise.
- **Shiqu Wu:** We've learned a lot about how to design and implement a middle project, and most important thing I've learned is teamwork. My teammates are so great and so kind that they automatically take the responsibility of working out some difficult part of the project. They really let me understand the significance of trust in your teammates. When I run into trouble, rate of our project process is slowed. But my teammates always encourage me to keep my own pace and don't worry about the other parts. When using the *CPLEX* to work out the optimal results, I was stuck nearly for 3 days for one simple mistakes. And at first I don't know the computational ability of my computer and I push over half of the data into the *CPLEX* studio, my computer soon crashed and I've try a lot of times to figure out the reasons. It's really annoying that each time my computer crashed it take me a long time to reset all the other stuffs and open all the necessary materials. It's the patience and the trust of my teammates that push me through this difficulty. In the end, I want to thanks Pro. Gao for her kindness and patience about teaching us a lot of knowledge and a way to correctly treat the difficulty. Thank you.

## References

1. Shijuan Hu. Research on multi traveling agent problem based on Improved Genetic Algorithm. Jiangnan University, 2019.

2. [https://blog.csdn.net/weixin\\_43981664/article/details/104326573](https://blog.csdn.net/weixin_43981664/article/details/104326573).
3. <http://youjia.chemcp.com/sichuan/>
4. [https://mbd.baidu.com/newspage/data/landingshare?context=%7B%22nid%22%3A%22news\\_9583968445811533527%22%2C%22ssid%22%3A%220761f69f%22%7D&pageType=1](https://mbd.baidu.com/newspage/data/landingshare?context=%7B%22nid%22%3A%22news_9583968445811533527%22%2C%22ssid%22%3A%220761f69f%22%7D&pageType=1)
5. [http://www.xinhuanet.com/fortune/2019-03/01/c\\_1124178178.htm](http://www.xinhuanet.com/fortune/2019-03/01/c_1124178178.htm)
6. <https://zhidao.baidu.com/question/553058868370993452.html>
7. <https://www.carpenterbus.com/2015/01/annual-mini-bus-maintenance-costs/>