Design and Analysis of Online Ride-Hailing Platform

Ke Wang (019033910065, onecall@sjtu.edu.cn), Kaijian Li (119033910076, likaijian@sjtu.edu.cn), Shuyu Zhang (019033910066, carsonz@sjtu.edu.cn)

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

Abstract. Order dispatching and fleet management are two key challenge for Online Ride-Hailing Platform.In this paper,we have analyze the characteristics of Online Ride-Hailing and choose accumulated driver income (ADI) and order response rate (ORR) as our optimization indicator.First of all, we briefly described the solutions of the three tasks and answered the relevant questions of each task. Next, we define the problems we need to solve, point out the goals we need to optimize, and then analyze the methods we choose.Finally, we tested the model on a real dataset to evaluate the effectiveness of our approach.

Keywords: Order Dispatching, Fleet Management.

1 Introduction

In the ride-hailing platforms such as Didi chuxing and Uber, millions of orders and thousands of vehicles need to be matched with each other. there are two major decision-making tasks for such ride-hailing platforms, namely (i) order dispatching, i.e., to match the orders and vehicles in real time to directly deliver the service to the users[1], and (ii) fleet management, i.e.,to reposition the vehicles to certain areas in advance to prepare for the later order dispatching[2,3].

It appears that decisions to compare a pair of vehicles or to relocate a vehicle to an area require consideration of the future location of the vehicle and orders nearby. Thus, much of the work modeled the ordering and management of the fleet as a consistent decision-making problem and solved it with reinforcement learning[4]. Most of the previous work is either the ordering of the dispatch or the management of the fleet without the high correlation of these two tasks, especially for large-scale ride platforms in large cities, which leads to sub-optimal Performance. To achieve almost optimal performance, we model the entire highing platform in the form of dispatching (sending orders) and permutations (fleet management). We are similar to the vehicle and order as different molecules and aim to create stability of the system by reducing their number by sending and repositioning. To address this complex criterion, we provide two new views: (i) the interconnected ordering of dispatch and fleet management, and (ii) a joint review of intra-district (level grid) and inter-district (district level) distribution. With this practical motivation, we focus on modeling joint order dispatching and and fleet management with a multi-scale decision-making system[5].

The challenge of finding a trade-off between immediate and future rewards is seeking for an optimal strategy in terms of accumulated driver income (ADI).Greedily matching vehicles with long-distance orders can receive high immediate gain at a single order dispatching stage, but it would harm order response rate (ORR) and future revenue especially during rush hour because of its long drive time and unpopular destination.So, we have two goals to optimize.We need to consider ORR while considering ADI, and in the following discussion, we will use heuristic algorithms to analyze our optimization objectives.

You can find our work from
https://github.com/Brandonnogithub/order-dispatching-fleet-management

2 Tasks and Answer

2.1 Task 1

Task 1 You are required to study some situations of a ride-hailing problem. In general if you add too many constraints to a single problem it will become NP-hard. You can set many criterions on a ride-hailing algorithm, like the overall profit, the total waiting time, or the total travel distance to pick up passengers. In which situations a polynomial time algorithm is possible? How to make it an NP-hard problem?

ANSWER In this case, we first consider our criterions.We assume that the optimization objective is only Accumulated Driver Income (ADI). In this case, we only need to ensure that each region (giad) has enough small distance orders to ensure that drivers can continue to receive local short orders after completing local short orders. In the principle of maximum ADI, it is possible that we can get a polynomial time algorithm.And to make it an NP-hard problem,we must guarantee that we have known all orders and vehicles in advance. If we can know all orders and vehicles information in advance, we can predict the ADI, to move forward a single step to distribute the order.So we can add the constraints for our goal to make sure our problem is an NP-hard problem.

2.2 Task 2

Task 2 Now you are required to design the kernel of this ride-hailing platform. To simplify the problem,Suppose you have known all orders and vehicles in advance. Please design a (approximation) dispatching algorithm for it. To seek the optimal strategy, you may need to make some additional assumptions. As you address the problem requirements, it should be clear how you assumptions are related to the challenge. Also, be clear about what factors you are considering when addressing these questions.

ANSWER If we have known all orders and vehicles in advance,we can use Greedy Algorithm to achieve order dispatching as shown in Algorithm 1.Greedy is a better approximation if the distribution is stable.

Algorithm 1: Greedy Algorithm

```
1 Sort jobs by finish times so that $f_1 \le f_2 \le f_3 \le \cdots \le f_n$ ;
2 $A \leftarrow \varnothing$;
3 For j = 1 to n do;
4 if job j is compatible with A then
5     $A \leftarrow A \cup \{j\}$
6 end
7 return A;
```

2.3 Task 3

Task 3 You are now in the future when unmanned vehicles have finally become available. To take the full advantage of this exciting new technology you need a better algorithm for your platform.

1 If you dont make any assumptions on the future orders, the best strategy for your unmanned vehicles would simply be move randomly. Suppose you have a fixed number (the actual number is up to you) of unmanned vehicles. Based on the real data, design a system to process the historical data and correlate the order dispatching and your fleet on the second day. Show your simulation result and the performance of your system.

2 Another 10 years have passed and now humans are banned from driving. How many unmanned vehicles do you need to serve your customers? It would be better to draw a diagram to show the replacement ratio with the number of unmanned vehicles.

ANSWER In this task, we use heuristics method to achieve our goals. The specific methods will be explained in detail later in the article.

3 Problem Definition

We formulate the problem of controlling large-scale vehicles in online ride-hailing platforms, which combines order dispatching system with fleet management system with the goal of maximizing the city-level ADI and ORR. We use the rectangular-grid world to represent the map and take a grid as an regionand we set distance between grids based on the distance from the beginning of the order to the end.So we can make sure the vehicles located at the same grid share the same setting.

Definition We define every order $u \in U$ ang every driver $v \in V$.

Definition We define a tuple $G = (M, S, T, R)$, where M, S, T, R are the number of region, set of states, time step, reward function, respectively.

Region We consider each region cell as an grid identified by $i \in I$, where $I = \{i \mid i = 1, ..., M\}$. We define the grid function $Reg = G(m)$,$m \in i$.

State We set the state $S = (S_t, S_f)$,S_t is the start state and S_f is the finish state.

Time step We set time step $T = \{t \mid j = 1, ..., N\}$ and every time step is 600 seconds.

Reward We consider the reward function R determines the direction of optimization and is proportional to both immediate profit and potential value. More specifically, the reward is a linear function of distance from the pick-up place to get-off place.For the reward function,passengers start at 8 yuan for boarding and travel within two kilometers is 8 yuan.The unit price for each additional kilometer is 1.9 yuan after the trip exceeds two kilometers.So the reward function is a linear function of distance if we make a assumption that the vehicles travel at a constant speed.

Definition We define the order dispatching function $OD = F(i)$, where $i \in I$ is a set of grid, $F()$ is order dispatching function.

Definition We define the fleet management function $FM = H(i)$, where $i \in I$ is a set of grid, $H()$ is fleet management function.

Definition We define the left order dispatching function $od = f(i)$, where $i \in I$ is a set of grid, $f()$ is left order dispatching function.The left means there are left unmatched orders and vehicles after we use the function $Reg = G(m)$ search every giad.

Definition We define the left fleet management function $fm = h(i)$, where $i \in I$ is a set of grid, $h()$ is left fleet management function..The left means there are left unmatched orders and vehicles after we use the function $Reg = G(m)$ search every giad.

Definition We define the maxflow $mf = maxflow(od, fm)$, where maxflow is maximum flow, od is order dispatching function,fm is fleet management function.This is a goal that needs to be optimized.

4 Design and Analysis

We will design the heuristic algorithms to achieve our goals, and we choose the Greedy as a baseline to evaluate our results.

Model First, the city center is divided into 15*17 rectangular grids. In each time step, we traverse all the rectangular grid. There are two situations : If the number of vehicles in the area is greater than the number of orders, all orders in the area can be allocated in one time step;if the number of vehicles in this area is less than the order number($v \leq u$)), we adopt the greedy strategy to arrange the distance of all orders (u) from small to large, and take the first (v) orders to match the vehicles. Then, we can use the left order dispatching function and the left fleet management function to judge the remaining orders and vehicles.Finally,we use the maximum flow strategy to optimize our final objective function. The Algorithm of order dispatching and fleet management is shown in Algorithm 2.

When a passengers from a place S_t to another place S_f,the reward function R can be defined as:

$$R = 1.9* \mid S_f - S_t \mid +2 \tag{1}$$

Algorithm 2: Algorithm of order dispatching and fleet management

```
Data: The set of state $S$, the set of order $U$ and driver $V$
Result: ADI and ORR
1  $M \leftarrow G(m)$;
2  for $i = 1, 2, ..., M$ do
3      $OD \leftarrow F(i)$;
4      $FM \leftarrow H(i)$;
5      $od \leftarrow f(i)$;
6      $fm \leftarrow h(i)$;
7      if $od \geq 0 or fm \geq 0$ then
8          $mf \leftarrow maxflow(od, fm)$;
9      end
10     $updateG(m)$;
11 end
```

As mentioned above,we can design the Algorithm of Reward Function as our optimization objective.As illustrated in Figure 1, we resemble vehicle and order as different molecules and aim at building up the system stability via reducing their number by dispatching. In a region(orange rectangular-grid),a vehicle can confirm an order when the passenger position is within this area.In our experiment, the side length of the rectangular-grid is approximate 2 kilometers.

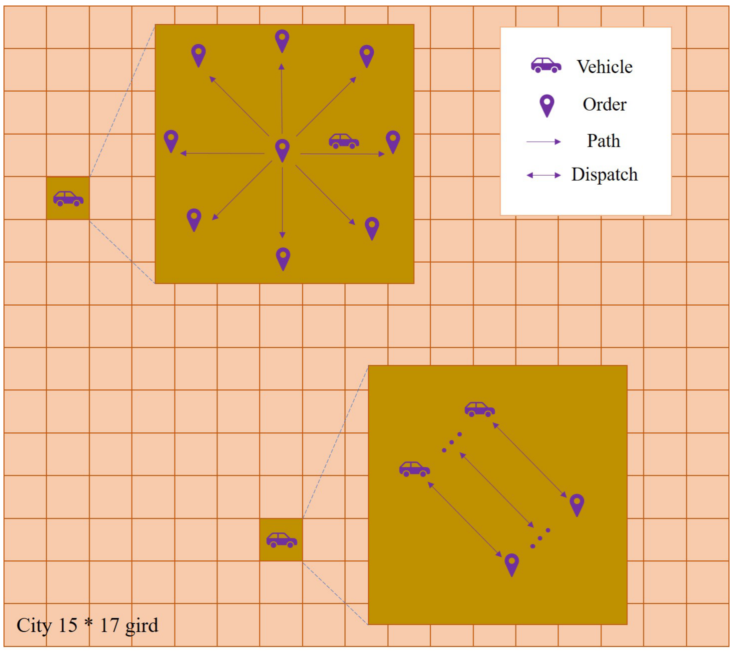


Fig. 1. Overview of Online Ride-hailing task

5 Experiment

We conduct extensive experiments to evaluate the effectiveness of our proposed method in joint order dispatching and fleet management. But there are no published methods fitting our task. Thus, we first compare our proposed method with Greedy models based on a single order dispatching environment. Then, we further evaluate our proposed method in joint setting and compare with its performance in our datasets.

Data Description The real world data provided by Didi Chuxing includes order information and trajectories of vehicles in the central area of cities with millions of orders in a month. Data of each day contains million-level orders and tens of thousands vehicles in each city. The order information includes order ID,start billing time, end billing time, origin(latitude and longitude), destination(latitude and longitude). The trajectories information contain the driver ID, order ID, positions of all vehicles. As the edge of grid is approximate 2 kilometers, the central area of the city is covered by a rectangular-grid world consisting of 15*17 grids in the city.

Experiment Set In the grid-based simulator, the city is covered by a rectangular grid world. At each timestep t, we observate all the vehicles and available orders including real orders. Specifically, we suppose the current timestep of simulator is t, we randomly sample real orders occuring in the same period. Our time step is 600 seconds, all order which are finished in less one time step are setted as one time step.

In our experiment there are about 4,000 drivers in one day, but they won't be online all day. So we add a bias(λ) to driver number and we choose λ as a hyper-parameter.

Because coordinate system from dataset is GCJ02, we change GCJ02 coordinate system to World Geodetic System1984 coordinate system(wgs84).At the same time, we remove sone of the data about wrong time and far position by data preprocessing.And we last contain 6934371 pieces of data.

Result Analysis We train the model, store the trained model periodically, and conduct the evaluation on the stored model.And we compare the performance of different models regarding two criteria, including ADI, computed as the total income in a day, and ORR, calculated by the number of orders taken divided by the number of orders generated.

As shown in Table 1, When we choose a random strategy, although some order may every far,them will still be dispatched,and the reward function(ADI) is high. When we choose a greedy strategy, all orders will be given priority to the nearest vehicle, which will result in an increase of order response rate(ORR), while the ADI will decrease because greedy algorithm increase the idle time for each vehicle. As shown in Table 1, for ADI ,the random strategy is higher than the greedy strategy, for ORR ,the random strategy is lower than the greedy strategy.

When we consider there are not unmanned vehicles ,we can choose a greedy and maximum flow strategy, the ADI is higher than the random strategy,the ORR is higher than the greedy strategy,it means greedy and maximum flow strategy is a good choice for our optimization problems.

Table 1. Performance comparison of competing methods in terms of ADI and ORR(λ =0.3).

Method	ADI(ten thousand RMB)	ORR
Random Strategy	306.92	0.7484
Greedy Strategy	294.85	0.8024
Greedy and Maxflow Strategy	**382.73**	**0.9692**

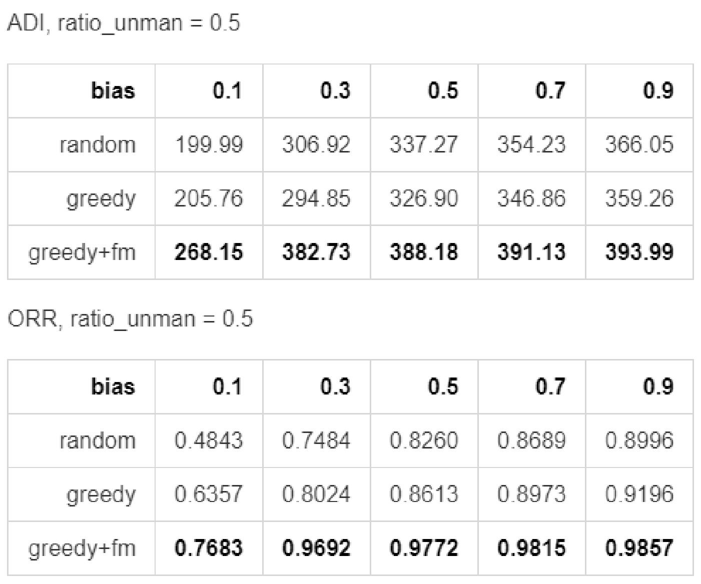
When we consider there are some unmanned vehicles,we set the proportion of unmanned vehicles in all vehicles as $ratio_unman$, the ADI is higher than the random strategy,the ORR is higher than the greedy strategy,it means greedy and maximum flow strategy is a good choice for our optimization problems as shown in Figure 2.

As shown in Figure 3 and Figure 4, we set λ =0.1,0.3,0.5,0.7,0.9. With the increase of λ, ADI and ORR also starts to get larger. When the parameters $\lambda \geq 0.3$, ADI and ORR increases slowly.

As shown in Figure 5 and Figure 6, The x-coordinate is the proportion of the total number of cars that are unmanned.We can conclude that if all of them were replaced by unmanned vehicles, it would take about 86 percent of the original cars to keep the performance of the system unchanged.

Acknowledgements

In the process of writing and realizing this paper, we would like to thank the instructor for her professional and careful guidance, we would like to thank the teaching assistants for her help,we would like to thank our classmates for their support.In this course, we systematically learned some basic knowledge of algorithms, and we were deeply impressed by Mr. Gao's rigorous and pragmatic teaching style.Thank you again!



ADI, ratio_unman = 0.5

bias	0.1	0.3	0.5	0.7	0.9
random	199.99	306.92	337.27	354.23	366.05
greedy	205.76	294.85	326.90	346.86	359.26
greedy+fm	**268.15**	**382.73**	**388.18**	**391.13**	**393.99**

ORR, ratio_unman = 0.5

bias	0.1	0.3	0.5	0.7	0.9
random	0.4843	0.7484	0.8260	0.8689	0.8996
greedy	0.6357	0.8024	0.8613	0.8973	0.9196
greedy+fm	**0.7683**	**0.9692**	**0.9772**	**0.9815**	**0.9857**

Fig. 2. Performance comparison of competing methods in terms of ADI and ORR($ratio_{unman}$ =0.5)

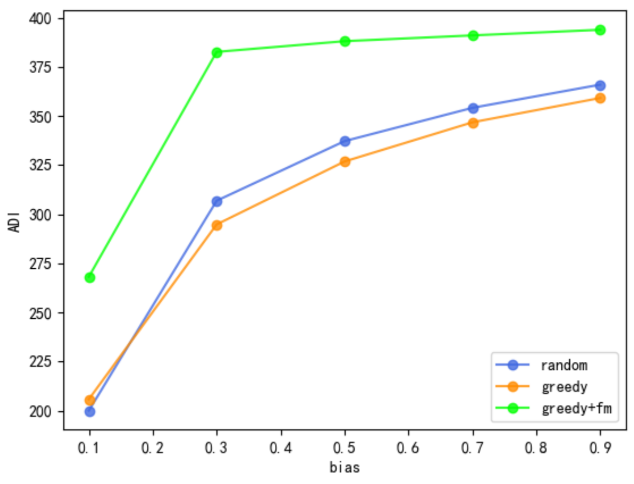


Fig. 3. Performance comparison of competing methods in terms of ADI

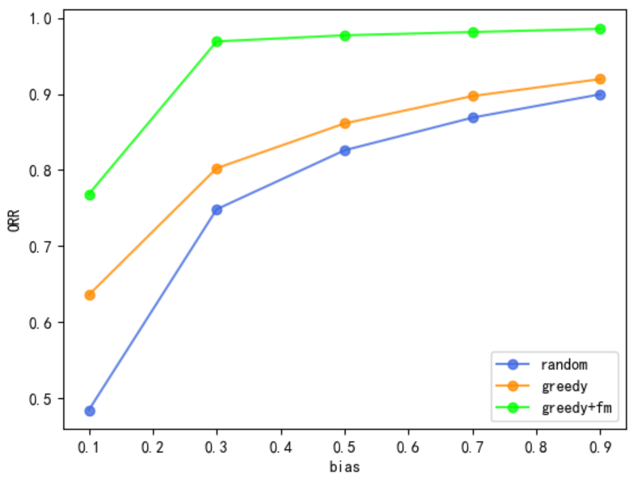


Fig. 4. Performance comparison of competing methods in terms of ORR

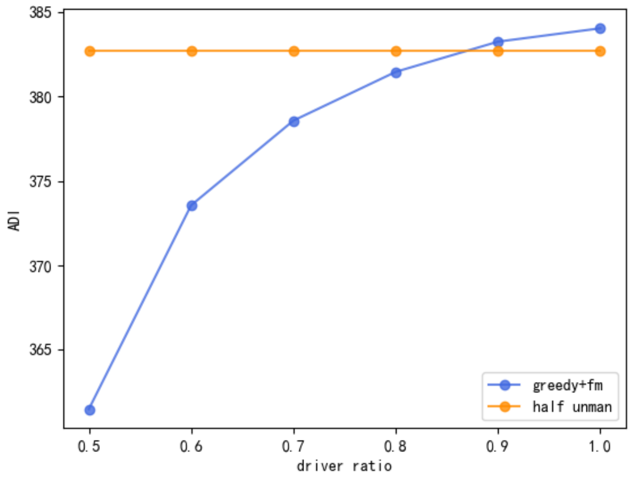


Fig. 5. Performance comparison of competing methods in terms of ADI

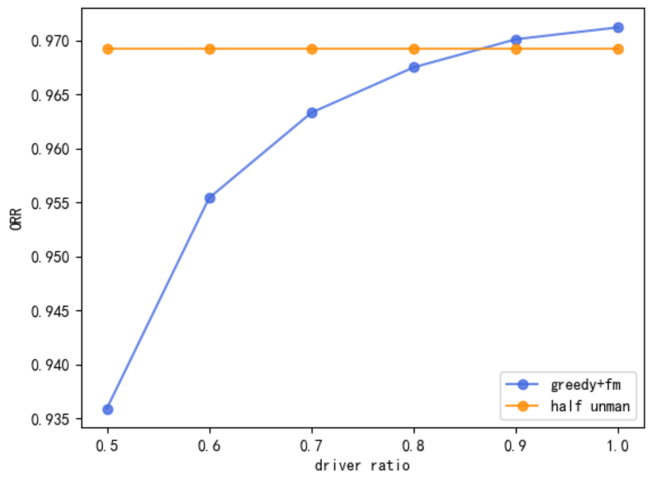


Fig. 6. Performance comparison of competing methods in terms of ORR

References

1. Qingnan Zou, Guangtao Xue, Yuan Luo, Jiadi Yu, and Hongzi Zhu. A novel taxi dispatch system for smart city. In International Conference on Distributed, Ambient, and Pervasive Interactions. Springer, 326C335(2013)
2. Hugo P Simao, Jeff Day, Abraham P George, Ted Gifford, John Nienow, and Warren B Powell. An approximate dynamic programming algorithm for large-scale fleet management: A case application. Transportation Science 43, 2, 178C197(2009)
3. Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. Efficient Large-Scale Fleet Management via Multi-Agent Deep Reinforcement Learning. arXiv preprint arXiv:1802.06444 (2018)
4. Xiaocheng Tang and Zhiwei Qin. A Deep Value-network Based Approach for Multi-Driver Order Dispatching. Technical Report (2018)
5. Jiarui Jin, Ming Zhou, Weinan Zhang, Minne Li, Zilong Guo, Zhiwei Qin, Yan Jiao,Xiaocheng Tang, Chenxi Wang, Jun Wang, Guobin Wu, Jieping Ye. CoRide: Joint Order Dispatching and Fleet Management for Multi-Scale Ride-Hailing Platforms. International Conference on Information and Knowledge Management (2019)