Seeking Based on Dynamic Prices: Higher Earnings and Better Strategies in Ride-on-Demand Services

Suiming Guo, Qianrong Shen, Zhiquan Liu, Chao Chen, Chaoxiong Chen,
Jingyuan Wang, *Member, IEEE*, Zhetao Li, *Member, IEEE*,
and Ke Xu, *Senior Member, IEEE*

***Abstract*— In recent years, ride-on-demand (RoD) services such as Uber and DiDi are becoming increasingly popular. Different from traditional taxi services, RoD services adopt dynamic pricing mechanisms to manipulate the supply and demand on the road, and such mechanisms improve service capacity and quality. Seeking route recommendation has been widely studied in taxi service. In RoD service, the dynamic price is a new and accurate indicator describing the supply and demand, but it is yet rarely studied in providing clues for drivers to seek for passengers. In this paper, we propose to incorporate the impacts of dynamic prices as a key factor in recommending seeking routes to drivers. We first justfiy why it is necessary to recommend seeking routes and consider dynamic prices, by analyzing real service data from a typical RoD service. We then design a reinforcement learning model based on order and GPS trajectories datasets, and take into account dynamic prices in the design. Results prove that our model improves both driver earnings and seeking strategies. On driver earnings, the reinforcement learning model increases revenue efficiency by up to 34.52%, and considering dynamic prices leads to another increase of 6.19%. On seeking strategies, drivers are encouraged to serve local demand first, and they are redistributed more evenly and effectively.**

***Index Terms*— Ride-on-demand, dynamic pricing, seeking strategy, reinforcement learning.**

I. Introduction

RIDE-ON-DEMAND (RoD) services such as Uber and DiDi are becoming increasingly popular, bringing convenience to both passengers and drivers. Compared with taxi, for passengers, RoD service is more convenient, affordable, and flexible; for drivers, they no longer need to obtain expensive licenses or medallions in most cities, and they are able to arrange working shifts more flexibly [1], [2].

Compared with traditional taxi service, RoD service is more intelligent from the following two perspectives:

A. Data-Driven

Thanks to the introduction of mobile apps for both passengers and drivers, the variety and volume of data generated within RoD service grow tremendously. For example, the service provider now has access to the spatio-temporal distribution of both drivers and potential passengers; by using the mobile app, passengers leave a trail of accurate order information, which was only possible to be inferred from GPS trajectories of taxi cabs previously; the mobile app and the on-board GPS device also enable drivers to exhibit their real-time locations and movements to passengers, giving more information to passengers and sometimes making them creating orders with less hesitation; lastly, the mobile app is able to record the behavior of passengers and drivers, which could then be used to study their behavior patterns.

Such datasets could help, in turn, to provide guidance or information to drivers, passengers and the service provider: drivers are able to learn the distribution of potential passengers and the estimated revenue of picking them up in different locations; passengers could view on the mobile app the distribution of drivers nearby, and the description of supply-and-demand condition; the service provider could design mechanisms to redistribute drivers to meet demands accordingly.

B. Dynamic Pricing

Dynamic pricing is widely used in most RoD services, as an effort to redistribute drivers to meet potential demand under different spatio-temporal circumstances. In most cases, the price of a trip is represented as the product of a *dynamic* price multiplier and a *fixed* fare calculated based on trip distance and duration. A higher price multiplier attracts more drivers to come and defers those passengers not in a hurry; and a lower price multiplier does just the opposite. In addition to manipulating drivers and passengers, the price multiplier also serves as an accurate and reliable indicator of the current supply and demand condition on the road.

Manuscript received 5 July 2022; revised 10 November 2022; accepted 2 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62002135, Grant 62172066, Grant 72222022, Grant 72171013, and Grant 62272195; and in part by the National Natural Science Foundation for Distinguished Young Scholars of China under Grant 61825204. The Associate Editor for this article was B. Singh. *(Corresponding author: Chao Chen.)*

Suiming Guo, Qianrong Shen, and Zhiquan Liu are with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China (e-mail: guosuiming@email.jnu.edu.cn).

Chao Chen and Chaoxiong Chen are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: cschaochen@cqu.edu.cn; cxchen@cqu.edu.cn).

Jingyuan Wang is with the Laboratory for Low-carbon Intelligent Governance, School of Economics and Management, Beihang University, Beijing 100191, China, also with the Pengcheng Laboratory, Shenzhen 518055, China, and also with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: jywang@buaa.edu.cn).

Zhetao Li is with the Guangdong Provincial Key Laboratory of Data Security and Privacy Protection, National and Local Joint Engineering Research Center of Network Security Detection and Protection Technology, College of Information Science and Technology, Jinan University, Guangzhou 510632, China, and also with the Guangzhou Information Technology Research Institute, Guangzhou 510075, China (e-mail: liztchina@hotmail.com).

Ke Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: xuke@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3243045

It is important for drivers to make decisions about how to seek for passengers, in either taxi or RoD service. Most drivers make decisions based on personal experience and some naive, ad-hoc strategies such as finding hot spots like CBD. Such strategies have two major drawbacks:

- These strategies are inaccurate, as different drivers may have varying experience and drivers always make decisions based on short-term predictions.
- They usually lead to city-level supply and demand imbalance – for example, it becomes more difficult to request for service in suburb locations that have high demand.

Previous work in taxi service tries to recommend seeking route to drivers by models and algorithms, such as finding local or global hot spots [3], modelling driver behavior using Markov decision process [4], [5], simulating driver behavior using a force-directed approach [6], [7], and etc.

In RoD service, dynamic pricing is designed to be an indicator of the supply and demand condition on the road, and should be used in providing clues to drivers. For instances:

- Dynamic prices describe a location's or region's attractiveness to a driver. For example, in taxi service, a region with a higher demand or a higher probability of picking up passengers is already good enough. But in RoD service, among two regions with the same pick up probability, the region with higher price multipliers appears to be more attractive.
- Dynamic prices are determined in real-time, and such characteristic also helps drivers to obtain the most up-to-date information about the supply and demand, and to avoid being guided by unreliable personal experience. For example, if a large number of drivers flock to a region with high demand, the price multipliers in the region would drop, preventing other drivers from going there.

In this paper, we focus on the seeking route recommendation problem in RoD service with dynamic prices. We first answer two questions, i.e., *why recommending seeking routes* and *why considering dynamic prices*, by analyzing real service data from a typical RoD service. We then establish a reinforcement learning model to tackle the problem at the level of city cell, recommending the next cell a driver should go to during seeking. Our study is based on the order and GPS trajectories datasets. In addition to the common knowledge about taxi GPS trajectories used in previous studies, we are now able to extract the trip fare and the associated dynamic price multiplier of each order. In our model, we incorporate the price multiplier into the design of reward, so that the decision-making of a driver depends not only on the probability of picking up passengers in a cell, but also on the potential profitability and the supply-demand condition in that cell. Simulation results show that the reinforcement learning model increases drivers' average revenue efficiency by up to 34.52%, and considering dynamic prices leads to a further increase of 6.19%. We also perform strategy evaluation, and it is shown that our model redistributes drivers more evenly and effectively, and encourages those in suburb to serve local demand first, avoiding supply-demand imbalance to some extent.

Our contributions are three-fold:

- Our study is one of the very few on seeking route recommendation in RoD service with dynamic pricing. Previous studies either are on taxi service, or do not take into account new features such as dynamic pricing. Our study, on the other hand, explores how dynamic prices help to recommend better seeking routes.
- We adopt a reinforcement learning model to tackle the problem. This helps to consider the long-term effects and redistribute drivers more effectively. Also, dynamic prices are incorporated into the model.
- We conduct extensive experiments based on real service datasets. Firstly, our datasets are from a real-world service provider, making our analyses and results more convincing and tenable. Secondly, our experiments exhibit the improvement of not only drivers' revenue, but drivers' seeking strategies as well.

The remainder of this paper is organized as follows. Section II reviews related work and section III provides a detailed discussion based on data analysis about why recommending seeking routes and considering dynamic prices. Section IV elaborates on the reinforcement learning model, which is then evaluated in section V. Section VI gives a brief summary and some discussions based on evaluation results. Finally, section VII concludes the paper.

II. Related Work

We provide discussions on related work about two topics: (a) seeking in taxi service, and (b) RoD service.

A. *Seeking in Taxi Service*

There are two steps, i.e., seeking strategies analysis and seeking route recommendation, in making drivers earn more, and both of them have been studied extensively in traditional taxi service. Seeking strategies analysis aims to identify, at the macro-level, the relationship between certain seeking strategies (e.g., going to local hot spots, driving faster, etc.) and driver revenue. By mining taxi GPS trajectories, as an example, [8], [9] identify the most profitable strategies under different circumstances. Some studies focus on targets other than driver revenue – e.g., driving style [10] or travel purpose [11]. In RoD service, [12] studies seeking strategies by mining the relationship between driver revenue and features extracted from multi-source urban data.

Seeking route recommendation is, by comparison, at the micro-level, and concentrates on recommending the next road segment or city cell a driver should go to, so that driver revenue is optimal. [13] performs recommendation by minimizing the distance between the taxi and anticipated customer requests; [4], [5], [14], [15] build a Markov decision process model; [16] also uses a Markov decision process model, but works for electric taxis as the model incorporates the charging process and battery constraint; [17], [18] applies reinforcement learning to solve the seeking route recommendation problem; [19] solves the problem from a different perspective – it improves the driver-passenger matching probability by enabling one driver to be matched to more than one passengers; [20] builds theoretical models of drivers, cities and the service to optimize earning in taxi and on-demand ride-hailing service.

B. RoD Service

Also known as on-demand ride-hailing, RoD service is a relatively new transportation service and thus has received limited attention. Most early studies try to compare RoD with taxi service based on the natural similarities in-between. For examples, [21] studies “waiting time” and claims that Uber reduces the waiting time significantly but may be more expensive in some cases. References [1], [22] focus on “market share” of taxi, Uber, and public transportation services. References [23] and [24] discusses the market effects of Uber’s entrance such as the changes to drivers’ behavior.

As the key feature that makes RoD service more intelligent, dynamic pricing has also been studied from different perspectives. For examples, [25], [26], [27] discuss how could dynamic pricing balance and redistribute the supply and demand, increase driver revenue and reduce passenger waiting time. [28] tries to place simulated users across the city and evaluates Uber’s surge pricing mechanism as a black-box. References [29], [30], and [31] explore real service data from a typical RoD service and analyze demand, the effect of dynamic pricing, passenger behavior and dynamic price prediction. Reference [32] studies the joint pricing and dispatching problem in on-demand ride-hailing, and proposes a distributed pricing framework. Some studies [25], [33], [34] focus on the effects of dynamic pricing on supply and demand from economics perspective.

Different from the above works, our study concentrates on the difference between RoD and taxi service. We analyze new patterns related to dynamic pricing in RoD service and justify that it is necessary to consider dynamic pricing in seeking route recommendation. In our model, we incorporate dynamic prices, and evaluate its effects on both driver revenue and seeking strategies. Besides, our study is based on city-scale real service data, making our results more tenable.

III. Data and Analysis

A. Datasets

Previous studies on seeking route recommendation in taxi service are usually based on GPS trajectories of taxis. In RoD service, on the other hand, more datasets are available due to the data-driven characteristic. Besides GPS trajectories, we also use order dataset with dynamic prices. All datasets are from Shenzhou UCar (https://bit.ly/2MG47xz), a major RoD service provider in China [7].

Within RoD service, the order request and creation is done through an mobile app. The passenger opens the app and fills in basic information such as the addresses of origin and destination. The app then sends back the information to the service provider and retrieves the estimated trip fare as well as current price multiplier. If the trip fare is satisfactory, the passenger could then press a button to request for service; otherwise, s/he could just give up the request and exit the app. In our data, the price multiplier is within the range [1.0, 1.6].

We obtain the following datasets:

GPS trajectories. This is similar to the common dataset used in taxi studies, and contains the GPS records of every single car in operation. Each record consists of the following fields: location (i.e., longitude and latitude), time-stamp, speed, direction, the unique ID of the car, etc. The time interval between two consecutive records is two minutes. This dataset spans from Nov. 2015 to Mar. 2016, and on each day there are, on average, about 3,500 to 3,800 cars on the road working for the service provider in the city of Beijing, China. In our data, the city of Beijing is defined with a range of longitude $[116.22, 116.56]$ and a range of latitude $[39.81, 40.07]$.

Order dataset with dynamic prices. In taxi service, order information is usually inferred from GPS trajectories – i.e., the flipping of status from “passenger on board” to “vacant” means the end of an order. In RoD service, due to the use of mobile app, it is now possible to record accurate order information. Each entry in order dataset corresponds to an order, describing origin and destination, boarding and arriving time, the unique ID of the passenger/driver/car/order, the type of order, etc. The time span is the same to that of GPS trajectories, and our dataset covers about 14 million orders.

Our order dataset also gives information about dynamic price multiplier, obtained through the *EstimateFee* event generated when the app sends back all the information filled in by the passenger. The service provider returns the estimated trip fare and the current price multiplier upon receiving the event. Each event contains information such as the event time, event location, estimated trip fare, price multiplier, the unique passenger ID, etc. We thus associate each order with the closest event from the same passenger ID, and use the price multiplier in the event as the price multiplier of the order.

In the above datasets, all unique IDs of drivers, passengers, and cars are anonymized so that one cannot relate an ID to a person or car in the real world.

B. Data Analysis

In data analysis, we mainly show the results that inspire us to answer *why to recommend seeking routes* and *why to consider dynamic prices*.

We first show the number of orders in different hours on a typical weekday in Fig. 1. An obvious observation is that the number of orders fluctuates dramatically throughout the day – it is very small during the night, and there are two peak periods. In the remainder of the paper, we regard $[7am, 9am]$ and $[4pm, 6pm]$ as the morning and evening peak hours, respectively.

We omit the number of orders on weekends due to the limited space. There are, additionally, two more reasons for this omission. Firstly, the number of orders is much smaller and more “randomly” spread across hours throughout the day, showing that the supply-demand imbalance is less severe. By “more random”, we mean that there is not obvious peak or off-peak hours such as morning or evening peak on weekends. On weekends, a common observation is that the number of

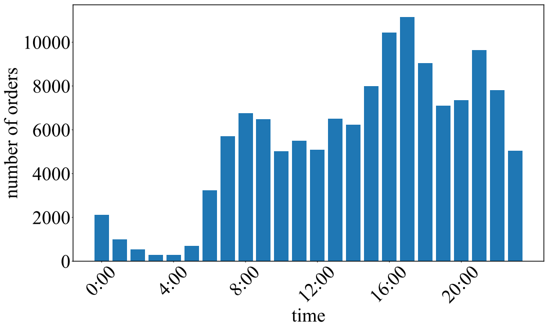


Fig. 1. The number of orders in different hours on a typical weekday.

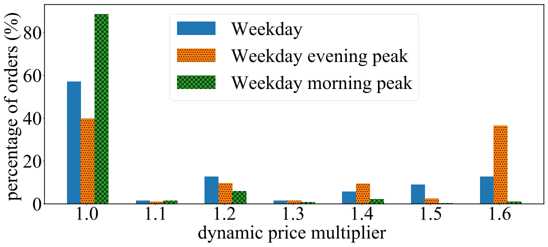


Fig. 2. The percentage of orders of different price multipliers.

orders is small during early morning, and then gradually rises to, and stays at, a higher level after that [12] and [31]. When the imbalance is less severe, the room for improvement by seeking route recommendation is also reduced. Secondly, due to the less severe supply-demand imbalance, the price multiplier is relatively stable and is usually close to 1.0 on weekends. This is natural, as there is no need to use dynamic pricing to manipulate drivers and passengers. As our focus is "seeking based on dynamic prices", it is thus clear that data analysis or further evaluation of our models on weekends is not necessary. In fact, without dynamic pricing, previous works such as [14] and [18] in taxi service are probably enough.

The above observations on a weekday are common in similar studies on taxi service, and in the following we turn to dynamic prices. Fig. 2 shows the percentage of orders of different price multipliers throughout the day, during morning and evening peak hours on weekdays. We have the following observations:

- The price multipliers vary among orders, no matter in peak hours or throughout the day. Hence, to improve driver revenue, it is necessary to give them guidance to obtain orders with higher prices.
- The fluctuations of price multipliers are different during different time periods. For example, during evening peak, the percentage of orders with price multiplier 1.6 is about 37%, very close to that of price multiplier 1.0. This shows that during evening peak high price multipliers are more common. Curiously, things are just the opposite during morning peak – even with a large number of orders, the price multipliers remain as low as 1.0 in most cases. One possible reason is that the origins of orders are scattered across city during morning peak, and hence higher prices are rare at the city level.

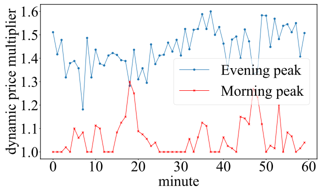


Fig. 3. The fluctuation of price multipliers during peak hours in business regions.

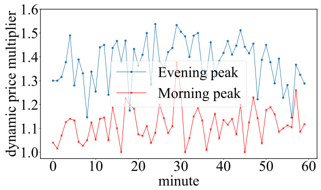


Fig. 4. The fluctuation of price multipliers during peak hours in residential regions.

It is thus not enough to study the distribution of price multipliers at the city level, and we need to explore the spatio-temporal characteristics of the distribution. We first choose some special regions and illustrate the fluctuation of price multipliers during morning and evening peak in these regions. Fig. 3 and Fig. 4 show the price multipliers in business and residential regions. For business regions, we choose some well-known central business districts (e.g., the Financial Street, the Zhongguancun Street, etc.); for residential regions, we choose some communities that are home to tens of thousands of people (e.g., Tiantongyuan Residence and Huilongguan Residence). In these figures, we show the average price multiplier among all orders starting at each minute, and if no orders are created during one minute, we regard the average price multiplier to be 1.0. We observe that:

- The characteristics of price multipliers are more obvious than in Fig. 2. For example, the price multipliers of residential regions during morning peak are much higher than average, and sometimes higher than that of business regions. During the evening peak, price multipliers of business regions are always higher than 1.4, sometimes climbing to as high as 1.6.
- The degree of fluctuation is also important. For example, price multipliers of business regions during morning peak do not fluctuate drastically – showing that in fact fewer orders appear and hence the average price multiplier is always counted as 1.0.
- Observations from Fig. 2 to Fig. 4 show that price multiplier is relevant to both spatial and temporal features. Price multipliers may be different in different locations during the same period; even in the same location and during the same period, price multiplier may also fluctuate due to the changes of supply and demand. Hence, besides spatio-temporal features, price multipliers also need to be considered in recommending seeking routes.

We then further explore the spatial distribution of price multipliers during morning and evening peak, across the whole city of Beijing, as shown in Fig. 5 and Fig. 6. For these figures,

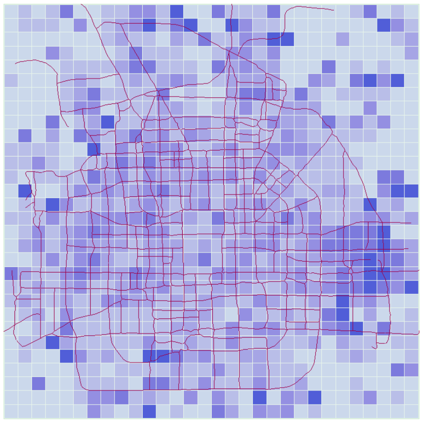


Fig. 5. The spatial distribution of price multipliers during morning peak.

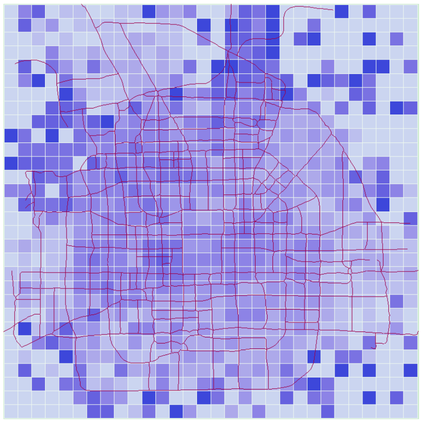


Fig. 6. The spatial distribution of price multipliers during evening peak.

we divide the map of Beijing into $30 * 30$ rectangular cells. The color of a cell illustrates the average price multiplier of all orders starting at this cell: the darker the blue color, the higher the average dynamic price multiplier. It is shown that:

- The spatial distribution of price multipliers is highly uneven across the whole city.
- For morning peak, the distribution is more even, and darker cells are mainly the sub-urban residential regions. This is in line with our previous observations, and the even distribution is the result of the fact that people go from scattered locations to the center of the city.
- For evening peak, it is clear that the distribution is much more uneven. We could roughly divide the color of cells into three categories: the lightest cells are mostly under-populated locations such as parks or sub-urban highways; the medium dark cells are around the city center, and most of them are the typical crowded business regions; the darkest cells appear around the suburb of the city.

The above observation is somehow counter-intuitive, as one may expect that price multipliers should be the highest around business regions in the city center. The reason why highest price multipliers appear in city suburb, as we consider, is supply-demand imbalance. During evening peak, drivers may flock to city center to seek for passengers based on, say, their experience, resulting in a low supply in the suburb. The lowered supply may not be enough to meet the demand, even though the demand is relatively low compared to that of city center, leading to the highest price multipliers in the suburb. To verify that, we show in Fig. 7 the spatial distribution of the number of orders during evening peak, and it could be seen that most orders indeed are concentrated in the city center, and there are relatively fewer orders in the suburb. We then choose two representative regions of equal size – one from Fig. 6 that covers the cells in suburb with higher price multipliers, and another from Fig. 7 that covers the cells in city center with higher demand. The number of idle drivers entering each region during an half-hour interval in the evening peak is counted, and it is shown that the number in city center is more than six times the number in suburb.

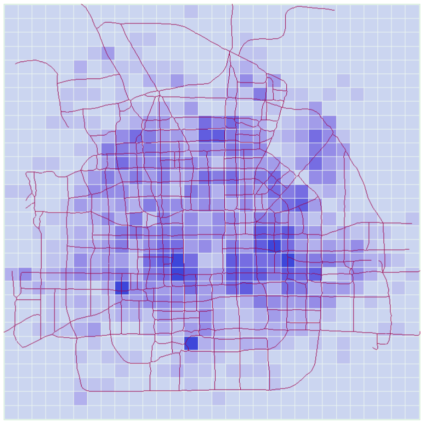


Fig. 7. The spatial distribution of the number of orders during evening peak.

We learn from this observation that more factors should be considered in recommending seeking routes. Drivers' experience or previous studies that are based on ad-hoc strategies such as finding hot-spots may encourage drivers to flock to city center, leaving the high demand around city suburb unsatisfied. If it is possible to guide those drivers who are already in the suburb to serve local orders first when the price multiplier is high – i.e., when the supply-demand imbalance occurs – then both driver revenue and passenger experience are improved.

We then pay attention to another question: as price multiplier is the real-time reflection of supply and demand, and may fluctuate up and down, is it still a good indicator to guide drivers? To answer it, we first divide all city cells into three categories: we calculate the average price multiplier of all orders starting at each cell during evening peak, and a cell is high (multiplier) cell if the average is in the range [1.5, 1.6]. Similarly, a cell is middle and low cell if the average is in the range [1.3, 1.4] and [1.0, 1.2], respectively. All high, middle or low cells form the high, middle or low area.

Fig. 8 shows the percentage of orders with different price multipliers in the high, middle, low area and the whole city during evening peak. We observe that:

- Regarding the whole city, the observation is the same as in Fig. 2. Orders with multiplier 1.0 account for about 38% – i.e., 62% of orders have higher price multipliers

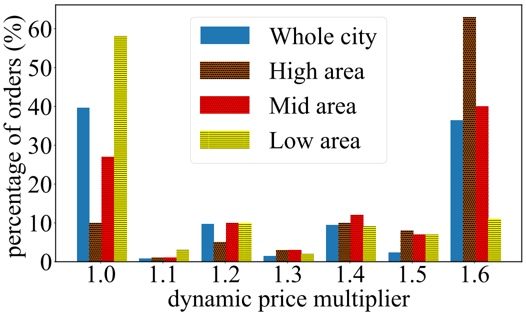


Fig. 8. The percentage of orders in the high, middle, low area.

– it is thus necessary to assist drivers to understand how to get high price orders.

- In high area, the probability of getting a higher price multiplier order is higher; and in low area, the probability of getting a lower price multiplier order is higher. For example, in high area, over 63% orders have a multiplier 1.6, and only 10% orders have a multiplier 1.0.

In other words, it is true that seeking in high price multiplier area does not necessarily mean getting a high multiplier order, but price multiplier is still indicative in a probabilistic way, and should be considered in seeking route recommendation.

Finally, we summarize our key findings from data analysis:

- The number of orders is different in different hours.
- The price multiplier is related to both spatial and temporal factors.
- A significant proportion of orders have price multipliers higher than 1.0, meaning it is necessary to guide drivers to obtain higher price multipliers.
- The supply-demand imbalance in city suburb tells us that drivers should not simply be encouraged to find hot-spots – they should serve local high price multiplier order first to improve driver revenue and passenger experience.
- The price multiplier is a good indicator and should be considered in seeking route recommendation.

IV. The Reinforcement Learning Model

A. Problem Formulation

Seeking route recommendation could be performed on the cell level, i.e., recommending the next cell a seeking driver should go after the current cell, or on the road segment level, i.e., recommending the next road segment when a seeking driver arrives at an intersection. Studying on the cell level means coarser spatial granularity and the inability to give segment-by-segment navigation, but has the advantage of being simple and efficient, leaving enough freedom to drivers, and still giving constructive recommendation results.

In this paper, we choose to study the problem on the cell level. Similar to section III-B, we first divide the city of Beijing, confined in the rectangular of $[116.22, 116.56]$ in longitude and $[39.81, 40.07]$ in latitude, into 900 $(= 30 * 30)$ rectangular cells of equal size. We also keep a timer and set it as $t = 0$ in the very beginning, and perform route recommendation whenever a car is vacant and $t < 60$ (in minutes) – i.e., recommendation is carried out throughout the whole hour. Our goal is to solve the following problem:

Definition 1 (Seeking Route Recommendation): Given the cell division of Beijing, the GPS trajectories and order datasets, and a subset of RoD cars X, try to find the optimal seeking route for each car in X to increase earnings. Specifically, for a vacant car, when it arrives at a cell, recommend the action for the driver, until s/he picks up a passenger. By "action", we mean going to a neighboring cell or seeking in the current cell. Recommendation is terminated when t reaches 60.

To solve this problem, we first use Markov decision process (MDP) to model the environment (e.g., the composition of driver revenue when delivering and seeking for passengers, the relationship between cells, etc.), and then use reinforcement learning to solve the problem based on such environment.

It should be noted that, similar to most previous studies on either taxi or RoD service, we have a basic assumption – *in modelling and solving the problem, only a relatively small proportion of drivers are assumed to adopt the recommended routes, and their behavior should not have a visible impact on the service*. Examples of "having an impact" include altering the spatio-temporal distribution of either drivers or potential passengers, changing the distribution of dynamic price multipliers, and etc. In other words, as the number of drivers adopting recommended routes is relatively small, it is not necessary to consider problems such as "whether we should re-calculate or re-predict the change of dynamic prices?" or "do passengers slightly change their demand pattern due to drivers' new seeking strategies?". Without such assumption, then we may need to consider drivers' adoption rate of recommendation, the prediction of dynamic price and its distribution, etc., and these should be left for future work.

B. Modelling the Environment

We use MDP to model the environment. In MDP, there is an agent with a starting state. In each state, the agent chooses an action and jumps to another state. The environment then gives a pre-specified reward to the agent, based on the action and state transition. The goal is to maximize the expectation of the sum of rewards or some other functions of rewards. The Markov property specifies that the probability of state transition, together with the reward generated, depends only on the current state and is not related to any previous states.

For our problem, "agent" is the driver, and "state" represents the time and location. "Action" means "the driver chooses a cell to seek for passenger", and "reward" covers the driving cost and the revenue made during passenger delivery.

MDP is suitable to model the decision-making during seeking, and has been used in a number of similar studies on taxi service. By comparison, our work is one of the very few, if there is any, that introduce dynamic prices into MDP and the solution of the problem.

All the notations used in modelling are listed in Tab.I.

1) States and Actions: A state s is described by three variables, such that $s = (l, t, d)$. l represents the driver's current location by cell ID. As we divide the city into 900 cells,

TABLE I

NOTATIONS USED IN MODELLING

Variable	Explanation
l	the cell ID that represents driver's location
t	current time kept by the timer
d	the incoming direction to a cell
D	the set of all possible incoming directions
s	a state such that $s = (l, t, d)$
S	the state space: the set of all states
a	an action
A	the action space: the set of all actions
$t_{seek}(j)$	the amount of time to seek for passengers in cell j
$d_{seek}(j)$	the driving distance to seek for passengers in cell j
$t_{drive}(j, k)$	the amount of time to drive from cell j to k
$d_{drive}(j, k)$	the driving distance from cell j to k
$P_{pickup}(j)$	the probability of picking up a passenger in cell j
$P_{dest}(j, k)$	the probability of a passenger picked up in cell j having a destination in cell k
$p(j)$	the dynamic price multiplier in cell j
$f(j, k)$	the trip fare from cell j to k
f_0	the flag-fall price
f_d	the unit price per kilometre
β	the driving cost per kilometre
α	the learning rate in Q-learning, $0 < \alpha < 1$
γ	the discount factor, $0 < \gamma < 1$
ϵ	the probability of exploration in Q-learning

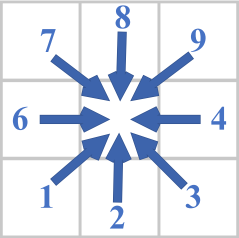


Fig. 9. Incoming directions: arriving at the center cell.

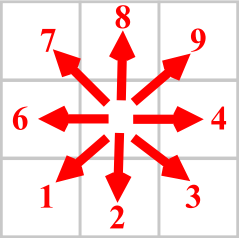


Fig. 10. Actions: outgoing directions from the center cell.

we have $l \in L = \{1, 2, \ldots, 900\}$. t is the current time, and $0 \leq t \leq 60$ as we specify in problem formulation.

d is the incoming direction – from which direction the driver arrives at the current cell during seeking. There are in total 10 possible directions, such that $d \in D = \{\emptyset, \nearrow, \uparrow, \nwarrow, \leftarrow, \circlearrowright, \rightarrow, \searrow, \downarrow, \swarrow\}$, as shown in Fig. 9. Among these directions, $\emptyset$ means "the driver has just dropped off a passenger and there is not a specific incoming direction", and $\circlearrowright$ means "keep seeking in current cell"; and they are not shown on Fig. 9. For simplicity, we denote these directions as 0 to 9.

An action a is actually the outgoing direction – the driver chooses an action and jumps to a neighboring cell. We denote actions in just the opposite directions to d, such that $a \in A = \{\swarrow, \downarrow, \searrow, \rightarrow, \circlearrowright, \leftarrow, \nwarrow, \uparrow, \nearrow\}$, as shown in Fig. 10. In Fig. 10, we also use 1 to 9 to index these actions, among which $\circlearrowright$ means "stay in the current cell" and is not shown on the figure. Under such definitions of d and a, it is clear that if a driver takes an action a and jumps to the next cell with an incoming direction d, then we have $d = 10 - a$.

Taking the incoming direction and action into consideration together makes it possible to avoid going into a loop. This has been shown in [4], and we don't provide discussions here as we adopt similar settings of d and a.

2) State Transitions: There are two kinds of state transition after taking an action and going to a neighboring cell:

- successfully picks up a passenger and delivers the passenger to destination;
- goes on seeking without picking up a passenger.

We assume that the current state is $s_0 = (i, t_i, d_i)$. In the first kind of state transition, the driver takes an action and goes to the neighboring cell j with a time and distance cost of $t_{drive}(i, j)$ and $d_{drive}(i, j)$, seeks for passenger for $t_{seek}(j)$ minute with a driving distance $d_{seek}(j)$, and picks up a passenger with probability $P_{pickup}(j)$. The passenger chooses cell k as the destination, with probability $P_{dest}(j, k)$. The driver delivers the passenger to cell k and drops him/her off, using $t_{drive}(j, k)$ minute with a distance $d_{drive}(j, k)$. As the driver drops off the passenger at cell k, the incoming direction is set to be 0. Hence, the driver is seeking at a new state $s_1 = (k, t_i + t_{drive}(i, j) + t_{seek}(j) + t_{drive}(j, k), 0)$.

In the second kind of state transition, the driver takes an action a and goes to the neighboring cell j, with the same time and distance cost of $t_{drive}(i, j)$ and $d_{drive}(i, j)$. But after seeking for passenger in cell j for $t_{seek}(j)$ minute with a driving distance $d_{seek}(j)$, the driver fails to find a passenger (with probability $1 - P_{pickup}(j)$), and ends up in cell j with incoming direction $d = 10 - a$. The driver goes on seeking at a new state $s_2 = (j, t_i + t_{drive}(i, j) + t_{seek}(j), 10 - a)$.

3) Rewards: We discuss the rewards between states based on the two different kinds of state transition. The rewards here represent the impacts from dynamic prices.

In the first kind of state transition, as the driver successfully picks up a passenger, the reward consists of two parts, the positive trip fare and the negative driving cost. The positive trip fare, $f(j, k)$, depends on trip distance and the price multiplier:

$$f(j, k) = p(j) * (f_0 + f_d \cdot d_{drive}(j, k)). \quad (1)$$

In (1), the dynamic price multiplier of the trip's origin (i.e., cell j) is denoted as $p(j)$. We regard all trips originating from cell j have the same price multiplier $p(j)$ during one hour – this is indeed an approximation, but it leads to no loss of generality and is the result of the difficulty of collecting real-time price multipliers. Also, f_0 is the flag-fall price, and f_d is the unit price per kilometre. The driving cost is an unified representation of fuel cost, car rental cost, etc., and is proportional to the driving distance. If we use r_1 to represent the reward in the first kind of state transition, then,

$$r_1 = f(j, k) - \beta(d_{drive}(i, j) + d_{seek}(j) + d_{drive}(j, k)). \quad (2)$$

In the second kind of state transition, the reward has a simple form as the driver does not pick up a passenger and

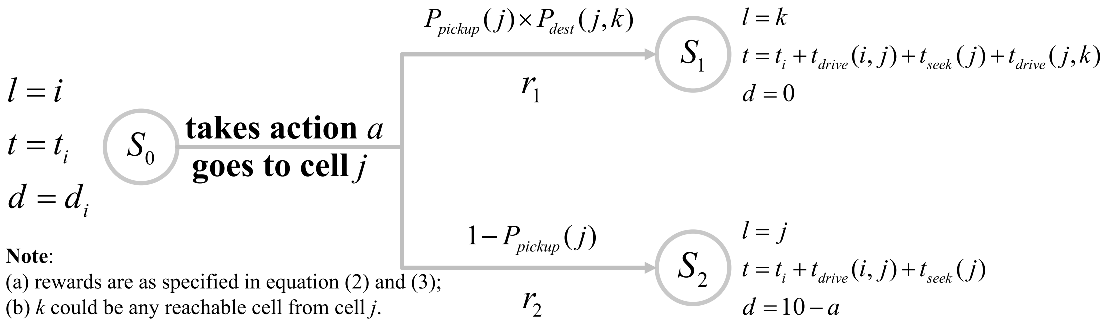

$l = i$
$t = t_i$
$d = d_i$
S_0
takes action a goes to cell j
$P_{pickup}(j) \times P_{dest}(j,k)$
r_1
S_1
$l = k$
$t = t_i + t_{drive}(i,j) + t_{seek}(j) + t_{drive}(j,k)$
$d = 0$
$1 - P_{pickup}(j)$
r_2
S_2
$l = j$
$t = t_i + t_{drive}(i,j) + t_{seek}(j)$
$d = 10 - a$
Note:
(a) rewards are as specified in equation (2) and (3);
(b) k could be any reachable cell from cell j.

Fig. 11. State transitions in the MDP environment.

does not have the trip fare term. The reward r_2 is,

$$r_2 = -\beta(d_{drive}(i, j) + d_{seek}(j)). \qquad (3)$$

Rewards and state transitions are illustrated in Fig. 11.

C. Solving With Q-Learning

There are two possible ways to find the driver's optimal action or policy in the above MDP environment. The first way works like dynamic programming, and tries to solve driver's optimal policy, i.e., a series of state transitions that lead to the highest reward. This is easy to understand, but has higher time complexity and requires agent's full knowledge of the environment. The second way belongs to reinforcement learning, and aims to find out the optimal action corresponding to each state that maximizes reward. By comparison, such methodology is faster as it does not go to great lengths to calculate rewards in all possible policies, and it does not require driver's full knowledge of the environment, as such knowledge could be obtained gradually through "learning".

Q-learning is a typical reinforcement learning algorithm used to solve for the state-action pair – the optimal action of each state. It is based on the idea of trial-and-error. The driver keeps a Q-table, storing a Q-value for each state-action pair that describes the utility of taking the corresponding action. At every state, the driver takes the action with the highest Q-value and jumps to another state, gets a reward from the environment (the form of reward could be unknown to the driver), and updates its Q-table. This is repeated until $t = 60$.

To ensure that the driver could discover better actions instead of being stuck in sub-optimal ones, the idea of exploration and exploitation is introduced. When the driver looks up the Q-table, in most cases s/he chooses the action with the highest Q-value (i.e., exploitation); but with a small probability ϵ s/he should choose a random action (i.e., exploration).

Algorithm 1 Q-Learning Algorithm

```
Input: the environment modelled by MDP.
Output: the Q-table $Q(s, a)$ for any state-action pair.
 1: $Q(s, a) = 0$ for any $s$ and $a$. //Initialize Q-table.
 2: while Q-table not converged do
 3:     $s = (l_{init}, t = 0, d = 0)$. //Initialize state.
 4:     Assign price multiplier $p(j)$ for each cell $j$.
 5:     while $t < 60$ do
 6:         Generate a random number $0 \le m \le 1$.
 7:         if $m \le \epsilon$ then
 8:             Choose a random action $a$.
 9:         else
10:             Choose the action $a$ with the highest $Q(s, a)$.
11:         Take action $a$, obtain reward $r$, jump to state $s'$.
12:         Update $Q(s, a)$ according to equation (4).
13:         $s \leftarrow s'$
14:         $t \leftarrow t'$
15: return Q-table
```

We show the Q-learning algorithm for a single driver in Algorithm 1. The algorithm has the environment modelled by MDP in section IV-B as input, and output the Q-table for the driver. There are two layers of loop. The inner loop is the trial-and-error of the driver throughout an hour. As the driver has no initial knowledge about the environment, it is necessary to perform the trial-and-error process for many times, as represented by the outer loop, until the Q-table converges – i.e., the differences between two consecutive updates of Q-table are smaller than a small threshold. We have the following explanations regarding some important lines:

- **Q-table initialization** (line 1): we set it with all zeros.
- **State initialization** (line 3): at the beginning of every iteration in the outer loop, we initialize the driver state to $s = (l_{init}, 0, 0)$. l_{init} is the cell that the driver starts seeking at this hour from our datasets.
- **Dynamic price multiplier assignment** (line 4): we calculate, from our datasets, the percentages of orders with different price multipliers in each cell, just similar to Fig. 8. We then use such percentages as the empirical distribution of price multipliers in each cell. Then, at the beginning of every iteration in the outer loop, we initialize the price multiplier of each cell, by randomly choosing a price multiplier based on the corresponding empirical distribution. In this way, the assign price multiplier not only represents the supply and demand variation in each cell in the hour, but also retains some randomness.
- **Choosing between exploration and exploitation** (line 6 to 10): as was mentioned earlier, the driver chooses a random action (i.e., exploration) with a small probability ϵ, and chooses the action with the highest $Q(s, a)$ at a given state s (i.e., exploitation) with a probability $1-\epsilon$.

- **Updating** $Q(s, a)$ (line 12): after taking action a, the driver jumps from state s to s', and the environment gives a reward r to the driver according to our MDP model. The corresponding Q-value, $Q(s, a)$, is updated based on the following equation of temporal difference:

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{\alpha'} Q(s', a') - Q(s, a)] \tag{4}$$

  In (4), α is the learning rate that is within the range $(0, 1]$ and controls the speed of convergence; γ is the discount factor that is within the range $[0, 1]$, representing the weight of future rewards. $Q(s', a')$ is the Q-values corresponding to taking any action a' at state s', and so $\max_{\alpha'} Q(s', a')$ means the maximum Q-value at state s'. The new $Q(s, a)$ is then used to update the original one.
- **Updating state and time** (line 13 to 14): finally, the state s and time t are updated to s' and t'.

V. Evaluation

We present the evaluation of our Q-learning model, including revenue evaluation and strategy evaluation. In revenue evaluation, we evaluate the effects of our model in increasing driver earnings; in strategy evaluation, we discuss the improvement of seeking strategy after applying our model.

A. Evaluation Setup

We simulate our Q-learning approach based on our datasets to verify its effectiveness. In the time range of our dataset, i.e., from Nov. 2015 to March. 2016, we choose a random Friday to simulate our approach. The chosen Friday should not be a holiday (such as Thanksgiving, Christmas, New Year, etc.), and even though we only present results of one Friday here, results from other Fridays and weekdays show similar effects.

Our focus is on the evening peak hours on this Friday. Firstly, our model runs at the unit of one hour; Secondly, we have already shown that during evening peak hours higher price multipliers are more common, and this is helpful to show the effects of introducing dynamic prices in our model. We choose the hour $[5pm, 6pm]$ to evaluate our model.

As to the number of drivers chosen to simulate, we randomly choose 500 out of the 3,650 drivers who work on this Friday. In some parts of evaluation, we pay attention to the "top 10%" (and "bottom 10%") drivers whose revenue are among the top 10% (and bottom 10%) of all drivers, and we randomly choose 100 drivers for each of these groups. The chosen drivers should satisfy the following criteria:

- they work for at least two hours on the chosen Friday, and their GPS trajectories have few errors;
- the number of orders they serve for on the chosen Friday should be greater than 0, and the orders should be effective (e.g., orders with close-to-zero trip duration or distance are considered as inaccurate and ineffective);
- they also work for most of other days.

These criteria ensure that these drivers work regularly and actively, and that their relevant data information are more accurate, helping us to avoid possible data inaccuracies.

Throughout our evaluation, we compare the results of the following three schemes:

- "**Real**": the results from our datasets (i.e., ground truth);
- "**Q-dp**": the results from our Q-learning model, with dynamic prices taken into consideration;
- "**Q**": the results from our Q-learning model, without dynamic prices considered. In other words, we keep dynamic price multipliers to be the fixed value 1.0 across all the cells during the hour.

B. Parameter Settings

We discuss how values of the following parameters are set.

$P_{pickup}(j)$: this is the pickup probability in cell j. To calculate pickup probability, we count the number of orders starting from cell j, denoted as $n_{pickup}(j)$, and the number of vacant car passing cell j during this hour, denoted as $n_{pass}(j)$. Then, the pickup probability is defined as:

$$P_{pickup}(j) = \frac{n_{pickup}(j)}{n_{pass}(j)} \tag{5}$$

$P_{dest}(j, k)$: this is the probability that a passenger picked up in cell j has a destination in cell k. We use $n_{j \to k}(j)$ to represent the number of orders that have a origin cell j and a destination cell k. Then the probability could be written as:

$$P_{dest}(j, k) = \frac{n_{j \to k}(j)}{n_{pickup}(j)} \tag{6}$$

$t_{drive}(j, k)$ and $d_{drive}(j, k)$: these are the time and distance to drive from cell j to cell k. If there exists orders that have the origin and destination cell as cell j and k, then we use the average driving time and distance as the estimation of $t_{drive}(j, k)$ and $d_{drive}(j, k)$. Otherwise (i.e., no order starts at cell j and ends at cell k), we find out all the trajectories that first pass cell j and then pass cell k, and use the average driving time and distance between these two cells to approximate $t_{drive}(j, k)$ and $d_{drive}(j, k)$.

$t_{seek}(j)$ and $d_{seek}(j)$: these are the average driving time and distance for a driver to seek in cell j. For simplicity, we let $d_{seek}(j) = 500m$, which is about half of the cell size. We also let $t_{seek}(j) = 1$ (minute) as the average driving speed is around 20 to 30 kilometres per hour, and it takes about 1 minute to drive for 500 meters.

Other parameters: we list the choice of α, β, ϵ and γ below:

- α: we set $\alpha = 0.1$ after testing many other choices.
- β: we set $\beta = 0.5$ – the driving cost is about 0.5 Yuan (RMB) per minute – according to previous studies (e.g., [15]).
- ϵ: we set $\epsilon = 0.3$ – the driver chooses exploration instead of exploitation with a probability of 0.3.
- γ: we set $\gamma = 0.5$. γ controls how future rewards are treated in updating Q-value. In our evaluation, we also show results of using different γs to justify our choice.

Also, the parameters f_0 and f_d are determined based on service provider policy. In our datasets, the service provider sets $f_0 = 15$ and $f_d = 2.8$, all in RMB Yuan.

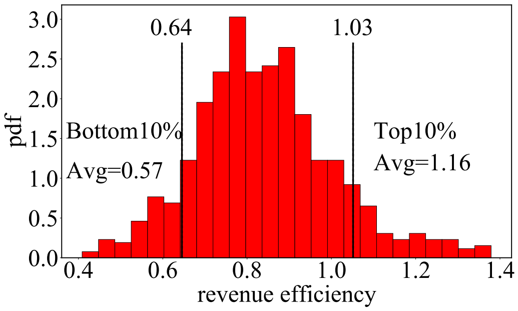


Fig. 12. Ground-truth: the pdf of revenue efficiency during $[5pm, 6pm]$ on the chosen Friday.

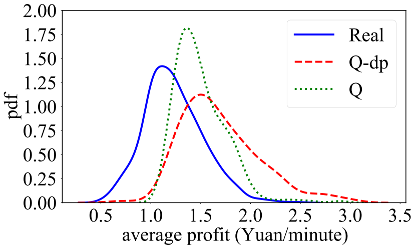


Fig. 13. The pdf of average profit of all drivers. (Real v.s. Q-dp v.s. Q.)

C. Revenue Evaluation

We first perform revenue evaluation – how our Q-learning model helps drivers to earn more.

Before presenting evaluation results, we define some necessary quantities and metrics. For a driver, we use F to denote the total revenue s/he makes during this hour; and use T_{order} as the sum of order durations during this hour. Also, we let T_{total} to represent the total working time during this hour (i.e., including the time of seeking and delivering passengers together). For most drivers, T_{total} is close to 60 minutes. Then we define the metrics used in revenue evaluation:

- **average profit** AP: it is the average revenue per minute during passenger delivery, and could be written as

$$AP = \frac{F}{T_{order}}. \tag{7}$$

- **revenue efficiency** RE: it is the average revenue per minute during both seeking and passenger delivery, i.e.,

$$RE = \frac{F}{T_{total}}. \tag{8}$$

- **utilization rate** UR: it measures the ratio of the time used in delivering passengers to the total working time:

$$UR = \frac{T_{order}}{T_{total}}. \tag{9}$$

These three metrics describe drivers' revenue-making capability from different perspectives: RE is the most comprehensive metric expressing a driver's revenue-making capability during this hour; AP emphasizes order quality as it pays attention to T_{order} instead of T_{total}; and UR explains the extent to which a driver makes himself/herself occupied.

Fig. 12 shows the probability distribution function (pdf) of revenue efficiency during $[5pm, 6pm]$ on the chosen Friday from our datasets (i.e., ground-truth). Among all drivers, the top 10% (and bottom 10%) drivers, in terms of revenue efficiency, are also identified. We observe that:

- For all drivers' revenue efficiency, the average is 0.84 Yuan/minute;
- For top 10% drivers' revenue efficiency, the lower bound is 1.03 Yuan/minute, and the average is 1.16 Yuan/minute;
- For bottom 10% drivers, the upper bound is 0.64 Yuan/minute, and the average is 0.57 Yuan/minute;
- There is a huge gap between top and bottom drivers. The average revenue efficiency of top drivers is 103.5% higher than that of bottom drivers. Hence, seeking route recommendation is necessary to narrow the gap.

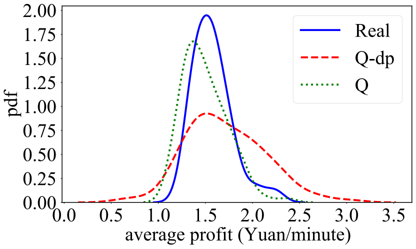


Fig. 14. The pdf of average profit of top 10% drivers. (Real v.s. Q-dp v.s. Q.)

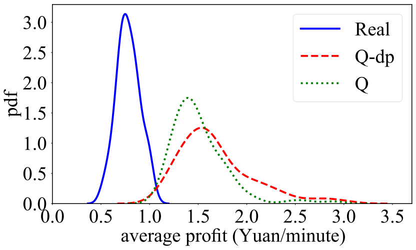


Fig. 15. The pdf of average profit of bottom 10% drivers. (Real v.s. Q-dp v.s. Q.)

TABLE II
THE MEAN OF AVERAGE PROFIT FOR ALL, TOP 10% AND BOTTOM 10% DRIVERS IN DIFFERENT EVALUATION SCHEMES

Evaluation scheme	**All drivers**	**Top 10% drivers**	**Bottom 10% drivers**
Real	1.23	1.57	0.78
Q	1.50	1.50	1.49
Q-dp	1.70	1.71	1.68

Fig. 13 to 15 show the pdf of average profit for all drivers, top 10% drivers and bottom drivers, respectively, and the mean value is listed in Tab. II. In each figure, we show the results from ground-truth, from Q-dp and Q together. It is shown that:

- Regarding the average profit among all drivers, the improvement between Q and real is 21.95%, and the improvement between Q-dp and Q is 13.3%. This indicates that both the Q-learning model and the incorporation of dynamic prices help increase average profit.
- The curves representing Q and Q-dp schemes are similar across these figures, and the mean values are also very close. In other words, after applying our Q-learning model, either with or without dynamic prices,

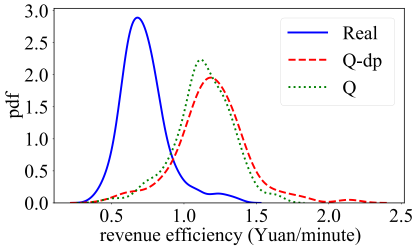


Fig. 16. The pdf of utilization rate of all drivers. (Real v.s. Q-dp v.s. Q.)

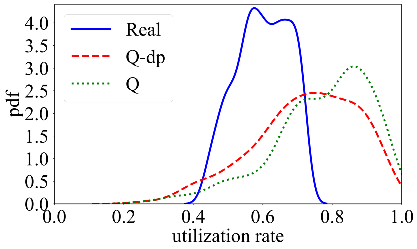


Fig. 17. The pdf of revenue efficiency of all drivers. (Real v.s. Q-dp v.s. Q.)

the differences between top and bottom drivers are highly eliminated.

- Moreover, comparing the curves of Q-dp and Q, it appears that Q-dp leads to a smoother distribution of average profit. Put it simply, the introduction of dynamic prices helps to further narrow the gap between drivers.

The above observations prove that it is not necessary to discuss metrics for top or bottom drivers separately after applying our Q-learning model, as they no longer show much differences. In the ground-truth, top drivers make more money because they know how to seek for passengers to obtain more or better orders based on personal experience, whereas bottom drivers fail to do that. With the Q-learning model, both top and bottom drivers follow the cell transition procedure based on Q-table, and even though they may start their seeking at different locations at the beginning of the hour, they soon become similar in terms of revenue-making capability. For this reason, in the following we don't distinguish between top and bottom drivers, and only show the results based on all drivers.

Fig. 16 shows the pdf of revenue efficiency among all drivers. Regarding revenue efficiency, we have similar observations with average profit. The mean values are 0.84, 1.13 and 1.20 for real, Q and Q-dp, respectively. The improvement between Q and real is 34.52%, and the improvement between Q-dp and Q is 6.19%. This, again, shows that both the Q-learning model and considering dynamic prices increase revenue efficiency.

As to the utilization rate, we have an interesting observation. Fig. 17 shows the pdf of utilization rate among all drivers, and the mean values are 0.60, 0.77 and 0.72 for real, Q and Q-dp, respectively. Curiously, even though Q and Q-dp increase utilization rate compared to the ground-truth, Q-dp leads to a mean utilization rate 6.5% lower than Q does. In other words, with dynamic prices considered, drivers have lower utilization rates. We hypothesize that drivers need more time to find out better orders (i.e., orders with high price multipliers). Hence, in Q-dp, while drivers indeed obtain better orders, they take less time in delivering passengers.

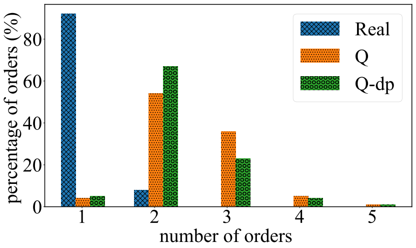


Fig. 18. The number of orders of all drivers. (Real v.s. Q-dp v.s. Q.)

TABLE III
THE MEAN OF UTILIZATION RATE DURING DIFFERENT TIME PERIODS

Evaluation scheme	Fri-17	Sat-17	Fri-8	Fri-12
Q	0.77	0.71	0.78	0.70
Q-dp	0.72	0.67	0.73	0.65

To verify that our observation is not a special case, we further evaluate our model during other representative time periods and show the mean of utilization rate among all drivers in Tab. III. We denote the original time period in the above discussions (i.e., [5pm, 6pm] on Friday) as "Fri-17", and also choose three other time periods: [5pm, 6pm] on Saturday ("Sat-17"), [8am, 9am] on Friday ("Fri-8"), and [12pm, 1pm] on Friday ("Fri-12"). These four time periods cover weekends and the evening peak, the morning peak, and an off-peak period around noon on weekdays.

It is shown in Tab. III that our observation regarding the utilization rate – drivers have lower utilization rates when dynamic prices are considered – holds for all these four representative time periods. In other words, drivers indeed take more time to search for orders, no matter on weekday or on weekend, during peak or off-peak hours, with the consideration of dynamic prices.

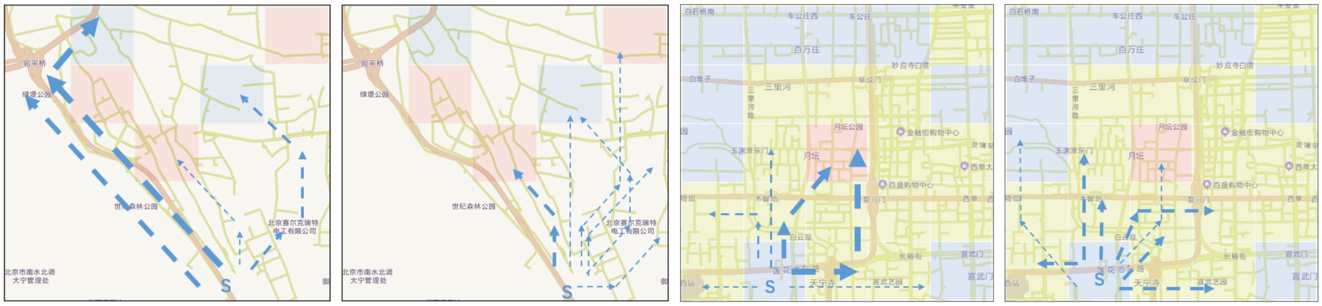
The above observations could be further verified by Fig. 18 that illustrates the number of orders drivers pick during one hour. In ground-truth, most drivers take only one order during this hour, with an average of 1.10. By comparison, drivers take 2.45 and 2.24 orders on average with Q and Q-dp. On one hand, our model, either Q or Q-dp, enables drivers to obtain more orders and earn more; on the other hand, when dynamic prices are considered, drivers take fewer orders, which has a similar effect with taking less time in passenger delivery.

Finally, we evaluate the effect of γ. We focus on the average revenue efficiency of Q-dp, and the figure is 1.14, 1.16, 1.20, 1.13, 1.13 for $\gamma = 0, 0.3, 0.5, 0.7, 1$, respectively. Therefore, setting $\gamma = 0.5$ achieves the highest revenue efficiency and is considered as a reasonable choice in most of our evaluation.

D. Strategy Evaluation

We now concentrate on the improvement of seeking strategy (i.e., "*how drivers seek?*") after applying Q-dp.

Specifically, we try to identify the seeking strategy and show the seeking trajectories in suburb and city center area.



(a) Suburb: seeking trajectories in gound-truth. (b) Suburb: seeking trajectories with Q-dp. (c) City center: seeking trajectories in gound-truth. (d) City center: seeking trajectories with Q-dp.

Fig. 19. Strategy evaluation: seeking trajectories under different circumstances. In these figures, (1) the cell with "S" is the starting cell; (2) lines with arrow are the representative trajectories, and the thicker the line, the more drivers choose it; (3) different colors are used to indicate the average price multiplier of a cell: red means high price multiplier (range [1.5, 1.6]), yellow means mid price multiplier (range [1.3, 1.4]), blue means low price multiplier (range [1.0, 1.2]), and no color means no order originates in the cell, possibly because it covers under-populated locations such as parks or highways.

To do that, we pick a starting cell in such an area, find out how drivers start seeking from this cell in ground-truth, simulate using Q-dp for 1000 times by placing a driver in the starting cell and following the driver's subsequent trajectories, and compare trajectories between real and simulated driver(s).

1) Seeking in Suburb: We first show the seeking trajectories in a chosen suburb area – located at the southwest corner of Beijing and containing 25 cells. Fig. 19 (a) & (b) illustrate the seeking trajectories in this suburb area, in ground-truth and with Q-dp, respectively. In these figures, the cell with "S" is the starting cell; lines with arrow are the representative trajectories, and the thicker a line is, the more drivers choose this line. We also use color to indicate the average price multiplier of a cell: red (high price multiplier), yellow (mid price multiplier) and blue (low price multiplier) color mean that the average price multiplier is in the range [1.5, 1.6], [1.3, 1.4] and [1.0, 1.2], respectively. If a cell has no color, it means that no order originates in the cell, possibly because it covers under-populated locations such as parks or highways.

In Fig. 19 (a) – seeking trajectories in ground-truth – 54.8% drivers leave the suburb area and go to city center to seek for passenger, and 45.2% choose to seek within the suburb area. The two thicker lines pointing to the northwest direction represent the common trajectories of leaving the suburb area, and they coincide with the major express way going into the city; other thinner lines represent seeking in red and blue cells where demand exists. Among those who choose to stay, 63.1% get orders, and the left 36.9% fail to pick up any passenger finally. It thus clear that in reality, more than half drivers in the suburb area choose to go to city center to seek for passenger, even though there are enough passengers waiting in the suburb. This may not be optimal, as drivers going to city center not only face fierce competition, but also pay a high driving cost as it is a long journey and the probability of finding passengers on the way is relatively low. Moreover, as we observe from Fig. 6 in section III-B, highest price multipliers often occur around city suburb, meaning that in some locations around suburb, demand indeed exceeds supply. If most drivers go to city center, then such demand is left unsatisfied, leading to a bad passenger experience.

By comparison, in Fig. 19 (b) – seeking trajectories with Q-dp – only 28.6% leave the suburb area and 71.4% choose to stay. Among those who choose to stay, 65.2% get orders, whereas the left 34.8% do not get any. It should also be noted that now no driver goes into the northwest direction (i.e., the major express way going into the city); in fact, those who leave the suburb area are attracted by red cells nearby that are located to the east of the chosen suburb area (not shown in the figure), represented by the lines going to the east and northeast direction. The above observations illustrate that:

- No driver chooses the northwest-bound express way because going into the city center is indeed sub-optimal, based on the expected reward calculated by our model.
- Our Q-learning model with dynamic prices guides drivers to stay in the local suburb area and find passengers. This not only increases driver revenue, but also improves passenger experience as they no longer need to wait for a long time because of a shortage of drivers.

2) Seeking in City Center: Similarly, Fig. 19 (c) & (d) show the seeking trajectories in a chosen city center area. This area also contains 25 cells and is located near the financial street and the second ring-road. In Fig. 19 (c), we have the following statistics:

- 67.3% drivers go directly to the red cell to seek, as shown by the two thicker lines. Among them, 5.9%, 32.5%, and 11.1% finally pick up passengers in high, mid, low cells, whereas the left 50.5% fail to pick up.
- 32.7% drivers do not choose the red cell and go to other cells instead. Among them, 28.1% and 12.2% finally pick up passengers in mid and low cells, whereas 59.7% fail.
- Considering all drivers, 4.0%, 31.0% and 11.5% finally pick up passengers in high, mid, and low cells, and 53.5% fail to pick up.

In Fig. 19 (d), we have the following statistics:

- Only 9.7% drivers go directly to the red cell, and among them 88.9% finally get an order in the red cell.
- Considering all drivers, 8.6%, 44.2% and 7.5% finally pick up passengers in high, mid, and low cells, and 39.7% fail to pick up.

These statistics suggest the following observations:

- In ground-truth most drivers have a naive strategy – simply going to high price cell – and such strategy is sub-optimal, as the large number of drivers in the high cell results in competition and a low pickup probability.
- Our Q-learning model, by considering the long-term reward, successfully avoids guiding drivers into the high price but crowded cell, and redistributes drivers in a number of mid cells more evenly. This not only makes the high price cell less crowded, but also significantly increases the pickup probabilities in city center.

VI. Summary and Discussions

We give a brief summary based on our evaluation results. The effects of our Q-learning model with dynamic prices could be attributed to the effects of considering long-term rewards and of dynamic prices. Besides, we present some discussions on the effects of exploration and exploitation, the effects of the window of optimization, model performance on weekends, training time and applicability, and avoiding recommending drivers to the same location.

A. The Effects of Considering Long-Term Rewards

An agent in Q-learning measures the utilities of taking different actions by considering long-term rewards, i.e., looking ahead. By trial-and-error, the agent finally obtains the optimal state-action pairs that maximize the expectation of the sum of rewards. Our results show that, considering long-term rewards has the following effects:

- On average, driver revenue is increased. The improvement of average profit, and revenue efficiency, is 21.95% and 34.52%, respectively.
- On average, the utilization rate is also increased by 28.33%. It is easier for drivers to find passengers.
- In ground-truth, there is a huge gap between top and bottom drivers in terms of revenue-making capability. After applying our model, the gap is highly eliminated.
- Drivers are distributed more evenly among cells with different price multipliers. Hence, the Q-learning model avoids guiding drivers into those cells that are crowded but have high price multipliers.

B. The Effects of Dynamic Prices

Our Q-learning model further takes dynamic prices into consideration, as the price multiplier serves as an accurate indicator of the supply and demand condition. This leads to the following effects:

- Driver revenue is further increased. The average profit and revenue efficiency are further increased by 13.3% and 6.19%.
- Utilization rate is slightly reduced. On average, the utilization rate is 6.5% lower.
- The above two effects show that, with dynamic prices considered, drivers spend more time in finding orders, but such orders are of higher quality: they have a higher dynamic price multiplier and generate more revenue.
- The gap between drivers is further narrowed. The distribution of average profit appears to be smoother.
- In suburb area, drivers are guided to stay in the suburb and serve local demand, instead of going directly to city center. This both increases driver revenue and improves passenger experience.

C. Exploration vs. ex ploitation

The trade-off between exploration and exploitation is controlled by ϵ. In our paper, we set $\epsilon = 0.3$ – the driver explores with a probability of 0.3 at each state. As our main goal is to incorporate dynamic prices into seeking route recommendation in RoD service, instead of evaluating the effects of ϵ, we do not discuss this trade-off in details in section V, but it is beneficial to have a brief discussion here.

To compare the effects of using different ϵs, we let $\epsilon = 0.05, 0.1, 0.3, 0.5, 0.7,$ and calculate the mean of average profit, revenue efficiency, utilization rate and the number of orders. We also record the convergence speed, i.e., the number of epochs before the Q-table converges. We observes that:

- When $\epsilon = 0.05$, the convergence speed is the fastest, taking 4350 epochs, but other metrics are the lowest. When ϵ is very small, there is almost no exploration, and the lack of randomness leads to fast convergence. Also, as we already mention, a small ϵ means that it is possible to get stuck in local instead of global optimum.
- When $\epsilon = 0.5$ or 0.7, the convergence speed is the slowest, taking more than 7700 epochs, and other metrics are also the lowest. The slow convergence is the result of the driver exploring without using the Q-table. Also, a large ϵ means that the driver keeps randomly choosing actions, losing the advantages of reinforcement learning.
- When $\epsilon = 0.1$ or 0.3, all the metrics are very close, and though the convergence speed when $\epsilon = 0.1$ is a little bit faster, the difference is only 0.7%. In these cases, the effects of exploration and exploitation cancel each other, leading to similar algorithm performances.

It is now safe to claim that setting $\epsilon = 0.3$ gives satisfactory results. The best choice of ϵ may be dependent on application scenarios, and should be evaluated case by case. Another common method of choosing an appropriate ϵ is to use larger values at the beginning of training, and then let it gradually decreases until convergence. After training, a small ϵ is used to recommend actions.

D. The Effects of the Window of Optimization

The window of optimization refers to the one-hour interval we use to perform route recommendation. In other words, we collect probabilities and average price multipliers, retrain the model, and obtain an updated Q-table every hour. The Q-table is used throughout the whole hour to recommend actions. There may be concerns that one hour is not long enough for a driver to look ahead (i.e., considering long-term rewards). We use the one-hour interval for the following reasons:

- In fact, [4], [14], [18] choose an one-hour interval, and [15] uses a three-hours interval. This indicates that either one-hour or three-hours (a much longer interval) is acceptable and appropriate.

- In the strategy evaluation in section V-D, we already show that our model helps drivers to look ahead and consider long-term rewards.

The most obvious difference between using the one-hour and three-hours interval is that three hours are long enough to capture more orders, especially those spanning across consecutive hours. We perform similar experiments using the three-hours interval, and results show that:

- For the number of orders, it should be around 3 times the number with the one-hour interval. In ground-truth, the number increases from 1.10 to 5.07, showing that we indeed miss some spanning-across orders with the one-hour interval. With Q (or Q-dp), the number increases from 2.45 (or 2.24) to 7.64 (or 7.56) – which is close to our anticipation.
- For utilization rate, the number with the three-hours interval is very close to that with the one-hour interval – 0.59 v.s 0.60 in ground-truth, 0.82 v.s 0.77, 0.81 v.s 0.72 with Q and Q-dp, respectively. The slight increases could also be attributed to the spanning-across orders.
- For average profit or revenue efficiency, observations are also similar. Quantities with three-hours interval are very close to that with one-hour interval.

E. Model Performance on Weekends

In section V, we only evaluate our model on a typical weekday, and we do not show results on weekends due to limited space. We also mention the two additional reasons of not evaluating on weekends in section III-B, i.e., the less severe supply-demand imbalance and the relatively stable price multipliers close to 1.0.

It is still necessary to present a concise discussion on the model's performance on weekends, not only to justify our above arguments, but to validate that our model works well on weekends as well. We perform similar experiments during the same time period [5pm, 6pm] on a random Saturday, and it is shown that:

- Comparing between "Q" and "real": On Friday, the increase of average profit and revenue efficiency are 21.95% and 34.52%, respectively, as shown in section V. On Saturday, the corresponding figures are 13.95% and 26.91%. Though looking ahead (by reinforcement learning) still increases driver revenue on Saturday, the amount of increase is significantly reduced.
- Comparing between "Q-dp" and "Q": On Friday, the increase of average profit and revenue efficiency are 13.33% and 6.19%, respectively. On Saturday, the corresponding figures are -2.1% and 0.5%. This validates our argument that dynamic pricing plays a less important role on weekends.

F. Training Time and Applicability

Training time is a key factor in applying our model to real practical applications. In fact, as the Q-learning model is retrained and the Q-table is updated every one hour in our study, the requirement on training time is loose to some extent.

Q-learning is a simple but effective algorithm, and a short training time is one of its advantages, compared to more complicated ones such as deep reinforcement learning. In our study, for a single driver, it takes less than 3 minutes to train our model, in about 5,000 to 6,000 epochs before Q-table converges. This is done on an ordinary PC with Intel i7-12700 CPU, and we anticipate that the training time could be further reduced if a faster system is used.

It is feasible to use our model in real practical applications. Firstly, as we already mention, the model update frequency is every one hour, which is much longer than 3 minutes. Even if finer granularity is desired and the update frequency is increased to, say, every half an hour, the training time is also small enough. Secondly, our model runs well on an ordinary PC, and does not require highly sophisticated hardware such as high-end GPUs, making it easier for practical deployment.

Additional discussions may be necessary when considering a large number of drivers. Firstly, as multiple Q-learning instances could run in parallel, our model is scalable to deal with a large number of drivers with enough computation resources. Secondly, when dealing with the behavior of multiple drivers, the interaction between drivers and passengers, or the interaction between drivers, needs to be considered. This requires using models such as multi-agent reinforcement learning and collecting data describing how drivers and passengers behave under different circumstances. We leave this as future work and describe it briefly in section VII.

G. Recommending Drivers to the Same Location

. A common problem in similar studies is that whether the algorithm guides drivers nearby or drivers with similar properties to the same location. If this happens, the number of drivers around such location soon becomes more than enough, giving rise to a sharp deterioration in algorithm performance. A complete solution to this problem may require the understanding of, say, drivers' emotional state, drivers' adoption rate of the recommended routes, changes to the environment after driver adoption, etc., which is hard to obtain unless systematic and laborious field tests are conducted, and hence we leave it for future work when field tests are possible. There are, however, heuristics to tackle this problem, such as generating a list of recommendations and randomly picking one for a driver.

We avoid recommending drivers to the same location based on the dynamic pricing mechanism. If the mechanism is well-designed, and could respond to the changes of supply and demand in time, or in real-time if possible, then a decreasing price multiplier would stop the Q-learning model from recommending a particular cell because of the reduced reward. We assume that the pricing mechanism related to our data satisfies this requirement, but the design of such mechanism is another story and is out of the scope of this paper.

VII. Conclusion and Future Work

In this paper we study the seeking route recommendation problem in RoD service: recommending the next cell to a seeking RoD driver. RoD service is more intelligent than traditional taxi service from two perspectives, i.e., *data driven* and *dynamic pricing*, and this enables us to take into account dynamic prices in recommending routes.

We design a reinforcement learning model with dynamic prices to tackle the problem. With reinforcement learning, long term effects of obtaining rewards are considered. With dynamic prices, we pay attention to the profitability of seeking in a particular cell as well as the supply and demand condition of that cell, instead of only focusing on the pick up probability.

Evaluation results show that, firstly, drivers' revenue efficiency and average profit are increased, and considering dynamic prices further increases both of them. Secondly, drivers' seeking strategy is improved, leading to a higher driver revenue and better passenger experience. For example, drivers are distributed more evenly instead of flocking to crowded cells with high price multipliers; and those in suburb area try to serve local demand first instead of going to city center.

In the near future, we would like to consider the scenario of seeking route recommendation for multiple drivers simultaneously. This adds plenty of complexity in both methodology and data. For methodology, multi-agent reinforcement learning is required. Things are more complicated for data. It is necessary to collect data that could describe the interaction between drivers, the interaction between passengers and drivers, the changes of passenger demand pattern due to price multiplier update, the adoption rate of recommended routes, etc. Collecting these data may be difficult and requires close collaboration with the service provider. We are actively working towards closer collaboration and deployment opportunity.

Acknowledgment

The authors would like to thank the anonymous editors and reviewers for their constructive feedback and comments.

References

[1] L. Rayle, D. Dai, N. Chan, R. Cervero, and S. Shaheen, "Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco," *Transp. Policy*, vol. 45, pp. 168–178, Jan. 2016.

[2] A. Brown and W. LaValle, "Hailing a change: Comparing taxi and ridehail service quality in Los Angeles," *Transportation*, vol. 48, no. 2, pp. 1007–1031, Apr. 2021, doi: 10.1007/s11116-020-10086-z.

[3] H.-W. Chang, Y.-C. Tai, and Y.-J. Hsu, "Context-aware taxi demand hotspots prediction," *Int. J. Bus. Intell. Data Mining*, vol. 5, no. 1, pp. 3–18, 2009.

[4] H. Rong, X. Zhou, C. Yang, Z. Shafiq, and A. Liu, "The rich and the poor: A Markov decision process approach to optimizing taxi driver revenue efficiency," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 2329–2334.

[5] X. Yu, S. Gao, X. Hu, and H. Park, "A Markov decision process approach to vacant taxi routing with e-hailing," *Transp. Res. B, Methodol.*, vol. 121, pp. 114–134, Mar. 2019.

[6] Y. Lai, Z. Lv, K.-C. Li, and M. Liao, "Urban traffic Coulomb's law: A new approach for taxi route recommendation," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3024–3037, Aug. 2019.

[7] S. Guo et al., "A force-directed approach to seeking route recommendation in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1909–1926, Jun. 2022.

[8] B. Li et al., "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 63–68.

[9] D. Zhang et al., "Understanding taxi service strategies from taxi GPS traces," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 123–135, Feb. 2014.

[10] C. Chen, Q. Liu, X. Wang, C. Liao, and D. Zhang, "Semi-Traj2Graph: Identifying fine-grained driving style with GPS trajectory data via multi-task learning," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1550–1565, Dec. 2021.

[11] C. Liao et al., "Enriching large-scale trips with fine-grained travel purposes: A semi-supervised deep graph embedding framework," *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 16, 2022, doi: 10.1109/TITS.2022.3203464.

[12] S. Guo et al., "ROD-revenue: Seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2202–2220, Sep. 2020.

[13] N. Garg and S. Ranu, "Route recommendations for idle taxi drivers: Find me the shortest route to a customer!" in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1425–1434.

[14] X. Zhou et al., "Optimizing taxi driver profit efficiency: A spatial network-based Markov decision process approach," *IEEE Trans. Big Data*, vol. 6, no. 1, pp. 145–158, Mar. 2020.

[15] Z. Shou, X. Di, J. Ye, H. Zhu, H. Zhang, and R. Hampshire, "Optimal passenger-seeking policies on E-hailing platforms using Markov decision process and imitation learning," 2019, *arXiv:1905.09906*.

[16] C.-M. Tseng, S. C.-K. Chau, and X. Liu, "Improving viability of electric taxis by taxi service strategy optimization: A big data study of New York city," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 817–829, Mar. 2019.

[17] Y. Gao, D. Jiang, and Y. Xu, "Optimize taxi driving strategies based on reinforcement learning," *Int. J. Geographical Inf. Sci.*, vol. 32, no. 8, pp. 1677–1696, Aug. 2018.

[18] M. Han, P. Senellart, S. Bressan, and H. Wu, "Routing an autonomous taxi with reinforcement learning," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 2421–2424.

[19] C. Yan, H. Zhu, N. Korolko, and D. Woodard, "Dynamic pricing and matching in ride-hailing platforms," *Nav. Res. Logistics*, vol. 67, no. 8, pp. 705–724, Nov. 2019.

[20] H. A. Chaudhari, J. W. Byers, and E. Terzi, "Putting data in the driver's seat: Optimizing earnings for on-demand ride-hailing," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 90–98.

[21] A. Picchi. (2016). *Uber vs. Taxi: Which Is Cheaper?*. [Online]. Available: http://bit.ly/2DMgrMc

[22] Y. M. Nie, "How can the taxi industry survive the tide of ridesourcing? Evidence from Shenzhen, China," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 242–256, Jun. 2017.

[23] J. D. Hall, C. Palsson, and J. Price. (2017). *Is Uber a Substitute or Complement for Public Transit?*. [Online]. Available: https://bit.ly/2K6Vs7L

[24] T. Berger, C. Chen, and C. B. Frey, "Drivers of disruption? Estimating the uber effect," *Eur. Econ. Rev.*, vol. 110, pp. 197–210, Nov. 2018.

[25] J. Hall, C. Kendrick, and C. Nosko. (Oct. 2015). *The Effects of Uber's Surge Pricing: A Case Study*. [Online]. Available: http://bit.ly/2kayk9O

[26] J. Gan, B. An, H. Wang, X. Sun, and Z. Shi, "Optimal pricing for improving efficiency of taxi systems," in *Proc. 22th Int. Joint Conf. Artif. Intell.*, 2013, pp. 2811–2818.

[27] L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero. (2014). *App-Based, on-Demand Ride Services: Comparing Taxi and Ridesourcing Trips and User Characteristics in San Francisco*. [Online]. Available: http://bit.ly/2kVkahg

[28] L. Chen, A. Mislove, and C. Wilson, "Peeking beneath the hood of Uber," in *Proc. ACM Conf. Internet Meas. Conf.* New York, NY, USA: ACM, 2015, pp. 495–508.

[29] S. Guo et al., "A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data," in *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, 2018, p. 112.

[30] S. Guo, C. Chen, Y. Liu, K. Xu, and D. M. Chiu, "Modelling passengers' reaction to dynamic prices in ride-on-demand services: A search for the best fare," in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, 2018, p. 136.

[31] S. Guo, Y. Liu, K. Xu, and D. Ming Chiu, "Understanding ride-on-demand service: Demand and dynamic pricing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 509–514.

[32] H. Chen et al., "InBEDE: Integrating contextual bandit with TD learning for joint pricing and dispatch of ride-hailing platforms," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 61–70.

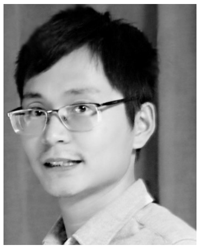
[33] M. K. Chen, "Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform," in *Proc. ACM Conf. Econ. Comput.* New York, NY, USA: ACM, 2016, p. 455.

[34] P. Cohen, R. Hahn, J. Hall, S. Levitt, and R. Metcalfe. (2016). *Using Big Data to Estimate Consumer Surplus: The Case of Uber*. [Online]. Available: http://bit.ly/2pqXiWo

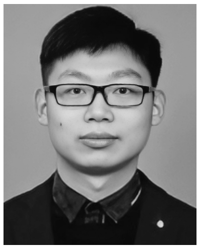
Suiming Guo received the Ph.D. degree from The Chinese University of Hong Kong. He is currently an Associate Professor with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include data mining, urban computing, pervasive computing, and smart cities studies.

Qianrong Shen is currently pursuing the master's degree with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include urban computing and smart cities studies.

Zhiquan Liu received the Ph.D. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2017. He is currently an Associate Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His current research interests include trust management and privacy preservation in vehicular networks and UAV networks.



Chao Chen received the Ph.D. degree from UMPC (Paris 6) and Telecom SudParis. He is currently a Full Professor of computer science with Chongqing University, China. His research interests include pervasive computing, social network analysis, and mobile crowdsensing.



Chaoxiong Chen received the bachelor's degree from the College of Informatics, Huazhong Agricultural University, China, in 2015, and the master's degree from the College of Computer Science, Chongqing University, China, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include data mining, pervasive computing, and big data analytics for smart cities.

Jingyuan Wang (Member, IEEE) received the Ph.D. degree from Tsinghua University. He is currently a Full Professor with Beihang University. His research interests include data mining and machine learning, with special interests in smart cities.

Zhetao Li (Member, IEEE) received the B.Eng. degree from Xiangtan University in 2002, the M.Eng. degree from Beihang University in 2005, and the Ph.D. degree from Hunan University in 2010. From 2013 to 2014, he was a Post-Doctoral Researcher in wireless network at Stony Brook University. He is currently a Professor with the College of Information Science and Technology, Jinan University. He is a member of CCF.

Ke Xu (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University. He is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include next generation internet, P2P systems, the Internet of Things (IoT), network virtualization, and optimization.