2019 IEEE 35th International Conference on Data Engineering (ICDE)

A Queueing-Theoretic Framework for Vehicle Dispatching in Dynamic Car-Hailing

Peng Cheng
HKUST
Hong Kong, China
pchengaa@cse.ust.hk

Chao Feng
AI Labs, DiDi Chuxing
Beijing, China
fengchaodavid@didichuxing.com

Lei Chen
HKUST
Hong Kong, China
leichen@cse.ust.hk

Zheng Wang
AI Labs, DiDi Chuxing
Beijing, China
wangzhengzwang@didiglobal.com

***Abstract*—With the rapid development of smart mobile devices, the car-hailing platforms (e.g., Uber or Lyft) have attracted much attention from both the academia and the industry. In this paper, we consider an important dynamic car-hailing problem, namely *maximum revenue vehicle dispatching* (MRVD), in which rider requests dynamically arrive and drivers need to serve as many riders as possible such that the entire revenue of the platform is maximized. We prove that the MRVD problem is NP-hard and intractable. To handle the MRVD problem, we propose a queueing-based vehicle dispatching framework, which first uses existing machine learning algorithms to predict the future vehicle demand of each region, then estimates the idle time periods of drivers through a queueing model for each region. With the information of the predicted vehicle demands and estimated idle time periods of drivers, we propose one batch-based vehicle dispatching algorithm to efficiently assign suitable drivers to riders such that the expected entire revenue of the platform is maximized during each batch processing. Through experiments over real data sets, we demonstrate the efficiency and effectiveness of our proposed framework.**

I. Introduction

Recently, with the popularity of the smart devices and high quality of the wireless networks, people can easily access network and communicate with online services. With the convenient car-hailing platforms (e.g., Uber [4] and DiDi Chuxing [1]), drivers can share their vehicles to riders to obtain monetary benefits and alleviate the pressure of public transportation. One of the crucial issues in the platforms is to efficiently dispatch vehicles to suitable riders. Although the platforms have become huge recently, during peak hours (e.g., 8 am) in some high demand areas (e.g., residential areas), riders need to wait for up to several hours before being served. To mitigate the shortage of vehicles in particular time and areas and improve the efficiency of the platforms, we investigate a queueing-theoretic dispatching framework in this paper, which aims to take riders' destinations into consideration to alleviate the shortage of taxis in particular areas such that the overall revenue of the platform can be maximized. Existing works in spatial matching or taxi dispatching only consider the pickup locations of riders and try to minimize the travel distance of taxis to pick riders [12], [10], which causes that some taxis will be hard to pick up new convenient riders after their last riders unless move long distances to far away riders and results in the low efficiency of the platform. In [6], the authors focused on improve the experience of riders through maximizing their utility. In [16], authors utilize auction mechanism to use the vehicle resources more efficiently in rush hours.

In this paper, we propose a *batch-based queueing-theoretic vehicle dispatching* framework. Specifically, we partition the whole space into regions and maintain a queue of waiting riders and drivers for each region. Once there are available drivers, the most-priority rider in the queue of waiting riders will be served. We first propose models to estimate the Poisson distributions of riders and drivers, then utilize the queueing theory to analyze the idle time interval for each driver after finishing his/her assigned rider. Finally, we propose one vehicle/driver dispatching algorithms to maximize the overall revenue of the platform in each batch processing. Note that, we maximize the overall revenue of the platform through improving the efficiency of the entire platform without increasing the charges to riders or decreasing the payment to drivers. In fact, the payment to drivers is usually a portion of the overall revenue of the platform. Thus, the more the overall revenue of the platform is, the more the payment to drivers is. In conclusion, our solution will benefit riders, drivers and the platform at the same time. In this paper, we proposed a batch-based queueing theoretic framework for vehicle dispatching in Section III, and conducted experiments on real data sets to show the efficiency and effectiveness of our queueing-theoretic framework in Section IV.

II. Problem Definition

In this paper, we use a graph $G = \langle V, E \rangle$ to represent a road network, where V is a set of vertices and E is a set of edges. Each edge $(u, v) \in E$ $(u, v \in V)$ is associated with a weight $cost(u, v)$ indicating the travel cost from vertex u to vertex v. Here the travel cost could be the travel time or the travel distance. When we know the travel speed of vehicles, we can easily convert one to another. In the rest of this paper, we will not differentiate between them and use *travel cost* consistently. To better manage the riders and drivers, we assume the entire space is divided into a set of n regions/grids $A = \{a_1, a_2, ..., a_n\}$.

Definition 1. (Rider) Let r_i be a rider, who submit his/her order o_i to the platform at timestamp t_i, and is associated with a source location s_i, a destination location e_i and a pickup deadline τ_i.

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00157

If a rider r_i is delivered to his/her destination, the platform will charge him/her for $\alpha \cdot cost(s_i, e_i)$, where α is the travel fee rate of the platform.

Definition 2. (Driver) Let d_j be a driver, who is located position $l_j(t)$ at timestamp t. Her status is either *busy* (i.e., on delivering any rider) or *available* (i.e., free to be assigned to a rider).

For region a_k at timestamp t, we denote the set of available drivers as $D_k(t)$ and the number of them as $|D_k(t)|$.

Definition 3. (Valid Rider-and-Driver Dispatching Pair) Let $\langle r_i, d_j \rangle$ be a valid rider-and-driver dispatching pair, where driver d_j can arrive at the pickup location s_i of rider r_i before the pickup deadline τ_i and driver d_j is in available status when he/she is picking up rider r_i.

Definition 4. (Maximum Revenue Vehicle Dispatching Problem, MRVD) For a given time period $\mathbb{T}$, a set of riders $R_{\mathbb{T}}$ and a set of drivers $D_{\mathbb{T}}$ may dynamically join or leave the platform. The MRVD problem is to select a set, $I_{\mathbb{T}}$, of valid rider-and-driver dispatching pairs such that the overall revenue of the platform is maximized, which is:

$$\max \sum_{\langle r_i, d_j \rangle \in I_{\mathbb{T}}} \alpha \cdot cost(s_i, e_i), \tag{1}$$

where α is the travel fee rate of the platform.

Hardness of MRVD. MRVD is NP-hard through a reduction from the 0-1 Knapsack problem [13], which is a well know NP-hard problem. In short, for each given 0-1 Knapsack problem, we can translate it to a MRVD problem through careful construct a special road network. In detail, for a given item i in 0-1 Knapsack problem, we can generate a rider r_i, such that the travel cost to serve the rider is equal to the cost of item i and the revenue is equal to the value of item i. Then, we let there is only one driver and each time after he/she finishes one ride request, he/she must come back to the center node as the constraint of the road network. Thus, MRVD is NP-hard.

Reduction of MRVD. We introduce a reduction to reveal practical rules to dispatching drivers to maximize the overall revenue of the platform. Let T_j be the lifetime of driver d_j from the time he/she joins to the time he/she exits the platform. We notice that only when driver d_j is *busy*, he/she contributes to the overall revenue of the platform. Then we can rewrite the objective function of MRVD as below:

$$\begin{aligned} &\max \sum_{\langle r_i, d_j \rangle \in I_{\mathbb{T}}} \alpha \cdot cost(s_i, e_i) \\ \Rightarrow &\max \sum_{d_j \in D_{\mathbb{T}}} \sum_{r_i \in R_j} \alpha \cdot cost(s_i, e_i) \\ \Rightarrow &\max \alpha \sum_{d_j \in D_{\mathbb{T}}} \sum_{r_i \in R_j} cost(s_i, e_i) \end{aligned} \tag{2}$$

where $D_{\mathbb{T}}$ is the set of drivers on the platform during the given time period $\mathbb{T}$, and R_j is the set of riders that are served by driver d_j. According to Equation 2, the platform should maximize the length of the total busy time of each driver to maximize its overall revenue. Since the lifetime T_j of each driver d_j is fixed, to maximize his/her total *busy* time, $\sum_{r_i \in R_j} cost(s_i, e_i)$, is equivalent to minimize his/her total idle time, $T_j - \sum_{r_i \in R_j} cost(s_i, e_i)$. Then, the objective of MRVD can be rewritten as follows:

$$\min \sum_{d_j \in D_{\mathbb{T}}} \left(T_j - \sum_{r_i \in R_j} cost(s_i, e_i) \right) \Rightarrow \min \sum_{d_j \in D_{\mathbb{T}}} \sum_{i=0}^{|R_j|} \psi_{ij} \tag{3}$$

where ψ_{ij} is the idle time of driver d_j after delivering rider r_i. Here ψ_{0j} indicates the idle time of driver d_j before picking up his/her first rider.

According to Equation 3, to maximize the overall revenue, the platform intuitively should reduce the number of served riders (e.g., $|R_j|$) and the time interval (e.g., ψ_{ij}) between any two consecutive riders for each driver. Then, we can have two practical and controllable rules for the platform to maximize its overall revenue during a given time period $\mathbb{T}$: **a)** ***associating higher priorities to the riders whose travel costs are high*****; b)** ***reducing the length of the idle time between serving any two consecutive riders for each driver.***

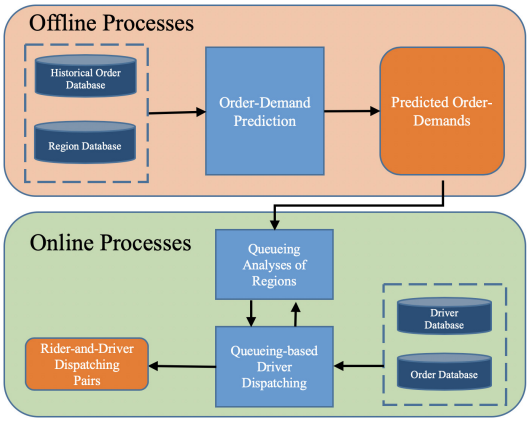


Fig. 1. Illustration of the Framework Work Flow.

III. Queueing-Based Dispatching Framework

In this section, we introduce an overview of our queueing-based vehicle dispatching framework. Figure 1 shows the overall work flow of the platform.

A. Offline Vehicle Demand-Supply Prediction

In practice, it is hard to predict the accurate location and timestamp of a particular rider or driver since the uncertain behaviors of a single user. To utilize the distribution of demand and supply of riders, we predict the number of riders and drivers for a given region (i.e., a spatial range of area, such as square regions or hexagon regions) in a given time period (i.e., next 5 minutes). For the rejoined drivers, we can easily estimate their availability based on their assignments and travel costs. For the newly coming riders, existing work can be applied offline to predict the demand of riders, such as demand-supply prediction of traffic [9], [7], and spatial-temporal data prediction [14], [8]. In this paper, we tested the representative state-of-the-art prediction algorithms on the real-world taxi

demand-supply data set and select the most effective one [15] for our offline Vehicle demand-supply prediction process.

B. Region Queueing Analysis

The available drivers in a region a_x in a time period $\mathbb{T}_y$ comes from the rejoined active drivers. With the predicted numbers of orders for the region a_x in a given time period $\mathbb{T}_y$ and the schedules of active drivers, we can know the demand and supply of drivers for the region a_x in time period $\mathbb{T}_y$.

Similar to the previous assumption in the related work [5], we assume that the arrivals of rejoined active drivers follow the Poisson distribution with rates λ_x in a region a_x during a short time period $\mathbb{T}_y$ with length $t_c = |\mathbb{T}_y|$ (e.g., a half hour). In addition, we also model the arrival rate of riders (in number per minute) follow a Poisson distribution with a rate of μ_x in a region a_x during a short time period with length t_c. Note that, although the arrival rate of riders and drivers may change during different time periods in a day (e.g., 8 to 9 A.M. and 8 to 9 P.M.), to facilitate the analysis of the queueing situation in a short time period, we model the arrival rates of riders and drivers as stable rates.

According to the queueing theory [11], when $\lambda_x < \mu_x$, the average length L_x of waiting drivers in region a_x can be estimated through $L_x = \frac{\rho^2}{1-\rho}$, where $\rho = \frac{\lambda_x}{\mu_x}$. Then, the average waiting time of each driver is $ET(\lambda_x, \mu_x) = \frac{L_x}{\lambda_x} = \frac{\lambda_x}{\mu_x(\mu_x - \lambda_x)}$. In our framework, we avoid assigning too many drivers to region a_x such that λ_x is always smaller than μ_x.

C. Queueing-Based Vehicle Dispatching

In this part, we iteratively assign drivers to riders every Δ seconds. To solve the assignment problem in each batch, we propose one heuristic algorithm to greedily maximize the revenue summation of the platform for the current scheduling time period $[\bar{t}, \bar{t}+t_c]$, where $\bar{t}$ indicates the current timestamp and t_c is the length of the current scheduling time period. Specifically, the current scheduling time period $[\bar{t}, \bar{t}+t_c]$ is a time period where we consider the arrivals of drivers and riders follow Poisson distributions for each region.

We propose an *idle ratio oriented greedy* approach to solve each batch process with a goal to maximize the revenue summation of the platform during the current scheduling time period $[\bar{t}, \bar{t}+t_c]$, where $\bar{t}$ is the current timestamp and t_c is the length of the current scheduling time window. We first define the idle ratio of driver d_j to server rider r_i, whose destination e_i is in region a_k, as follows:

$$IR(r_i, d_j) = \frac{ET(\lambda_{(k)}, \mu_{(k)})}{cost(s_i, e_i) + ET(\lambda_{(k)}, \mu_{(k)})}, \tag{4}$$

where $ET(\lambda_{(k)}, \mu_{(k)})$ is the expected idle time of driver d_j when he/she rejoins the platform at region a_k, and $cost(s_i, e_i)$ is the travel cost (travel time) on serving rider r_i. We notice that when the travel cost $cost(s_i, e_i)$ increases, $IR(r_i, d_j)$ will decrease; when the expected idle time $ET(\lambda_{(k)}, \mu_{(k)})$ increases, $IR(r_i, d_j)$ will also decrease. As a result, we only need to greedily select the rider-and-driver dispatching pairs with low idle ratio, then we can follow the two guiding rules in the end of Section II to maximize the overall revenue of the platform. Then, we propose an idle ratio oriented greedy approach as shown in Algorithm 1, which greedily select the rider-and-driver dispatching pair having the smallest current idle ratio value.

Algorithm 1: Idle Ratio Oriented Greedy Algorithm

Input: A set of Regions A, current timestamp $\bar{t}$
Output: A set of rider-and-driver dispatching pairs $I_{\bar{t}}$

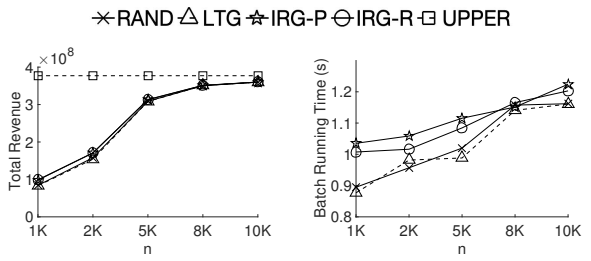
1 $I_{\bar{t}} \leftarrow \{\emptyset\}$, $I_v \leftarrow \{\emptyset\}$
2 **foreach** $a_k \in A$ **do**
3 retrieve a set I_k of valid rider-and-driver dispatching pairs from R_k and D_k
4 $I_v \leftarrow I_v \cup I_k$
5 estimate the arrival rate $\lambda_{(k)}$ of rejoined drivers and arrival rate $\mu_{(k)}$ of riders in region a_k in $[\bar{t}, \bar{t}+t_c]$
6 sort dispatching pairs in I_v based on their idle ratio
7 **while** I_v *is not empty* **do**
8 select the rider-and-driver pair $\langle r_i, d_j \rangle$ having the smallest idle ratio from I_v
9 add $\langle r_i, d_j \rangle$ to $I_{\bar{t}}$
10 update $\lambda_{(k)}$ of the destination region a_k of r_i
11 remove $\langle r_i, . \rangle$ and $\langle ., d_j \rangle$ from I_v
12 **return** $I_{\bar{t}}$

Complexity Analysis. Let the number of total waiting riders be m, the number of total available drivers be n and the number of total regions be x. Assume riders and drivers be evenly distributed in x regions and x is much smaller than m and n. In lines 3-6 of Algorithm 1, retrieving all the valid rider-and-driver pairs needs $O(\frac{mn}{x})$. To sort the valid pairs in I_v needs $O(\frac{mn}{x}\log_2(\frac{mn}{x}))$ (line 7). In each iteration of the while-loop (lines 8 - 12 of Algorithm 1), selecting the pair having the smallest idle ratio from sorted I_v needs $O(1)$ (lines 9-10); updating $\mu_{(k)}$ and the idle ratio of average $\frac{mn}{x^2}$ related pairs needs $O(\frac{mn}{x^2})$ (line 11); removing the related valid pairs $\langle r_i, . \rangle$ and $\langle ., d_j \rangle$ from I_v needs $O(\max(\frac{n}{x}, \frac{m}{x}))$ (line 12). Since in each iteration, at least one rider and one driver will be matched, thus there will be at most $\min(m, n)$ iterations. Then the complexity of the while-loop is $O(\frac{\min(m,n)mn}{x^2})$. Thus, the complexity of Algorithm 1 is $O(\max(\frac{mn}{x}\log_2(\frac{mn}{x}), \frac{\min(m,n)mn}{x^2}))$.

IV. Experimental Study

Data Sets. New York Taxi and Limousine Commission (TLC) Taxi Trip Data [2] is a data set recording the information of taxi trips in New York, USA. We use the taxi trip records of yellow and green taxis in our experiments. Each trip record includes the taxi's pick-up and drop-off taxi-zones and timestamps, the number of passengers and the total travel cost. In the data set, there are 262 taxi zones from zone 2 to zone 263. In our experiments, we use taxi trip data records from January 2018 to June 2018.

Experimental Configurations. We use the pickup location and timestamp of a taxi trip record to initialize the source location s_i and the posting timestamp t_i of a ride order r_i. Then the dropoff location of the taxi trip record is used to set



(a) Total Revenue (b) Batch Running Time

Fig. 2. Effects of Number of Drivers n.

the destination e_i of the ride order. For the pickup deadline τ_i of rider r_i, we configure it by adding a random noise $\tau' \in [1, 10]$ and a base pickup waiting time of 120 seconds to the posting timestamp t_i (e.g., $\tau_i = t_i + \tau' + 120$). To initialize the origin location $l_j(0)$ of driver d_j at the beginning timestamp 0, we first randomly generate zone a_k for the driver following the distribution of the number of orders over the whole 262 zones. Then uniformly generate the location $l_j(0)$ in zone a_k.

In our experiment, we run the batch process every 10 seconds. To estimate the arrival rate of new riders and rejoined drivers, we look up a time window of length t_c of 5 minutes with the "current" timestamp $\bar{t}$ as the beginning point.

Approaches and Measurements We evaluate the effectiveness and efficiency of our queueing-theoretic vehicle dispatching framework with a batch processing vehicle dispatching algorithm, namely *idle ratio oriented greedy* (IRG), in terms of the total revenue and the average batch running time. Specifically, for IRG we can further have two different combinations: IRG-P and IRG-R, which use the predicted taxi demand and the real taxi demand, respectively. In addition, we compare our framework with two baseline methods, namely *long trip greedy* (LTG) which greedily assigns orders with the highest revenue to available taxis, and *random* (RAND) which randomly assigns orders to available taxis. We report the average total revenue and the average batch processing time of our tested approaches for a whole day (from 00:00:00 to 23:59:59). All our experiments were run on an Intel Xeon X5675 CPU @3.07 GHZ with 32 GB RAM in Java. The code of our queueing-theoretic vehicle dispatching framework and prediction methods can be accessed in our GitHub project [3].

Effect of the Number, n, of Drivers. Figure 2 illustrates the experimental results on varying the number of drivers from 1K to 10K. In Figure 2(a), when the number of drivers increases from 1K to 10K, all the tested approaches can achieve results with increasing total revenue. The reason is that when more drivers are available, more riders can be served before their pickup deadlines. When the number of drivers is 1K, our IRG and LS approaches can achieve higher total revenue than RAND and LTG. The difference between the results of our IRG and LS are small. We will discuss the details in later and clearer results figures. When the number of drivers increases, the advantage of our IRG and LS in terms of the total revenue become narrow. We also notice that when the number of drivers reaches 10K, all the tested approaches can achieve results with total revenue close to the upper bound. The reason is that when there are 10K drivers, almost all the riders can be served as long as he/she joins the platform. The vehicle dispatching algorithms have no difference in term of the total revenue, when drivers are sufficient. To clearly show the differences between the total revenues of our tested approaches, we will not plot out the results of UPPER as they are always same with the results in Figure 2(a).

In Figure 2(b), when the number of drivers increases, the batch running time of all the tested approaches also increases, which is because in each batch there are more drivers to process requiring more time to process. We can see that all the tested approaches can finish each batch processing within 2 seconds, which is unnoticeable to the users.

In summary, IRG, in terms of the total revenue, can beat RAND and LTG. Real taxi demand can result in the higher total revenue. Thus, taxi demand prediction models with higher accuracy are more valuable for the platform. Our framework is efficient. In all the experiments, the running time of each batch for all the tested approaches is less than 2 seconds, which is affordable for the platform.

V. Acknowledgment

The work is partially supported by the Hong Kong RGC GRF Project 16207617, the National Science Foundation of China (NSFC) under Grant No. 61729201, Science and Technology Planning Project of Guangdong Province, China, No. 2015B010110006, Hong Kong ITC ITF grants ITS/391/15FX and ITS/212/16FP, Didi-HKUST joint research lab project, Microsoft Research Asia Collaborative Research Grant and Wechat Research Grant. Peng Cheng is the corresponding author.

References

[1] [online] DiDi Chuxing. https://www.didichuxing.com.
[2] [online] NYC Taxi & Limousine Commission Dataset. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[3] [online] Source Code. https://github.com/haidaoxiaofei/queueing-car-hailing.
[4] [online] Uber. https://www.uber.com.
[5] S. Banerjee, R. Johari, and C. Riquelme. Dynamic pricing in ridesharing platforms. *ACM SIGecom Exchanges*, 2016.
[6] P. Cheng, H. Xin, and L. Chen. Utility-aware ridesharing on road networks. In *ACM SIGMOD*, 2017.
[7] J. Chu, K. Qian, et al. Passenger demand prediction with cellular footprints. In *IEEE SECON*, 2018.
[8] N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.
[9] Y. Li, Y. Zheng, H. Zhang, and L. Chen. Traffic prediction in a bike-sharing system. In *ACM SIGSPATIAL*, 2015.
[10] K. T. Seow, N. H. Dang, and D.-H. Lee. A collaborative multiagent taxi-dispatch system. *IEEE T-ASE*, 2010.
[11] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris. *Fundamentals of queueing theory*, volume 399. John Wiley & Sons, 2018.
[12] Y. Tong, J. She, et al. Online minimum matching in real-time spatial data: experiments and analysis. *PVLDB*, 2016.
[13] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
[14] J. Zhang, Y. Zheng, et al. Dnn-based prediction model for spatio-temporal data. In *ACM SIGSPATIAL*, 2016.
[15] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. *AAAI*, 2017.
[16] L. Zheng, P. Cheng, and L. Chen. Auction-based order dispatch and pricing in ridesharing. In *IEEE ICDE*, 2019.