When Recommender Systems Meet Fleet Management: Practical Study in Online Driver Repositioning System

Zhe Xu, Chang Men, Peng Li, Bicheng Jin, Ge Li, Yue Yang, Chunyang Liu, Ben Wang, Xiaohu Qie
DiDi Chuxing
{xuzhejesse,menchang,lipengkarllee,jinbicheng,lige,peanutyangyue_i,liuchunyang,wangben,tiger.qie}@didiglobal.com

ABSTRACT

E-hailing platforms have become an important component of public transportation in recent years. The supply (online drivers) and demand (passenger requests) are intrinsically imbalanced because of the pattern of human behavior, especially in time and locations such as peak hours and train stations. Hence, how to balance supply and demand is one of the key problems to satisfy passengers and drivers and increase social welfare. As an intuitive and effective approach to address this problem, driver repositioning has been employed by some real-world e-hailing platforms. In this paper, we describe a novel framework of driver repositioning system, which meets various requirements in practical situations, including robust driver experience satisfaction and multi-driver collaboration. We introduce an effective and user-friendly driver interaction design called "driver repositioning task". A novel modularized algorithm is developed to generate the repositioning tasks in real time. To our knowledge, this is the first industry-level application of driver repositioning. We evaluate the proposed method in real-world experiments, achieving a 2% improvement of driver income. Our framework has been fully deployed in the online system of DiDi Chuxing and serves millions of drivers on a daily basis.

CCS CONCEPTS

• **Applied computing** → **Transportation**; *Multi-criterion optimization and decision-making*; • **Computing methodologies** → *Multi-agent systems.*

KEYWORDS

driver repositioning, recommender system, fleet management

ACM Reference Format:
Zhe Xu, Chang Men, Peng Li, Bicheng Jin, Ge Li, Yue Yang, Chunyang Liu, Ben Wang, Xiaohu Qie. 2020. When Recommender Systems Meet Fleet Management: Practical Study in Online Driver Repositioning System. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3366423.3380287

1 INTRODUCTION

Taxi services provide flexible door-to-door transit and play an important role in public transportation. To achieve this flexibility, taxi drivers need to cruise on streets and try to find the next passenger through physical meeting. The passenger-seeking process, which may occupy up to 50 percent of the drivers' time, introduces additional cost to taxi drivers and also public road resources. It has been researched extensively on how to improve the efficiency and experience of taxi cruising by route recommendation [14, 28, 30].

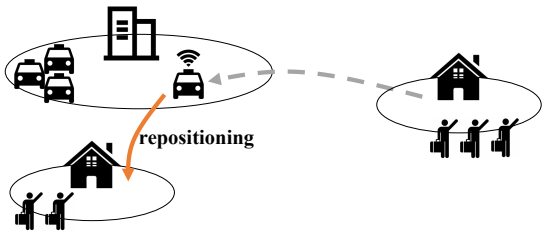


Figure 1: Motivation of driver repositioning.

Over the past few years, e-hailing service platforms [41] such as Uber, Lyft and DiDi Chuxing emerged as alternatives of traditional taxi services and created a new chance of revisiting the aforementioned problem. In these platforms, drivers and passengers report their status to the platform in real-time, and the matching between them is performed by the platform through a centralized mechanism. In this new setting, drivers can be appointed to pick up a passenger before they actually meet. Thus, cruising aims to increase the opportunity of getting a ride, instead of traditional passenger-seeking.

Idle driver cruising is essentially caused by the imbalance between supply (online drivers) and demand (passenger requests). For example, as shown in Figure 1, drivers who have just sent a passenger from home to office in the morning have to travel back to residential zones for better chances of finding the next passenger. Various efforts have been made to alleviate such imbalance, including advanced matching mechanisms [20, 22, 31, 35], pricing schemes [15, 26] and market regulation [36]. Among them, optimizing driver's cruising behavior, with its unique characteristics of flexibility, efficiency, and the ability to improve driver's experience, acts as an important strategic component in e-hailing platforms.

In this paper, we study the problem of *driver repositioning*, where driver's idle cruising is interposed by the platform through certain kind of interactions. With rich real-time information from both supply and demand side available, the goal of driver repositioning is to improve individual driver's experience, as well as the whole platform's efficiency. Driver repositioning in large-scale e-hailing platforms has the following unique opportunities and challenges:

1) **Driver experience satisfaction**. Unlike the idealized quasi-self-driving-vehicle setting where drivers unconditionally comply with the platform's instructions, in practice the top priority for all

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380287

kinds of driver interactions is to provide good driver experience. In particular, considering that it is the platform who performs matching between drivers and passengers in e-hailing platforms, drivers may doubt and lose confidence in the platform when being idle for a long while. The situation could be even worse if a repositioning advice, also sent by the platform, ends up with a failure of getting rides. Therefore, it is crucial for a repositioning algorithm to provide consistent user experience and build long-term confidence of drivers.

2) **Multi-driver collaboration**. A prominent advantage of the centralized repositioning system over driver's self-movement lies in the ability of effective decision-making in a higher dimension. Thanks to the rich information brought by advanced Global Positioning Systems (GPS) and powerful cloud computing infrastructures, it is possible to capture real-time information for each driver and passenger, including their location and status, and further to obtain the supply and demand distribution in real time. This enables effective collaboration between multiple drivers and also avoids potential conflicts between multiple repositioning instructions.

To address these challenges and design a system that can be successfully adopted in real-world e-hailing platforms, we propose a novel framework of driver repositioning that optimizes driver experience and platform efficiency simultaneously. Firstly, a new driver interaction design called "repositioning task" is proposed to provide robust driver experience satisfaction through failure-rate optimization and subsidy compensation. A novel algorithm is then developed to generate repositioning tasks in real-time. Inspired by algorithms in recommender systems [5, 7, 9], we perform candidate generation for possible repositioning tasks, removing ineffective ones, and value the remaining tasks using a carefully designed scoring function. Utilizing real-time supply and demand information, the scoring function together with the capacity of each spatial-temporal state is computed in a bilateral formulation. Repositioning tasks for multiple drivers are then collaborated using a network flow algorithm widely used in fleet management solutions [2, 16, 29]. After extensive experiments on real-world systems, the proposed algorithm shows a significant improvement in driver income and platform efficiency, and has been successfully deployed in online production systems.

The contributions of this work are summarized as follows: 1) We formally define the driver repositioning problem in e-hailing platforms, and propose, to our knowledge, the first industry-level solution for driver repositioning. 2) We design a new form of driver interaction called "repositioning task" that provides robust driver experience. 3) A novel modularized framework is proposed for driver repositioning, which meets various requirements in practical situations, including robust driver experience satisfaction, multi-driver collaboration, and multi-objective optimization. 4) We conduct experiments in real world and reports a 2% improvement of driver's income. The proposed method has been deployed in the online production system, serving millions of drivers everyday.

The remainder of the paper is structured as follows. Section 2 reviews existing ideas related to taxi route recommendation and fleet management. We define important concepts and describe the form of driver repositioning tasks in Section 3. Section 4 introduces the proposed framework. We evaluate the performance of the proposed approach in Section 5 and close this paper with some conclusive remarks in Section 6.

2 RELATED WORK

In this section, we review related research from two sub-areas: taxi recommendation and fleet management.

2.1 Taxi Recommendation

Research on taxi recommendation emerged after GPS was widely equipped on taxis. Records of GPS with location, time and fare status enable the analysis and mining of the taxi trajectories. Since then many researchers tried to reveal the hidden pattern buried in the trajectories and summarize better cruise routes for idle taxis. These techniques could be grouped into two categories: spot recommendation and route recommendation:

Spot recommendation. A hotspot is a specific position with a high likelihood of finding a customer. Recommending one or multiple pick-up spots at the immediate next step has been widely discussed. Ge et al. [14] developed a recommender system to offer taxi drivers a sequence of pick-up points and potential parking positions. Powell et al. [27] presented a method, so-called Spatio-Temporal Profitability (STP) map, to provide vacant taxicabs with the next pickup and profitable locations. By considering distance, waiting time and expected fare, Hwang et al. [17] also proposed a taxi recommender system to determine the next cruising location. Zhang et al. [40] established a demand hotspots prediction framework to generate recommendation for taxi drivers.

Route recommendation. Rather than a single spot, cruising routes comprised of a sequence of pick-up locations may be more practical and effective for drivers. Within recent years, Markov Decision Process (MDP) and Reinforcement Learning (RL) have been becoming increasingly popular methods to solve these route recommendation problems. Rong et al. [28] employed the Grid-based MDP to the passenger seeking process for maximizing the profit in an extended time window. Based on [28] , Zhou et al. [42] further generalized the MDP approach to spatial networks and proposed statistical models to estimate time-variant parameters for the MDP model. Garg et al. [13] modeled taxi route recommendations as a Monte Carlo Tree Search problem whose objective is to minimize the distance to the next anticipated customer. Shou et al. [30] also considered the competitions among taxis and incorporated e-hailing drivers' new characteristic features to formulate MDP models. In order to optimize taxi driving strategies for cabdrivers based on reinforcement learning, Gao et al. [12] utilized the Q-learning algorithm and Verma et al. [32] employed the Monte Carlo RL method. Different from previous studies, Gao et al. [38] introduced temporal Poisson arrivals of passengers and spatial Poisson distributions of competing vacant taxis to calculate state transition probabilities into the MDP models.

It is worth noting that in e-hailing platforms, not only the GPS trajectories of taxis are available, but also the unsatisfied requests from the passengers can be recorded. Therefore, the distribution of both the supply side and the demand side can be captured, which is a significant information advantage. Route recommendation for e-hailing drivers is more practical and more potential in agility and comprehensiveness. In our study, we will exploit the global demand

distribution and take the mutual effect among multiple taxis into account by introducing methods in fleet management.

2.2 Fleet Management

Fleet management is a broad concept that incorporates decisions about fleet sizing and configuration, fleet allocation, vehicle routing, applied to a variety of different transport modes such as motor vehicles, bicycles, rail cars, aircraft and ships [3].

The majority of fleet management problems concentrated on the optimal design of routes to be used by a fleet of vehicles to serve a set of customers. After it was first proposed by Dantzig and Ramser [8], hundreds of studies were devoted to the exact or approximate solution of the many variants of this problem, such as the Vehicle route problem (VRP), Pick up and delivery problems with time windows (PDTW) and Dial-a-ride problems (DARP) [2, 16, 29]. Within recent years, dynamic and real-time routing problems have been brought into the focus of scholars in transportation research. A common framework "rolling horizon approach", in which plans are made by using all known information within a planning horizon while decisions are not finalized until necessitated by a deadline, was introduced to the process of continually assigning orders to idle drivers on the fly [23, 24, 39].

There are also some researches focusing their attention on the fleet redistribution and re-balance. Given the rapid development of autonomous vehicles (AV), Bike-sharing (BS) and centralized decision making e-hailing, redistributing available mobility has been playing critical roles in reducing the cost and promoting efficiency from perspective of operating platforms. Babicheva et al. [1] discussed empty vehicle redistribution algorithms for autonomous taxi services to minimize the passengers' waiting time. Also, Legros et al. [19] developed an implementable decision-support system which enable bike-sharing operators allocate and prioritize resource among bike-stations. To tackle the large-scale fleet management problem in e-hailing platforms, a contextual multi-agent reinforcement learning framework was proposed in [21]. The two concrete algorithms, contextual multi-agent actor-critic and contextual deep Q-learning, enabled the framework to achieve efficient explicit coordination and also to adapt to dynamically changing action spaces. By building a multi-agent reinforcement learning framework, Jin et al. [18] proposed a model CoRide to handle the tasks of joint order dispatching and fleet management of e-hailing platforms.

3 REPOSITIONING TASK

Upon centralized decision making system, the e-hailing app will establish a direct channel between the drivers and the platform. Under this circumstance, e-hailing drivers can upload their real-time positions to the platform, and platform will then dispatch appropriate orders to them. When there are not adequate orders nearby, the information channel is still available for idle drivers and can be further exploited to guide drivers to find the next passenger as soon as possible. In reality, the majority of e-hailing idle drivers usually depend only on their personal experience to reposition themselves, which is too subjective and partial to make an optimal decision. Therefore, in this research, rather than implementing offline recommendations in simulation environments, we utilize the ability of real-time communication between drivers and platform

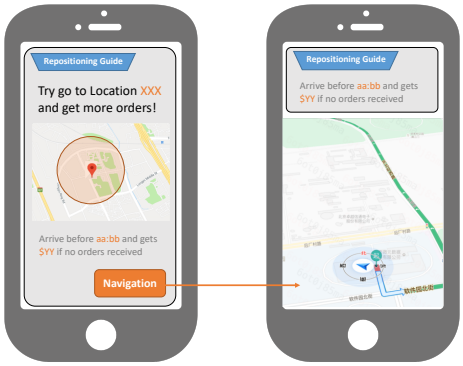


Figure 2: Interaction design for a *repositioning task*.

to send immediate repositioning tasks to drivers. It's obvious that these real-time repositioning tasks will be beneficial for idle drivers to make decisions.

In our settings, a repositioning task is triggered as the form of sending messages when the drivers have been idle for a period exceeding a certain duration threshold. As depicted in Figure 2, a guide message pops up on the driver's app after repositioning task has been triggered. By clicking the "Navigation" button, the drivers are regarded as accepting the repositioning task, after that they will be guided to the repositioning destination instantly. While on the way driving to the destination, the driver will be always available to pick up the upcoming orders on the routes. The purpose of this setting is to fully utilize the supply available in the routes, as well as to maximize the probability to find a new order during the process of repositioning task.

To sum up, there are three procedures in a repositioning task:

- **(1) Sending an explicit destination**: This implies the platform will suggest the idle driver a specific destination, which indicates a new position with a higher likelihood of finding a customer. According to the driver's choice (accepting or refusing), the platform will generate a recommended route using techniques including route planning and estimated time of arrival [33, 34], if the "repositioning task" has been accepted. Otherwise, the driver will cruise on the street without any external assistance.
- **(2) Judging criterion of repositioning success**: There are four possible outcomes (Figure 3) after a driver receives a repositioning task. We summarize them as the ending state s of the task.
  - $s^{(1)}$: A driver does not follow the repositioning task and drives in the opposite direction, or fails to arrive the destination before the scheduled time. The system will recognize the behavior and terminate this repositioning task automatically.
  - $s^{(2)}$: A driver accepts the task, heads for the destination, and then is dispatched to an order on the way. We regard this case as a successful recommendation as well.
  - $s^{(3)}$: A driver accepts the task and arrives at the destination, and then receives an order within a fixed time period

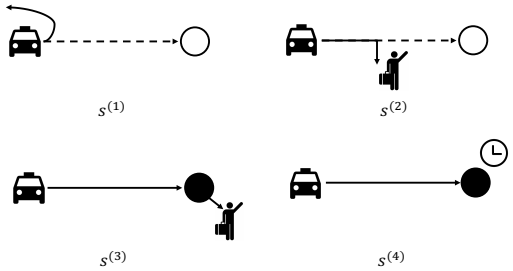


Figure 3: Ending states of a repositioning task. $s^{(4)}$ is regarded as a failure case.

(say 10 minutes). This is also regarded as a successful recommendation.
  - $s^{(4)}$: A driver accepts the task and arrives the destination, but ends up with no orders dispatched during the fixed time period. This is regarded as a kind of *failure* and worth special attention.
- **(3) Providing a compensation in the case of failure.** As discussed before, a failed repositioning task ($s^{(4)}$) will lead to extremely unsatisfying user experience and even cause drivers losing confidence in the platform. In practice, we introduce a monetary compensation to drivers in the case of task failure. This is a crucial setting for building relationships and trust between drivers and the platform.

4 PROPOSED FRAMEWORK

Having stated the problem, we present the proposed recommendation framework in this section. As shown in Figure 4, the proposed framework consists of three phases. Inspired by e-commerce recommendation systems [5, 7, 9], the first two phases aim to generate candidate tasks and conduct scoring for each one of them. A programming algorithm is used in the third phase to compute the final task assignments considering collaboration of multiple drivers.

Complete notations which will be used in the subsequent analysis are listed in Table 1.

Table 1: Notations.

Variable	Explanation
g	the symbol of a hexagonal grid
t	the symbol of a time interval
d	the symbol of a driver
o	the symbol of an order
r	the symbol of failure compensation
rep	a repositioning task $rep = \langle d, g, t, r \rangle$, with a driver, a destination, an expiring time and a subsidy
s	the ending states after the driver receives a repositioning task: $s \in \{s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\}$.
D^*	the candidate driver set in repositioning task
Rep	the candidate repositioning tasks set
Rr	the response rate, which is the ratio of the responded orders quantity and the number of total orders
δ	the idle time threshold
ϵ	the distance threshold from driver's position to destination
ζ	the travel time threshold from driver's position to destination
$\widehat{F}$	the failure probability threshold
$\widehat{Rr}_{\langle g,t \rangle}$	the upper-bound response rate at $\langle g, t \rangle$
$\Delta_{\langle g,t \rangle}$	the required maximum drivers quantity of destination g at time interval t

4.1 Task Candidate Generation

A repositioning task consists of four elements: an eligible driver, a destination, an expiring time and a subsidy associated to it.

Driver candidate generation. Intuitively, drivers who have been waiting for passengers for a long time need more assistance in passenger-seeking, and thus are more willing to accept repositioning tasks. In practice, we select drivers being idle for minutes longer than a threshold δ, as the initial candidates D^*.

Destination generation. For each driver d_i selected as a candidate for repositioning, a list of possible destinations is then generated. Since there are an arbitrary number of possible destinations available, we quantify a city into hexagon grids g, which is a commonly used spatial splitting method in e-hailing platforms, pioneered by Uber [4]. Within each grid, we further select several point of interests (POIs) with the highest number of passenger requests in historical data as potential repositioning destinations.

On the grid level, we explored a variety of grid candidates generation methods summarized as follows:

- **Geographic neighborhood**. The simplest idea of picking candidate grids is to select grids geographically adjacent to the driver's current location. We heuristically select grids inside the 3rd-order neighborhood as candidates to ensure a reasonable repositioning distance.
- **Trajectory mining**. As a supplement to neighborhood grids, we also extract grid pairs which frequently appeared in driver trajectories to augment the candidate grids list. This is inspired by the fact that geographically distant locations may still be well connected by the road network (e.g., a highway).

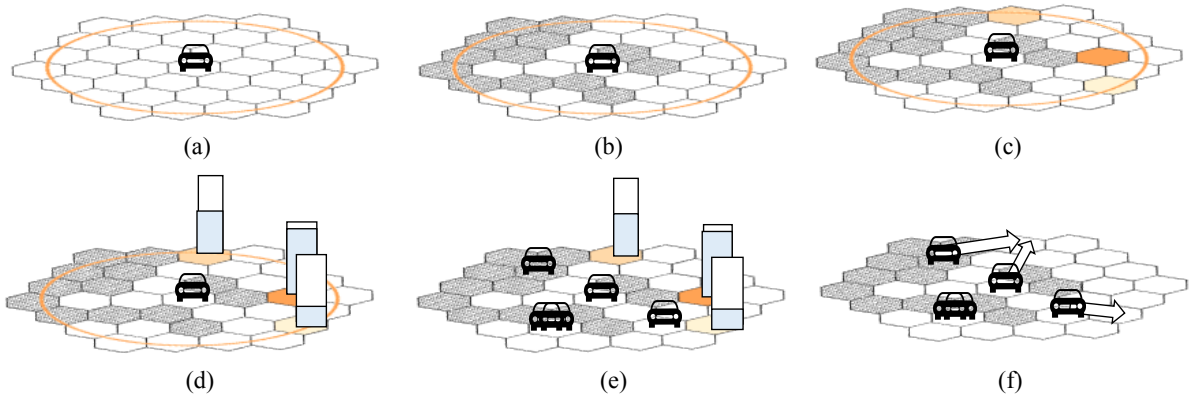


Figure 4: Flowchart of the proposed framework. In the candidate generation phase, the system will first retrieve candidate destinations for repositioning (a) and filter out some of them based on failure probability (b). In the following scoring phase, a spatial-temporal value function is exploited to compute the value and capacity for each state (c, d). The collaboration of multiple drivers is achieved in the programming phase (e). Finally, only one repositioning task is selected and sent to each driver (f).

- **Global hotspots**. Finally, global hotspots where the most portion of passenger requests burst are included as the basic candidates.

Given the resultant candidate grid lists denoted as $G^*_{d_i}$, we further introduce a distance and time limit in road map. This constraint is requisite to eliminate extremely bad experience (e.g., a repositioning task over $10km$ tends to be declined by drivers) and also reduce the computational cost for the following phases. In particular, we employ the route planning module and advanced estimated time of arrival (ETA) algorithms [11] to compute the road distance and assign an expiring time for each repositioning task candidate. All repositioning tasks with route distance exceeding ϵ or ETA exceeding ζ are removed.

Failure rate control. The aforementioned driver and destination generation process ensures that with an appropriate driver and a reasonable route, candidate repositioning tasks have a good chance of being accepted and implemented. On the other hand, once a task is implemented by a real driver, it is the platform's duty to regulate the probability of failure cases ($s^{(4)}$ described in Section 3) into a reasonable range. We introduce a failure-probability prediction model to explicitly control the trade-off between more candidate tasks and the risk of increasing failure rates.

The failure prediction model is formally defined as:

$$F(rep) = P(s = s^{(4)}) \tag{1}$$

Reasons leading to a failure repositioning task are of many folds, for example the driver's competitiveness in dispatching against other drivers, the driver's status, the route itself and the real-time supply and demand distribution. We model the failure prediction problem as a typical binary classification task and summarize available features as follows.

- Driver related features: idle time, status, the driver's failure probability in previous tasks, etc.
- Supply-and-demand related features: the number of orders created and idle drivers in the last 5/10 minutes in the current grid and destination grid, the forecasting number of orders and available drivers in the next 30 minutes.
- Route related features: ETA (Estimated Time of Arrival), distance, the failure probability and acceptance rate of repositioning tasks of this particular grid-pair in historical data.
- Environment related features: workday, hour, weather, etc.

We employ XGBoost [6] to categorize the failure probability in each city. Repositioning tasks with failure probability more than a threshold $\widehat{F}$ are discarded.

Failure compensation. In case when a repositioning task fails, we provide an experience guarantee for drivers by offering a compensation r for their fuel cost and time consuming. The compensation comes in the form of a subsidy, for which the amount is set as being comparable to the average order fare in each city.

After the aforementioned operations, a set of candidate repositioning tasks is generated and forwarded to the following scoring phase. Each task is denoted as $rep = \langle d, g, t, r \rangle$ which directs driver d to grid g, with an expiring time t and failure compensation r.

4.2 Task Scoring

The scoring module described here measures the advantage of each candidate repositioning task. Unlike the previous step where individual driver's experience is the primary consideration in generating task candidates, here we focus more on the platform as an aggregation of the will of all drivers. That is to say, the score evaluates how a repositioning task can benefit to the re-balance of real-time supply and demand, which leads to an increase of the whole system's efficiency.

We achieve this by computing the marginal increment of adding one driver in regard to the number of responded orders. Given a specific spatial-temporal state $\langle g, t \rangle$, we introduce a *piece-wise linear approximation* to describe how response rate Rr (the ratio

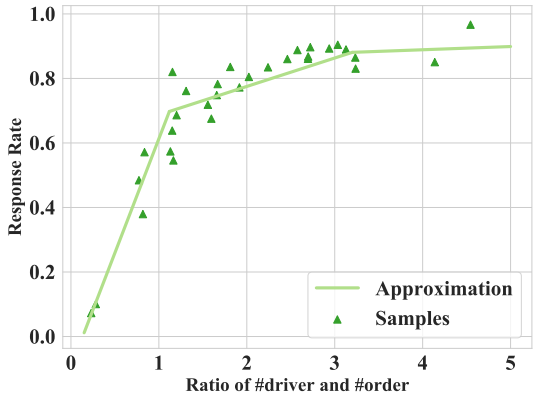


Figure 5: Example of the response-rate approximation function.

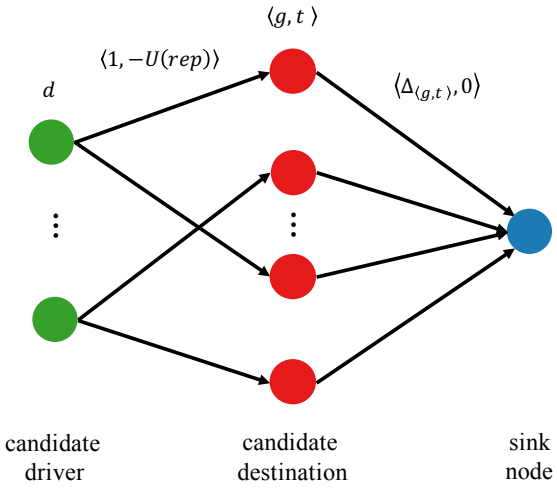


Figure 6: Graph of the converted Minimum Cost Flow problem. Labels on edges represent capacity and cost.

of responded orders over created orders) changes with the supply-demand ratio (the drivers quantity $|D|$ over orders quantity $|O|$):

$$Rr_{\langle g,t\rangle} = f_{\langle g,t\rangle}(\frac{|D|}{|O|}) \tag{2}$$

where $Rr_{\langle g,t\rangle}$ is the response rate at $\langle g,t\rangle$. Figure 5 shows an example of the approximation function $f_{\langle g,t\rangle}(\cdot)$. Intuitively, if the supply-demand ratio increases infinitely as more and more drivers entering into the platform, the function will approach 1, indicating that all passenger requests are fulfilled.

For each repositioning task $rep = \langle d, g, t, r\rangle$, we can then define its score $U(rep)$, named Spatial-temporal Marginal Utility, as below:

$$\begin{aligned} U(rep) &= \frac{\partial Rr_{\langle g,t\rangle}}{\partial |D|} \\ &= f_{\langle g,t\rangle}(\frac{|D|}{|O|})' \cdot \frac{1}{\overline{|O|}_{\langle g,t\rangle}}, \forall rep \in Rep \end{aligned} \tag{3}$$

where the number of orders denoted by $\overline{|O|}_{\langle g,t\rangle}$ at a future time interval t can be computed through forecasting [37].

Another interesting result extracted from the approximation function is the capacity of drivers in each spatial-temporal state. Unlike traditional recommendation engines [5] where the same item can be recommended unlimited times, in this work we must be aware that if all drivers are directed to a same place, the marginal value of adding a driver could be of much difference from the current state. Therefore, we compute the capacity by setting an upper-bound response rate $\widehat{Rr}_{\langle g,t\rangle}$ and restrict that the quantity of repositioning tasks guided to a grid should not result in a higher response rate than the upper bound. With the estimated order quantity $\overline{|O|}_{\langle g,t\rangle}$ and drivers quantity $|D|_{\langle g,t\rangle}$, (4) computes the required drivers quantity $\Delta_{\langle g,t\rangle}$ introduced into destination g. Subsequently, the required drivers quantity $\Delta_{\langle g,t\rangle}$ is further applied to the next phase for multi-driver collaboration.

$$\Delta_{\langle g,t\rangle} = \max\left(f_{\langle g,t\rangle}{}^{-1}(\widehat{Rr}_{\langle g,t\rangle} \cdot \overline{|O|}_{\langle g,t\rangle}) - |D|_{\langle g,t\rangle}, 0\right) \tag{4}$$

4.3 Programming

A recommendation problem may have various objectives, ranging from maximizing global reward to minimizing rejection rate. This objective can vary due to the nature of the agency who is managing the recommendation system. If the system operates for profit, the agency may pursue to maximize global reward because it is significantly related to the Gross Merchandise Volume (GMV). However, some systems may have a societal objective like maximizing the market share, which also coincides with minimizing rejection rate in the system. Therefore, based on candidate tasks Rep obtained by Section 4.1, the objective in the paper is to plan a set of tasks Rep^* with maximal global reward from the perspective of e-hailing platform, subject to the driver's experience constraints. The optimization problem can be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{rep \in Rep} \mathbb{I}(rep) \cdot U(rep) \\ s.t. \quad & \sum_{rep} \mathbb{I}(rep|_{d=d_i}) \le 1; \forall d_i \in D^* \\ & \sum_{rep} \mathbb{I}(rep|_{g=g_k, t=t_j}) \le \Delta_{\langle g_k,t_j\rangle}; \forall g_k, t_j \end{aligned} \tag{5}$$

where $\mathbb{I}(rep) \in \{0,1\}$ is the indicator variable of whether or not the task rep is included in the final task set Rep^*. Specifically, the first constraint ensures that each candidate driver d_i is assigned at most one repositioning task at a time. The second constraint guarantees the number of repositioning tasks directed to a grid g at one time interval t should be no more than its estimated capacity $\Delta_{\langle g,t\rangle}$.

Since the exact solution for (5) can cost a high computational time, in this work we further convert the optimization problem into a Minimum Cost Flow formulation [10]. As depicted in Figure 6, our framework consists of three layers, including candidate drivers, candidate destinations and a sink node. Each edge between two layers is associated with parameters $\langle\theta, w\rangle$, representing the capacity

and unit cost of an edge, respectively. These parameters are set as follows:

- The edges between the candidate driver layer and the candidate destination layer represent candidate repositioning task, with parameter tuple of a cost and a unit capacity, $\langle 1, -U(rep)\rangle$.
- The edges between the candidate destination layer and the sink node are flow-in control constraints, where each edge has its parameters of $\langle \Delta_{\langle g,t\rangle}, 0\rangle$.

A classical Minimum Cost Flow (MCF) solver [25] is used to solve the optimization problem in Figure 6. The resultant repositioning tasks Rep^* are then forwarded to drivers by the app.

5 EXPERIMENT

In this section, we will evaluate the proposed driver repositioning framework and present several insights from real-world experiments. Since we consider several practical issues in model design, such as earning trust from drivers to improve their willingness for accepting repositioning tasks, we skip an offline evaluation on simulators and directly evaluate the proposed framework through online experiments.

5.1 Experimental Setup

Hyper-parameters. There are some hyper-parameters used in this phase: the idle time threshold $\delta = 10min$, the distance threshold $\epsilon = 10km$, the travel time threshold $\zeta = 20min$, the failure-rate threshold $\widehat{F} = 0.6$, and the upper-bound response rate $\widehat{Rr}_{\langle g,t\rangle} = 0.95$. Experiments in all cities share a same set of hyper-parameters.

Evaluation metrics. We evaluate the proposed method mainly upon the following metrics.

- Call-in Per Order (CPO). Drivers who serve no orders for a long time tend to get anxious and try to call customer services for help. We record the number of call-ins per order (CPO) to measure the overall experience of drivers.
- Total Income and Income Per Hour (IPH). The total income and average IPH represents the efficiency of drivers on the platform as a whole.
- Gross Merchandise Volume (GMV). GMV records the total fee of all orders served by drivers on the platform. Improving driver's serving time and improving the service efficiency can both contribute to the boost of GMV.

Observation metrics. We monitor how each step in the repositioning task flow works out for reference.

- Average Task Number. It refers to the number of repositioning tasks that a driver receives on average in one day.
- Acceptance Rate. It is the ratio of the number of accepted repositioning tasks (ending with $s^{(2)}$, $s^{(3)}$ and $s^{(4)}$) over the total number of repositioning tasks.
- Failure Rate. It is the ratio of the number of failure (ending with $s^{(4)}$) over the number of accepted repositioning tasks (ending with $s^{(2)}$, $s^{(3)}$ and $s^{(4)}$).

A/B testing design. To evaluate the performance from different aspects, we adopted two customized A/B testing designs.

- **Design A**: a random split of traffic according to driver's ID into a balanced experimental group and control group. This setting is used to evaluate metrics from the driver's perspective, including acceptance rate, failure rate and CPO.
- **Design B**: a random split according to time slices (2 hours in our experiment). This design is specialized for experiments with network effects where drivers in group A can compete with ones in group B on limited resources. Traffic split A/B testing design is error-prone in this setting. We use the time-split design for evaluation metrics from the platform's view, such as GMV.

Table 2: Statistics of repositioning tasks.

City	Average Number	Acceptance Rate	Failure Rate
City A	2.03	71.80%	4.36%
City B	2.83	69.30%	2.99%
City C	2.54	77.26%	5.17%
City D	2.60	74.91%	5.90%
Average	2.50	73.32%	4.61%

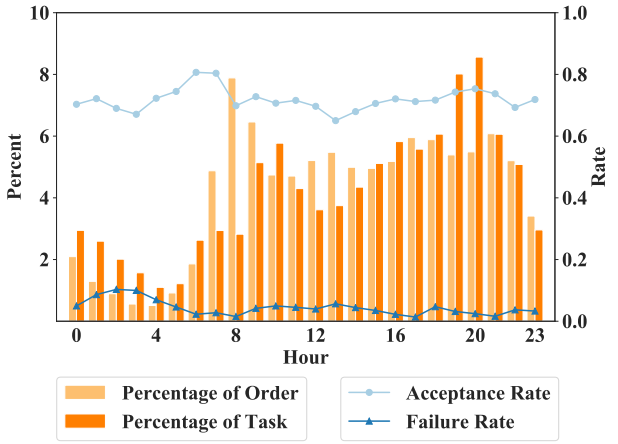


Figure 7: Hourly statistics on weekdays.

System deployment. To ensure the stability and reduce the computation time, our system was deployed on a cluster of 8 compute nodes. The computation time of one iteration ranges from 1 second to 20 seconds, which is related to the number of idle drivers and the size of the city.

5.2 Result Analysis

We performed multi-rounds of online A/B testings in several cities of different sizes. The proposed framework was compared with the drivers' self-motivated movements. Specifically, drivers in the experimental group received real-time repositioning tasks sent by the platform, while drivers in the control group relied on their own to select the cruising route. Note that repositioning tasks are *not mandatory*, i.e., drivers have their right to reject the repositioning suggestions and follow their own judgement.

We first employed A/B testing design A to study the driver's behavior and feedback for our repositioning tasks. Table 2 shows some major statistics of repositioning tasks. On average, a driver

Table 3: Quantitative results of the proposed framework.

City	Income	IPH	CPO
City A	+0.84%	+0.28%	-8.48%
City B	+0.90%	+0.63%	-10.30%
City C	+1.33%	+1.40%	-25.27%
City D	+2.01%	+1.17%	-7.186%

in the experimental group received 2.5 repositioning tasks per day. Nearly 75% repositioning tasks were accepted, with the failure rate below 5%. Figure 7 demonstrates a detailed analysis of the relationship between statistics of repositioning tasks and the number of passenger requests in a city among different hours. Several insights can be concluded here.

1) Morning peak, where the outburst of passenger requests makes nearly all the drivers busy serving, recorded the minimum amount of repositioning tasks generated. The number of repositioning tasks were steadily high in non-rush hours, especially 7pm and 8pm.

2) Acceptance rate was rather consistent over the daytime.

3) In peak hours the failure rate was extremely low, while other hours in the daytime also recorded a failure rate of less than 10%, which is very encouraging.

4) After midnight the driver experience of repositioning was lower, with a higher failure rate and lower acceptance rate. It is understandable since there are less riding requests and drivers operating at night, making it harder to find consistent places to go for passenger-seeking.

Comparison results against the control group is demonstrated in Table 3. Drivers in the experimental group delivered a significant improvement in their income ranging from 0.8% to 2% across different cities. Their IPH were also higher than those in the control group by 0.3% to 1.4%. We also noticed encouraging results on the experience side, for which the CPO statistics for the experimental group was 7% to 25% lower. To evaluate the improvement in respect of platform's GMV, we performed experiments in another 4 cities using A/B testing design B. Results proved that the improvement of driver experience and income also led to a more promising result in the platform's view, where the gains of the platform's GMV was approximately 2%. In summary, the proposed method showed a better performance on all evaluation metrics, indicating amelioration on both driver experience and platform efficiency.

5.3 Ablation Study

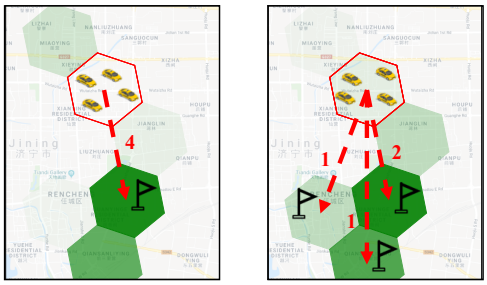
Here we conduct a set of ablation studies to evaluate the effectiveness of several key designs in the proposed framework.

5.3.1 Scoring: Marginal Utility VS. Local Hotspot. We first evaluated the design in the Scoring phase, where a marginal value function considering information from both the demand and supply side is employed to assess each repositioning task. The goal of this design is to optimize the income of all drivers as a total, instead of greedily selecting an optimal destination individually.

We compared the marginal-utility scoring function with a baseline local-hotspot strategy [38]. This method essentially suggests the driver to move into grids with the highest demand intensity,

Table 4: Comparison results in ablation experiments.

Strategy	GMV	Acceptance Rate	Failure Rate
Marginal utility VS. Local hotspot	+0.41%	+0.64pp	-0.05pp
MCF VS. Self-optimal	+0.12%	+1.34pp	-1.13pp
Subsidy VS. Non-subsidy	+1.20%	+7.43pp	+3.41pp



(a) (b)

Figure 8: Repositioning tasks generated by the self-optimal strategy (a) and MCF (b). Green grids are candidate destinations, whose shade is positively correlated with score. From the same source grid (red), self-optimal strategy directed all four drivers to the same destination, while MCF produces more diversed recommendations.

where each repositioning task is assigned by a score computed as the forecasting number of orders in the finishing state:

$$V(rep) = \overline{|O|}_{\langle g_k, t_{end} \rangle}, \forall rep \in Rep \tag{6}$$

Table 4 presents the performance difference between two strategies. Marginal utility achieved an out-performed consequence, including 0.41% improvement in GMV, 0.64% improvement in acceptance rate and 0.64% decrease in failure rate.

5.3.2 Programming: Minimum Cost Flow Method VS. Self-optimal. In the proposed framework, the Programming module is an important part to realize multi-driver collaboration. To demonstrate this advantage, we compared our results with a greedy self-optimal strategy through an ablation study. In particular, drivers in the control group of this experiment received repositioning tasks generated by a greedy method choosing the destination with a maximum score among task candidates individually.

Figure 8 intuitively demonstrates repositioning tasks generated for the two groups. It is clear that in the control group candidate drivers at same location were directed to same destination, leading to a redundancy of supply at the local hotspot. As a contrast, repositioning tasks generated by the experimental group had more diverse destinations and less conflicts.

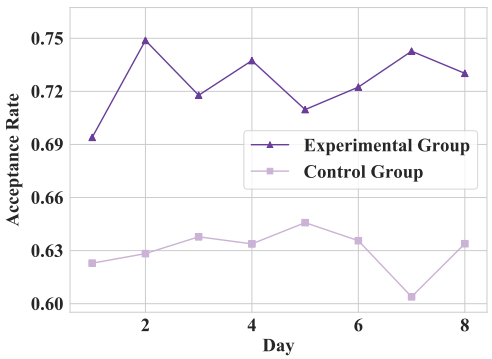


Figure 9: Comparison of acceptance rate with or without failure compensation.

As listed in Table 4, GMV of the experimental group was slightly higher for 0.12%, indicating the improvement of total income from the perspective of the platform. A more significant improvement comes from the experience side, where the acceptance rate of experimental group was higher than the control group for 1.34pp, while failure rate was 1.13pp lower.

5.3.3 Effectiveness of Failure Compensation. Failure compensation provides a guarantee of minimum experience for driver repositioning and is supposed to benefit to the acceptance rate. We conducted an A/B testing in Design A to prove this assumption. Here drivers in the experimental group received repositioning tasks with a conditional subsidy when a task ends up with failure, while drivers in the control group received repositioning tasks with no failure compensation.

Results show that experimental group were superior to the control group on both platform efficiency and driver experience. Specifically, GMV of the experimental group were 1.2% higher, while CPO was 11.34% lower. An interesting phenomenon occurred that failure compensation in fact led to a higher failure rate. We argue that it was caused by the fact that with failure compensation, driver's will of accepting repositioning tasks was higher (acceptance rate of the experimental group is 8pp higher on average). As a result, more drivers arrived at the destination and increased the probability of receiving no orders after arrival. Figure 9 shows daily acceptance rate of the two groups. It can be concluded that drivers' compliance and willingness was significantly improved.

5.4 User Study

In addition to quantitative experiments, we are also curious about the subjective feedback from real drivers. A short online questionnaire was sent in the app to all drivers who had experienced repositioning tasks. The questionnaire contains five following questions regarding the experience about repositioning tasks.

- **Question 1**. Have you accepted any repositioning task?
- **Question 2**. If you receive a repositioning task again in the future, will you accept it?
- **Question 3**. Open question: for an unaccepted repositioning task, what is the main reason?
- **Question 4**. Open question: what do you think is the main problem of current repositioning tasks?
- **Question 5**. Rating scale question: how likely is it that you would recommend other drivers to accept a repositioning task?

2.5% of the investigated drivers responded to the questionnaire. In general, 94.6% respondents said that they had accepted a repositioning task, and 64.6% said they would continue to accept the repositioning task if they received it again.

Question 5 is a typical question whose answer is used to calculate a key experience metric, **Net Promoter Score (NPS)**. The results of this question show that there are 55.9% promoters and 28.9% detractors, leading to a NPS of $55.9\% - 28.9\% = 27.0\%$. This indicates a quite positive overall reaction on repositioning tasks.

The user study also draws some possible suggestions for further improving driver repositioning. For example, on Question 3, 59.8% respondents said the reason for not accepting the repositioning task is that the distance is too long, and 53.4% said the expected income of the next order being too low after reaching the repositioning destination is a key concern. Meanwhile, 42.9% mentioned that the traffic congestion on the way is another concern. On Question 4, 46.2% suggested the convenience of parking at the destination of the repositioning task should be taken into consideration. These discoveries indicate the importance of driver's expectation management, and the need of investigating new forms of subsidizing associated to repositioning tasks.

6 CONCLUSION & FUTURE WORKS

In this paper, we proposed a novel framework for driver repositioning which provides effective guidance for drivers seeking the next passenger on e-hailing platforms. The proposed method is carefully designed for practical industrial usage, including a user-friendly driver interaction design for improving driver experience, and the implementation of multi-driver collaboration that re-balances supply and demand and increases social welfare. Through extensive online A/B testings, the proposed method delivered a remarkable improvement of both driver income and user experience related metrics. It has been fully deployed in DiDi Chuxing, a real-world e-hailing platform, serving millions of drivers in a daily basis.

Various extensions of the proposed framework will be explored in the future. These include investigating on frequency control methods in task candidate generation, soft capacity in programming, different subsidizing forms and more refined subsidizing algorithms, etc. Another possible extension lies in adopting reinforcement learning to provide an end-to-end computation of candidate generation, scoring and subsidizing. To be open-minded, using road segments and optimizing route directly is also a promising kind of framework for driver repositioning. We are actively pursuing these methods.

REFERENCES

[1] Tatiana Babicheva, Wilco Burghout, Ingmar Andreasson, and Nadege Faul. 2018. The matching problem of empty vehicle redistribution in autonomous taxi systems. *Procedia computer science* 130 (2018), 119–125.

[2] Roberto Baldacci, Vittorio Maniezzo, and Aristide Mingozzi. 2004. An exact method for the car pooling problem based on lagrangean column generation. *Operations Research* 52, 3 (2004), 422–439.

[3] Maurizio Bielli, Alessandro Bielli, and Riccardo Rossi. 2011. Trends in models and algorithms for fleet management. *Procedia-Social and Behavioral Sciences* 20 (2011), 4–18.

[4] I Brodsky. 2018. H3: Uber's Hexagonal Hierarchical Spatial Index.

[5] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-commerce Recommendation in Alibaba. *CoRR* abs/1905.06874 (2019). arXiv:1905.06874 http://arxiv.org/abs/1905.06874

[6] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA.

[8] George B Dantzig and John H Ramser. 1959. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.

[9] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube Video Recommendation System. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 293–296. https://doi.org/10.1145/1864708.1864770

[10] Jack Edmonds and Richard M Karp. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)* 19, 2 (1972), 248–264.

[11] Ruipeng Gao, Xiaoyu Guo, Fuyong Sun, Lin Dai, Jiayan Zhu, Chenxi Hu, and Haibo Li. 2019. Aggressive driving saves more time? multi-task learning for customized travel time estimation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19). AAAI Press*. 1689–1696.

[12] Yong Gao, Dan Jiang, and Yan Xu. 2018. Optimize taxi driving strategies based on reinforcement learning. *International Journal of Geographical Information Science* 32, 8 (2018), 1677–1696.

[13] Nandani Garg and Sayan Ranu. 2018. Route recommendations for idle taxi drivers: Find me the shortest route to a customer!. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1425–1434.

[14] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. 2010. An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 899–908.

[15] Fang He, Xiaolei Wang, Xi Lin, and Xindi Tang. 2018. Pricing and penalty/compensation strategies of a taxi-hailing platform. *Transportation Research Part C: Emerging Technologies* 86 (2018), 263–279.

[16] Liwen Hou, Dong Li, and Dali Zhang. 2018. Ride-matching and routing optimisation: Models and a large neighbourhood search heuristic. *Transportation Research Part E: Logistics and Transportation Review* 118 (2018), 143–162.

[17] Ren-Hung Hwang, Yu-Ling Hsueh, and Yu-Ting Chen. 2015. An effective taxi recommender system based on a spatio-temporal factor analysis model. *Information Sciences* 314 (2015), 28–40.

[18] Jiarui Jin, Ming Zhou, Weinan Zhang, Minne Li, Zilong Guo, Zhiwei Qin, Yan Jiao, Xiaocheng Tang, Chenxi Wang, Jun Wang, et al. 2019. CoRide: Joint Order Dispatching and Fleet Management for Multi-Scale Ride-Hailing Platforms. *arXiv preprint arXiv:1905.11353* (2019).

[19] Benjamin Legros. 2019. Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. *European Journal of Operational Research* 272, 2 (2019), 740–753.

[20] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. 2019. Efficient Ridesharing Order Dispatching with Mean Field Multi-Agent Reinforcement Learning. In *The World Wide Web Conference*. ACM, 983–994.

[21] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1774–1783.

[22] Yifang Liu, Will Skinner, and Chongyuan Xiang. 2019. Globally-Optimized Realtime Supply-Demand Matching in On-Demand Ridesharing. In *The World Wide Web Conference*. ACM, 3034–3040.

[23] Meghna Lowalekar, Pradeep Varakantham, and Patrick Jaillet. 2018. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence* 261 (2018), 71–112.

[24] Jiaqi Ma, Xiaopeng Li, Fang Zhou, and Wei Hao. 2017. Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. *Transportation Research Part C: Emerging Technologies* 84 (2017), 124–141.

[25] Andrew Makhorin. 2012. GNU Linear Programming Kit. https://www.gnu.org/software/glpk

[26] Robert L Phillips, Michael S Gordon, Ozgur Ozluk, Stefano Alberti, Robert A Flint, Jorgen K Andersson, Keshava P Rangarajan, Tom Grossman, Raymond Mark Cooke, Jeremy S Cohen, et al. 2006. Dynamic pricing system. US Patent 7,133,848.

[27] Jason W Powell, Yan Huang, Favyen Bastani, and Minhe Ji. 2011. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *International Symposium on Spatial and Temporal Databases*. Springer, 242–260.

[28] Huigui Rong, Xun Zhou, Chang Yang, Zubair Shafiq, and Alex Liu. 2016. The rich and the poor: A Markov decision process approach to optimizing taxi driver revenue efficiency. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2329–2334.

[29] Stefan Ropke and Jean-François Cordeau. 2009. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science* 43, 3 (2009), 267–286.

[30] Zhenyu Shou, Xuan Di, Jieping Ye, Hongtu Zhu, and Robert Hampshire. 2019. Where to Find Next Passengers on E-hailing Platforms?-A Markov Decision Process Approach. *arXiv preprint arXiv:1905.09906* (2019).

[31] Xiaocheng Tang, Zhiwei Tony Qin, Fan Zhang, Zhaodong Wang, Zhe Xu, Yintai Ma, Hongtu Zhu, and Jieping Ye. 2019. A Deep Value-network Based Approach for Multi-Driver Order Dispatching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1780–1790.

[32] Tanvi Verma, Pradeep Varakantham, Sarit Kraus, and Hoong Chuin Lau. 2017. Augmenting decisions of taxi drivers through reinforcement learning for improving revenues. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

[33] Zheng Wang, Kun Fu, and Jieping Ye. 2018. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 858–866.

[34] Zheng Wang, WANG Ziteng, and Xiaowei Zhong. 2019. Systems and methods for route planning. US Patent App. 16/212,633.

[35] Zhe Xu, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, and Jieping Ye. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 905–913.

[36] Zhengtian Xu, Yafeng Yin, and Jieping Ye. 2019. On the supply curve of ride-hailing systems. *Transportation Research Part B: Methodological* (2019).

[37] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[38] Xinlian Yu, Song Gao, Xianbiao Hu, and Hyoshin Park. 2019. A Markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological* 121 (2019), 114–134.

[39] Xianyuan Zhan, Xinwu Qian, and Satish V Ukkusuri. 2016. A graph-based approach to measuring the efficiency of an urban taxi service system. *IEEE Transactions on Intelligent Transportation Systems* 17, 9 (2016), 2479–2489.

[40] Kai Zhang, Zhiyong Feng, Shizhan Chen, Keman Huang, and Guiling Wang. 2016. A framework for passengers demand prediction and recommendation. In *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 340–347.

[41] Yu Zheng, Jing Yuan, Wenlei Xie, Xing Xie, and Guangzhong Sun. 2010. Drive smartly as a taxi driver. In *2010 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing*. IEEE, 484–486.

[42] Xun Zhou, Huigui Rong, Chang Yang, Qun Zhang, Amin Vahedian Khezerlou, Hui Zheng, M Zubair Shafiq, and Alex X Liu. 2018. Optimizing Taxi Driver Profit Efficiency: A Spatial Network-based Markov Decision Process Approach. *IEEE Transactions on Big Data* (2018).