This article has been accepted for publication in IEEE Transactions on Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TITS.2022.3219381

A Learning Method for Real-time Repositioning in E-hailing Services

Yue Yang, Mohsen Ramezani*

***Abstract*—Internet-based e-hailing services have become a major component of urban transportation systems in recent years. The spatio-temporal mismatch between supply (available vehicles) and demand (passenger requests) deteriorates e-hailing platforms' performance. Hence, repositioning available vehicles can be productive. In this paper, we propose a real-time repositioning method in ride-sourcing systems that considers both the responsiveness to immediate demand and the long-term (i.e., several hours) operational efficiency simultaneously. The proposed approach integrates the solutions of two procedures: i) a single-agent Markov Decision Process (MDP) model to evaluate the long-term influence of the repositioning on platform efficiency and ii) a binary linear program (BLP) to tackle the multi-driver repositioning problem in real-time taking into account the elapsed time of each not-responded order. Numerical experiments using real-world demand data with impatient passengers and contractors (i.e., drivers) demonstrate that the proposed method outperforms several repositioning benchmarks with regard to platform efficiency, e.g., reducing order cancellations, passengers' experience, e.g. reducing waiting times, and drivers' gains, e.g., increasing occupied rates.**

***Index Terms*—Mobility On-Demand, Fleet Management, Ride-hailing, Relocation, Transportation Network Company (TNC).**

I. Introduction

With the development of GPS-enabled technologies and the proliferation of smartphones, traditional taxi industries have witnessed radical changes. The emergence of mobile-based E-hailing services, such as Uber, Lyft, and Didi Chuxing, enabled taxi drivers (or self-scheduled contractors) to be systemically matched with passengers with no need for random cruising on the streets. It is reported that Uber served 5.22 billion trips worldwide in 2017, up from 140 million trips in 2014 [1]. DiDi also provided service for over 25 million trips each day in 2018 in 400 cities in China [2]. The scale and richness of digital data collected by these platforms offer unprecedented opportunities for various qualitative and extensive analysis including supply management [3], hot-spot identification [4], spatio-temporal demand estimation [5], autonomous taxis [6], labour incentives [7], dynamic pricing [8], and public safety [9].

E-hailing platforms continuously receive passengers' trip requests, geographical coordinates, and occupancy status of E-hailing vehicles (driven by contractors), and periodically dispatch idle vehicles to serve unassigned orders. Also, platforms might reposition the vehicles that fail to receive a pick-up order to a different location for the prospect of less cruising time in the future. Assuming dispatching decisions are made by using all known information, the operation of E-hailing platforms can be summarized into four steps:

1) **Collecting:** The platform collects the information of vacant vehicles, new orders, and unserved orders in each planning horizon.
2) **Dispatching:** The platform makes dispatching decisions by a centralized optimization approach at each decision point.
3) **Announcing:** The platform notifies passengers and the vehicles when they are matched. Afterwards, the vehicles will follow the platform's routing guide to serve the passengers.
4) **Repositioning:** Gathering the occupancy information of vehicles and the service information of orders, the platform redistributes vacant vehicles to some locations for upcoming or unserved orders.

From the perspective of vehicles, the E-hailing services can be divided into three stages, as illustrated in Figure 1. (1) **Searching**: Vacant vehicles cruise or park on the street until they have been assigned to a passenger request. The position they receive the dispatched order is called the dispatching point. (2) **Picking up**: Assigned vehicles from their current dispatching point head for the origin of the order to pick up the passenger. The time from being assigned to pick up the passenger is called deadheading time. (3) **Serving**: Occupied vehicles take the passenger to the destination, and then they will start searching again. Evidently, the efficiencies of these three stages primarily affect the service quality for passengers, vehicles, and the platform, including waiting times, individual income, occupied rate, market share, service profit, social welfare, and other externalities. Therefore, E-hailing platforms employ a large number of operational strategies to improve the efficiencies of these three stages.

Aiming to minimize pick-up times or maximize system profit, a plethora of order dispatching algorithms have been investigated. Generally, these dispatching techniques naturally fall into the category of ride-matching problems [10], [11], [12], [13], [14], [15], [16], [17], [2], [18], [19], [20], [21]. Nonetheless, once vehicles are not matched in the dispatching procedure, E-hailing systems face a critical question (or an opportunity) in the Searching stage; where is the best location for vacant vehicles to find a passenger at subsequent times? In the traditional taxi service, taxi drivers usually rely on their personal experience to find the next passenger, which is myopic and selfishly near-optimal. Providing a comparison with current taxi operation, Santi et al. [22] demonstrate cumulative trip length can be cut by 40% or more by the

The authors are with the University of Sydney, school of Civil Engineering, Australia. (e-mail: mohsen.ramezani@sydney.edu.au).

This article has been accepted for publication in IEEE Transactions on Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TITS.2022.3219381

implementation of repositioning and fleet management, Vazifeh et al. [23] also show that effective repositioning methods can allow a 30% reduction in fleet size. Considering the time-varying effect of congestion, Ramezani et al. [24] model and control repositioning in large-scale networks taking into account the impact of network congestion and demonstrate an improvement in the taxi service performance (reducing travellers' waiting times by 20%).

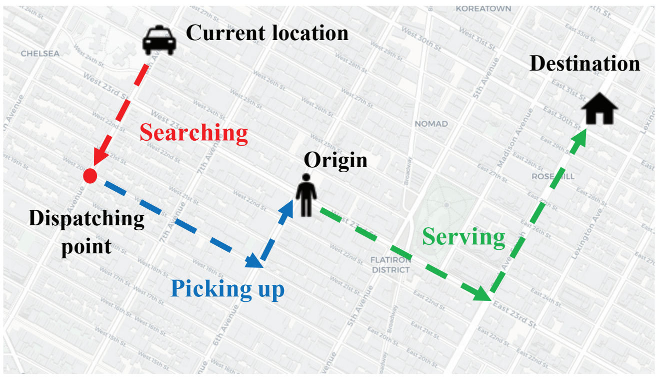


Fig. 1. Three stages in E-hailing services.

The majority of literature, e.g., [25], [26], [27], [28], [29] redistribute vacant vehicles from their current position to one or multiple pick-up locations with a high likelihood of being matched with a passenger at the immediate next step. A major drawback of these methods is overlooking the overall efficiency in the long-term (i.e., several hours). To tackle this issue, Markov Decision Process (MDP) has been adopted in passenger-seeking strategies in traditional taxi industries and repositioning strategies in E-hailing services. Considering the road segment and period of a vacant taxi, Zhou et al. [30] propose a network-based MDP model to recommend the next cruising direction for taxi drivers. The MDP-based model assumes the agent (single driver) knows how the environment shifts the state and the feedback rewards, and then finds the optimal policy based on the model to achieve the maximum cumulative reward. To estimate state transition probabilities of the MDP, Yu et al. [31] assume temporal Poisson arrivals of passengers and spatial Poisson distributions of vacant taxis in the network. The dynamic programming algorithm is introduced to solve the problem. In traditional taxi services, an idle taxi's searching process ends only when the driver sees a passenger, and the passenger accepts the ride. Shou et al. [32] develop an MDP to model e-hailing drivers' sequential decision-making in searching for the next passenger. However, all the above literature assumes each driver is an independent agent and ignores the impact of competition on each driver's policy. To capture competition among multiple drivers, a multi-agent optimization method is required for repositioning in e-hailing systems.

Different from the aforementioned model-based reinforcement learning (RL) studies, model-free reinforcement learning approaches are also studied in repositioning problems. For traditional taxi drivers, Verma et al. [33] develop a Monte Carlo learning recommendation system for advising drivers to find customers from the historical trajectory. Gao et al. [34] utilize the Q-learning algorithm to improve taxi operation efficiencies and increase the opportunities for passengers to find an available taxi. Considering the multi-agent interaction and E-hailing application, Lin et al. [35] propose a multi-agent deep reinforcement learning (MARL) model to design an effective fleet management strategy for e-haling platforms.

However, there are three key issues associated with Model-based and Model-free RL-based repositioning approaches: (i) an uncertain number of agents: the number of drivers (agents) changes based on time and their received income, which makes the e-hailing environment more challenging than traditional MARL environments. In the e-hailing system, the drivers can make working decisions (working at their preferred time shift and area) and have heterogeneous market-behavioural patterns (being full-time or part-time) based on their earning and preference [36], [37]. (ii) Curse of dimensionality: the joint action space considering hundreds of drivers and hundreds of repositioning destinations may cause scalability issues even using multi-agent deep learning frameworks [38], [39]. (iii) Service priorities: the e-hailing systems are highly time-varying and passenger-centric markets. Passengers are naturally impatient to receive a matching response and be picked up in a reasonable time. It is expected that once their patience is exhausted, they will cancel the trip order. Thus, service priorities among different trip requests and regions should be considered in real-time repositioning.

To tackle the above challenges, the integrated repositioning approach is developed to consider both the anticipated long-run efficiency and the real-time information (service priorities) in a centralized and coordinated way. By modelling the repositioning as a sequential decision-making problem, the single-agent MDP model is employed to generate the optimal policies and evaluate the long-term influence of repositioning policies on the platform efficiency using historical data. By accounting for the unmatched passengers' waiting times in real-time (the elapsed time from their order time), the real-time matching between multiple drivers and repositioning destinations is formulated as an optimization problem and solved using binary linear programming (BLP). In the final stage of the integrated repositioning, the priority is set to immediate requests. The vacant vehicles with a non-zero BLP solution are firstly repositioned to destinations with higher service priorities. Then, the idle vehicles with all-zero BLP solutions will follow the MDP solution to optimize the long-term operational efficiency of the platform.

The contributions of this paper are three-fold: (1) Based on the historical demand data and taxi trajectories, we design a single-agent MDP model to evaluate the anticipated long-run benefits of the repositioning. (2) To reposition multiple drivers in real-time, we formulate the problem as a binary linear programming (BLP) to minimize the total waiting times of unmatched trip orders. Afterwards, the solutions of the MDP and the BLP are integrated with different priorities in the final repositioning solution. (3) We examine the performance of the proposed approach in a detailed E-hailing simulator based on the Manhattan road network. The proposed method outperforms several repositioning benchmarks (such as Parking, Random Walk, MDP only, and real-time only methods) with regard to both platform efficiency and users' (passengers' and drivers') experience.

The remainder of the paper is structured as follows. Section II models the repositioning problem as a single-agent MDP and details the process of defining states, actions, and state transitions and extracting MDP parameters from historical data. Section III introduces a binary linear programming (BLP) for modelling the multi-driver repositioning problem in real-time and integrates the solutions of MDP and BLP as the final repositioning solution. Section IV evaluates the performance of the proposed method by using the data from Manhattan, New York. Finally, Section V concludes the study and discusses potential extensions for future work.

II. Single-agent Markov Decision Process

The objective of repositioning is to improve the platform efficiency (e.g., serving more orders and reducing the number of order cancellations) and to enhance the experience for both passengers and vehicles (e.g., reducing the waiting times and the vacant duration). To develop the repositioning method, we assume the following assumptions: (i) Although both Picking up and Serving stages are implemented in the full-service cycle of vehicles in the network, the routing problem is reduced to the shortest path problem and is not studied explicitly in this paper. (ii) When vehicles are in the Searching stage, they will follow the repositioning instructions given by the platform. (iii) The platform continuously determines the repositioning instructions for unassigned vacant vehicles. Vacant vehicles during repositioning are not considered for a new repositioning instruction. (iv) While on the way driving to the repositioning destination, vacant vehicles are considered available by the platform to be matched with the upcoming orders en-route. (v) Passengers are assumed to be impatient, and their requests will be cancelled if not being responded to in a reasonable time frame (e.g., less than 1 minute).

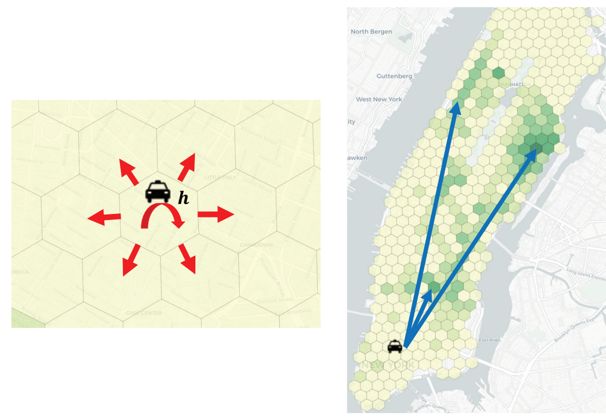
Without loss of generality, we employ hexagonal grids $H = \{h_1, ..., h_i, ..., h_N\}$ to represent an area unit in the digital map. Also, $T = \{t_1, ..., t_j, ..., t_M\}$ is introduced to indicate the repositioning decision step. Let Δ be the time interval between each two steps; i.e., $\Delta = t_j - t_{j-1}$, $\forall j \in 2, ..., M$. By viewing each vacant vehicle as an agent, we model the searching movement as a Markov Decision Process (MDP) endowed with a set of spatial actions. At the same time, the whole network is considered in the environment. The key elements of the MDP formulation are listed below.

State: The state of the vehicle (agent) is defined as a three-dimensional vector $s = (h, t, \mu)$, where $h \in H$ is the current hexagon as the location of the vehicle, $t \in T$ is the current time step, and $\mu \in \{0, 1, 2\}$ is the operating stages of the vehicle, where $\mu = 0$, 1, and 2 indicate Searching, Picking up, and Serving, respectively. States $(h, t, \mu), \mu = 1$ or $2, \forall h \in H$, $\forall t \in T$, which denote that the vehicle has been assigned to an order but not yet finished the ride are not considered for repositioning.

Action: For any vacant vehicles in States $s = (h, t, 0)$, $\forall h \in H$, $\forall t \in T$, their action sets are:

$$A(s) = \{h\} \cup A_{\text{neighbor}}(h) \cup A_{\text{global}}(t) \tag{1}$$

where $A_{\text{neighbor}}(h)$ is the set of neighboring hexagons of h and $A_{\text{global}}(t)$ indicates the set of top-k hexagons with the most unserved orders in current step t (see Figure 2). The purpose of introducing $A_{\text{global}}(t)$ is to enable vacant vehicles in distant locations to be repositioned to prosperous regions faster. The effect of considering global actions is tested where a Local MDP repositioning method ($A(s) = \{h\} \cup A_{\text{neighbor}}(h)$) is introduced as a benchmark in Section IV.



(A) Neighbor actions **(B) Global actions**

Fig. 2. Red arrows are neighboring action set $A_{\text{neighbor}}(h)$ and blue arrows represent global hot-spot set $A_{\text{global}}(t)$ comprised of top3 hexagons with the most unmatched orders at time step t. The darker color denotes the more unmatched orders.

Reward: Reward function evaluates the policy and quantifies the goal of the repositioning. In this work, reward function $R(s, a)$ is defined as the possibility that the vehicle can be matched after executing Action $a, \forall a \in A(s)$ in State $s = (h, t, 0)$, $\forall h \in H$, $\forall t \in T$.

$$R(s,a) = \begin{cases} \frac{\Delta}{\tau(s,a)} & \text{if vehicle receives a dispatch order} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\tau(s, a)$ is the shortest travel time from hexagon h of State s to repositioning destination hexagon a, and Δ is the interval between each two repositioning time steps. Intuitively, reward function $R(s, a)$ is defined as the ratio of Δ over $\tau(s, a)$ to account for the repositioning duration.

State Transition, $P(s, a, s')$ is the transition probability that State s' will be reached when Action $a, a \in A(s)$, is taken in State s. To define the transition process, we need to specify several parameters first.

(i) Matching probability, $p_{\text{match}}(h, t)$, estimates the probability that a vacant vehicle can be matched (by the central platform using global information) to an order when the vehicle is searching in hexagon h at time step t. To simplify, we assume that $p_{\text{match}}(h, t)$ is associated with the number of unserved orders and the number of vacant vehicles in h at time step t [40]:

$$p_{\text{match}}(h,t) = 1 - e^{-\theta \cdot \frac{C_{\text{order}}(h,t)}{C_{\text{vehicle}}(h,t)}} \tag{3}$$

where $C_{\text{order}}(h, t)$ and $C_{\text{vehicle}}(h, t)$ denote the number of unserved orders and vacant vehicles in hexagon h at time step

This article has been accepted for publication in IEEE Transactions on Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TITS.2022.3219381

t, respectively. In addition, θ is the parameter that describes how matching probability changes with demand-supply ratio $C_{\text{order}}(h,t)/C_{\text{vehicle}}(h,t)$.

(ii) Pick-up probability, $p_{\text{pickup}}(h,t,h^{'})$, denotes the probability that a vacant vehicle in h at time step t being matched with an order with origin at $h^{'}$. This parameter can be approximated as the ratio of the number of vehicles matched in h at time step t to pick up orders in $h^{'}$, denoted as $C_{\text{pickup\_vehicle}}(h,t,h^{'})$, to the total number of matched vehicles $C_{\text{matched\_vehicle}}(h,t)$ in h at time step t.

$$p_{\text{pickup}}(h,t,h^{'}) = \frac{C_{\text{pickup\_vehicle}}(h,t,h^{'})}{C_{\text{matched\_vehicle}}(h,t)}. \quad (4)$$

(iii) Destination probability, $p_{\text{dest}}(h,t,h^{'})$, measures the likelihood of the destination of an order being in $h^{'}$ when the order is picked up at h at time step t. This parameter can be estimated as the ratio of the number of orders ending in $h^{'}$ which originate from h at time step t, denoted as $C_{\text{dest\_order}}(h,t,h^{'})$, to the number of all orders $C_{\text{all\_order}}(h,t)$ in h at time step t.

$$p_{\text{dest}}(h,t,h^{'}) = \frac{C_{\text{dest\_order}}(h,t,h^{'})}{C_{\text{all\_order}}(h,t)}. \quad (5)$$

It is evident that the order dispatching algorithm (e.g., greedy matching, first come first served, instantaneous batch optimal) significantly affects the matching probability and pick-up probability in the system. However, destination probability is only an endogenous function of spatio-temporal distribution of the demand of the E-hailing system. All the above probabilities can be readily estimated from the historical data.

Figure 3 illustrates the outline of the state transition process. Suppose there is a vacant vehicle with State $s_0 = (h_0, t_0, 0)$ takes Action a to h_1, then if the vehicle is successfully matched with an order from h_2 to h_3, its state transition will be $s_0 \to s_1 \to s_2 \to s_3$, and transition probability $P(s_0, a, s_3)$ is defined as (See Figure 3 for the definition of States and time instances):

$$P(s_0,a,s_3) = p_{\text{match}}(h_1,t_1) \cdot p_{\text{pickup}}(h_1,t_1,h_2) \cdot p_{\text{dest}}(h_2,t_2,h_3). \quad (6)$$

If the vehicle is not matched with an order at h_1, it is in State s_4 and may be repositioned at the next decision step. In this case, the transition probability is:

$$P(s_0,a,s_4) = 1 - p_{\text{match}}(h_1,t_1). \quad (7)$$

In particular, note that once a vehicle executes Action a in $A_{\text{global}}(t)$, it is assumed it takes the shortest path to hexagon a. Under this scenario, the state transition process is rather sophisticated and complex since we need to consider each passing hexagon. For simplicity, the interactions in the intermediate hexagons are overlooked, and only the transitions in a are considered.

State-Action Value, $Q(s,a)$ is the expected reward that the vehicle can achieve being in State s after performing Action a, which in the Bellman formulation [41] is,

$$Q(s,a) = R(s,a) + \sum_{s^{'}} \gamma P(s,a,s^{'}) V^{*}(s^{'}) \quad (8)$$

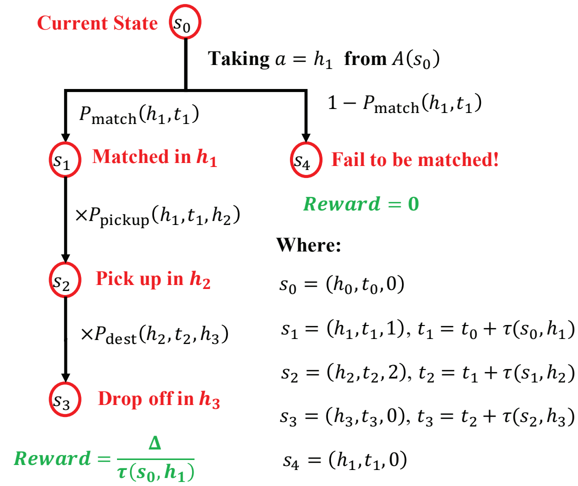


Fig. 3. Two scenarios of state transition: (i) a vehicle started with State s_0, took Action $a = h_1$ and found the order heading from h_2 to h_3, the state transition would be $s_0 \to s_1 \to s_2 \to s_3$, (ii) a vehicle started with State s_0, took Action $a = h_1$ and failed to be matched, the transition process would be $s_0 \to s_4$.

where γ is a discount factor, and is set as a constant slightly smaller than 1 to ensure the existence of a finite optimal expected payoff.

Optimal State Value, $V^{*}(s)$ is the optimal expected reward for the vehicle at State s,

$$V^{*}(s) = \max_{a \in A(s)} \{Q(s,a)\}. \quad (9)$$

To efficiently solve the proposed MDP and estimate $V^{*}(s)$ and $Q(s,a)$, dynamic programming approach is employed [42]. Since values of $V^{*}(s)$, $s = (h, t_M, 0)$, $\forall h \in H$ at step t_M are assumed to be zero, we can therefore solve the optimal value of $V^{*}(s)$ and $Q(s,a)$ by the backwards iteration as in Algorithm 1.

After solving the proposed MDP by Algorithm 1, optimal policy $\pi^{*}(s)$ is the policy that maximizes the expected reward of State s, which is

$$\pi^{*}(s) = \text{argmax}_{a \in A(s)} \{Q(s,a)\}. \quad (10)$$

However, note that the deterministic policy derived from Equation 10 is optimal (and effective) when there is only one vacant vehicle following the repositioning policy. As a simple instance, under single-agent MDP, all the vehicles with the same state might be repositioned to the same hexagon with the maximum Q-value. This 'over-reaction' phenomenon is undesirable since it may cause the hexagon with the maximum Q-value to become over-supplied in the future while leaving the other hexagons under-supplied. On the other hand, the e-hailing system is highly time-varying and passenger-centric, thus the real-time information should be considered to determine the service priority. Therefore, given interactions and competitions among multiple vacant vehicles available for repositioning, careful design of real-time optimization and its integration with the developed single-agent MDP is required. This issue is discussed in the next section.

Algorithm 1 Dynamic Programming Algorithm

Input: State S, Action A, Transition Probabilities P, Reward R

Output: $V^*(s)$ and $Q(s,a)$

```
Initialize two tables V*(s) and Q(s, a)
Let V*(s) = 0 for arbitrary State s = (h, t_M, 0), ∀h ∈ H
for t from t_{M-1} to t_1 do
    for h from h_N to h_1 do
        s ← (h, t, 0)
        Compute V*(s) and Q(s, a) by Equations 8 and 9
    end for
end for
return V*(s) and Q(s, a)
```

III. Real-time Multi-driver repositioning

To address the multi-driver repositioning problem in real-time, we develop an optimization program that takes the demand and supply real-time information as inputs and determines the repositioning actions as the optimal matching between vacant vehicles and hexagons. The repositioning is triggered in the form of sending instructions to idle vehicles that (i) are with no repositioning command (e.g., vehicles that just drop off a passenger) or (ii) have arrived at a previously announced repositioned destination.

Assume at time k vacant vehicles are collected as set D_k and unmatched orders in h are collected as set O_k^h. The variable $W_{h,k}$ is introduced to quantify the service priority of hexagon h at time k as:

$$W_{h,k} = (\sum_{o \in O_k^h} w_{o,k}^2) \cdot \eta_{h,k}, \forall h \in H \tag{11}$$

$$\eta_{h,k} = \frac{\max(|O_k^h| - C_{\text{dropoff\_driver}}(h,k,k+\delta), 0)}{|O_k^h|}, \forall h \in H. \tag{12}$$

Equation 11 defines $W_{h,k}$ as the sum of squares of orders' waiting times factored by $0 \le \eta_{h,k} \le 1$, where $w_{o,k}$ is the waiting time of unmatched order o at time k. Equation 12 defines $\eta_{h,k}$ to account for the number of drivers dropping off passengers in h within a short time range of $(k, k+\delta]$ (e.g. $\delta = 30$ [s]). It is assumed that those vehicles (becoming vacant in h during $(k, k+\delta]$) can pick up unmatched orders in h. Thus, those unmatched orders are not needed to be considered in the service priority of hexagon h at time k. Further, $|O_k^h|$ denotes the number of unmatched orders in h at time k. Overall, the service priority of hexagon h at time k, $W_{h,k}$, considers both the number of unmatched orders and their waiting times such that a higher value means more vacant vehicles are needed to be repositioned to hexagon h at time k.

Furthermore, it should be noted that if all vacant vehicles are directed to reposition to the hexagon with the maximum $W_{h,k}$, this will result in the same 'over-reaction' situation similar to the single-agent MDP. To address this issue, the answer rate function $A(\cdot)$ is introduced below

$$A(\frac{C_{\text{vehicle}}(h,k)}{C_{\text{order}}(h,k)}) = 1 - e^{-\beta \frac{C_{\text{vehicle}}(h,k)}{C_{\text{order}}(h,k)}}, \tag{13}$$

where $C_{\text{order}}(h,k)$ and $C_{\text{vehicle}}(h,k)$ denote the number of unserved orders and vacant vehicles in hexagon h at time step k, respectively. Note that β is the parameter that reflects how answer rate changes with respect to the supply-demand ratio $C_{\text{vehicle}}(h,k)/C_{\text{order}}(h,k)$.

Answer rate estimates the probability that an order can be matched with a driver when the order is waiting in hexagon h at time step k. Intuitively, suppose the supply-demand ratio increases infinitely as more and more drivers are repositioned to the target hexagon. In that case, the answer rate will approach 1, indicating that all passenger orders are fulfilled. Since the answer rate is a marginal diminishing function, the number of repositioned drivers should stay below a threshold to avoid an oversupplied situation. By setting an upper-bound answer rate $\hat{A}$, the maximum number of vehicles to be sent to hexagon h at time k is

$$c_{h,k} = |O_k^h| \cdot A^{-1}(\hat{A}) \tag{14}$$

where $A^{-1}(\cdot)$ is the inverse function of answer rate function and $A^{-1}(\hat{A})$ is the prescribed maximum supply-demand ratio.

The goal of the real-time multi-driver optimization program is to minimize the total waiting times of all unmatched orders so as to reduce the number of order cancellations. Let $x_{d,h}$ be a binary decision variable that equals 1 if vacant vehicle d is repositioned to hexagon h, and 0 otherwise. Let λ be the optimization interval for real-time repositioning component. In general, λ can be selected to be shorter than Δ and be equal to the order dispatching interval. At each decision point $k = 0, \lambda, 2\lambda, ..., K\lambda$, the real-time multi-driver repositioning problem is formulated as the following Binary linear program (BLP):

$$\max \sum_{h \in H} \frac{W_{h,k}}{\tau(\mathrm{p}_{d,k}, \mathrm{c}_h)} \cdot x_{d,h} \tag{15}$$

$$\text{s.t.} \sum_{d \in D_k} x_{d,h} \le c_{h,k}, \forall h \in H \tag{16}$$

$$\sum_{h \in H} x_{d,h} \le 1, \forall d \in D_k. \tag{17}$$

The above BLP problem in Equations 15-17 is a maximum-weight many-to-one matching problem [43] between vacant vehicles D_k and hexagons H, in which the weight function between driver d and hexagon h is the ratio of $W_{h,k}$ over $\tau(\mathrm{p}_{d,k}, \mathrm{c}_h)$, where $\tau(p_{d,k}, \mathrm{c}_h)$ is the shortest travel time from vehicle d's current position, $\mathrm{p}_{d,k}$, to the center of hexagon h, c_h. Equation 15 presents the objective function of the real-time repositioning. There are two main factors considered in the objective function: (i) repositioning priority, a hexagon with higher $W_{h,k}$ is more likely to be the destination of repositioned vacant vehicles, and (ii) repositioning duration, a hexagon with shorter travel time from the vacant vehicle, $\tau(\cdot,\cdot)$, will have a higher priority to be the repositioning destination of that vehicle. Equation 16 ensures the number of repositioned vehicles to hexagon h is less than or equal to the maximum capacity $c_{h,k}$. The constraint in Equation 17 guarantees vehicle d will be assigned to at most one repositioning instruction.

Note that the exact solution of BLP (15-17) can be computationally expensive because of the overwhelming number of

This article has been accepted for publication in IEEE Transactions on Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TITS.2022.3219381

vacant vehicles and hexagons. To tackle this, we transform the problem to a Minimum Cost Flow (MCF) problem [44]. A classical MCF solver [45] is used to solve the problem and achieve the optimal solution X^*. Afterwards, the following integrated repositioning approach is introduced:

- For every vehicle d that has a non-zero solution from BLP optimization, it would follow the repositioning instruction from X^*.
- For every vehicle d that is not assigned a repositioning instruction from BLP optimization, that is, $x_{d,h} = 0, \forall h \in H$, it would follow the repositioning instruction by MDP solution as in Equation 10.

Accordingly, the integrated approach allocates priority to each immediate request. The idle vehicles with a non-zero BLP solution are firstly repositioned to destinations with higher service priorities. Then, the idle vehicles with an all-zero BLP solution will follow the MDP solution to optimize the long-term operational efficiency of the platform.

IV. Numerical Experiments

In this section, we examine the performance of the proposed repositioning method. All the experiments are conducted in a simulation environment in which Manhattan island is considered. The objective of the experiments is to evaluate the effectiveness of the proposed method and several benchmark methods in terms of platform efficiency and passengers' and drivers' experience.

A. Network and data

As shown in Figure 4, we partition the Manhattan area into 672 similar hexagons with a diagonal length of approximately 340 [m]. The road network comprises 6,533 nodes and 10,206 directed links, including streets, highways, bridges, and tunnels. The shortest paths and the travel time among nodes are pre-calculated and stored in a look-up table. Although the proposed method is based on the spatial hexagonal abstraction, the experiments are done on the detailed road network, considering the repositioning, picking up, and serving procedures at link and node levels.

The data used are Manhattan taxi datasets in December 2020 collected from Yellow Cab's website[1]. The trip order data include order location (origin and destination) and order request time. Origins and destinations are mapped into the nodes of the road network. Trip orders with the origin or destination falling outside the spatial boundary are disregarded. The simulation is conducted for a 3 hours period between 07:00 AM and 10:00 AM. 150 vacant vehicles are initially generated in the road network at 07:00 AM randomly. The number and location of new arriving vehicles are considered to be stochastic and time-varying to replicate an erratic feature of the real world. The new vacant vehicles' arriving rate is sampled from uniform discrete distributions with ranges between 7 and 19, 1 and 7, and 5 and 12 units per minute during 07:00 AM - 07:30 AM, 07:30 AM - 08:30 AM, and 08:30 AM - 10:00 AM, respectively. Furthermore, vehicles are assumed to leave the network once they receive no matching for more than 30 [min] or stochastically (e.g., ∼10%) after each drop-off. This is to replicate drivers being part-time contractors and sensitive to platform efficiency.

[1]https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

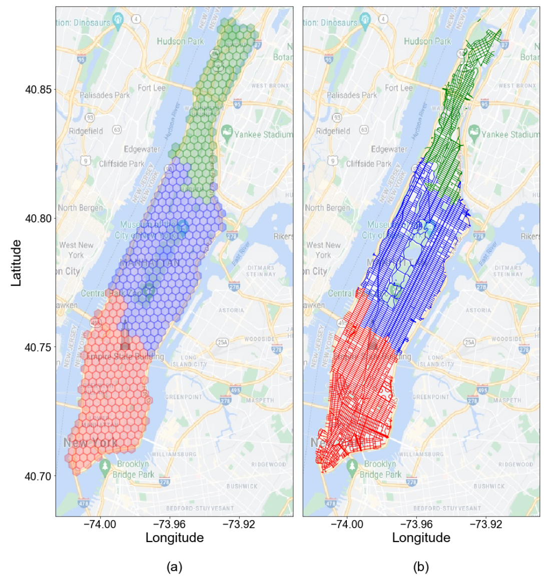


Fig. 4. There are two spatial abstractions of the Manhattan area: (a) the hexagonal grid and (b) the road link and node. Red, blue green parts in the figures represent lower, central and upper Manhattan, respectively.

Each passenger is assigned a matching patience time stochastically drawn from a truncated Gaussian distribution in the range of 0.5 [min] to 1 [min] with a mean of 0.75 [min] and a standard deviation of 0.15 [min]. Similarly, a pickup patience time is assigned to each passenger stochastically drawn from a truncated Gaussian distribution in the range of 3 [min] to 7 [min] with a mean of 5 [min] and a standard deviation of 2 [min]. An order will be cancelled if not being matched or picked up within the matching and pickup patience times, respectively. The platform determines the order dispatching every ten seconds based on the optimization model elaborated in Order Dispatching. The 'dispatching radius' is set to 2 [km] to avoid the wild goose chase problem (i.e., long-distance dispatching). All the vehicles are assumed to follow a fixed driving speed of 20 [km/h]. Future research can address the time-varying effect of congestion on the operation of repositioning methods.

B. Benchmarks and performance metrics

The performance of the following benchmark repositioning methods are evaluated.

1) **Parking:** The platform advises vacant vehicles to park at their current location once they drop off the passengers.
2) **Random Walk:** The platform advises vacant vehicles to head toward a hexagon randomly among their neighbor hexagons.
3) **Local MDP:** The platform chooses the repositioning action for vacant vehicles according to the optimal policy in Equation 10 without the set of global actions.

4) **MDP Walk:** The platform chooses the repositioning action for vacant vehicles according to the optimal policy in Equation 10.
5) **Real-time:** The platform advises vacant vehicle d at time k to reposition by solving the following optimization program.

$$\max \sum_{h \in H} x_{d,h} \cdot \frac{W_{h,k}}{\tau(\mathrm{p}_{d,k}, \mathrm{c}_h)} \tag{18}$$

$$\text{s.t.} \sum_{h \in H} x_{d,h} \leq 1. \tag{19}$$

If $W_{h,k} = 0$, $\forall h \in H$ at time k, the vehicles follow Random Walk repositioning. This benchmark only utilizes real-time information and does not incorporate the MDP model.
6) **Real-time Multi-driver:** The platform advises vacant vehicles to reposition according to the proposed method (integration of MDP and real-time components), explained in Section III.

Note that if any vacant vehicles are failed to be matched after arriving at its repositioning destination, they will be repositioned again by the same method.

We select Random Walk (see Parameter Estimation) as the initial repositioning method in the simulator to generate 10 weekdays (1st to 14th) training data. Subsequently, the matching, pick-up, destination probabilities and answer rate can be estimated from the training data. The single-agent MDP is solved by setting γ to 0.8 and the top three hot spots as global actions. Further, we introduce four new days (15th, 16th, 21st, and 22nd of Dec.) as the test data. We adopt seven evaluation metrics:

1) **Avg. Response**. The average number of orders that are successfully served per test day.
2) **Avg. Cancellation**. The average number of orders that are canceled per test day.
3) **Avg. Response time**. The average response time to orders from requesting (arriving in the network) to being matched.
4) **Avg. Pick-up time**. The average pick-up time of orders from being responded to being picked up. It is worth mentioning this is equal to the average deadheading time for vehicles.
5) **Avg. Occupied rate**. The average occupied rate per vehicle, defined as the ratio of the time spent on serving orders to the total operating time of the vehicle.
6) **Avg. Leaving vehicles**. The average number of leaving vehicles per test day.
7) **Avg. Repositioning distance**. The average distance of the repositioning instruction per vehicle.

C. Results

Table I summarizes the numerical results of the six repositioning methods, where parameters Δ, λ, δ, and $\hat{A}$ are fine-tuned based on multiple rounds of testing. The Real-time Multi-driver repositioning method results in the highest number of served passengers and the lowest number of order cancellations, 85.1% average response rate and 14.9% average cancellation rate. These indicate a higher profit and customer retention for the platform. More specifically, the proposed method achieves the lowest cancellation rate in 77% of 672 hexagons in the network; see Figure 5 in which the cancellation rates of all hexagons with each repositioning method is depicted. A notable observation in Figure 5 is that with the proposed repositioning method, the hexagons with relatively higher cancellation rates are mostly located at the edge of the network where connectivity is low. This is expected as areas with more connectivity have a higher chance to be the destination or on the route of repositioning vehicles.

The highest contractor (i.e., driver) retention is also achieved by using the proposed real-time multi-driver repositioning; out of 1576.3 individual drivers (averaged over the 4 test days), 762.5 of them exit the network during the 3 hours. That is 19% more retention rate with respect to the parking strategy. In addition, the time drivers carry a passenger increases to 61.5% of their service period in the network. The proposed repositioning method also leads to significant improvement in passengers' experience by achieving the lowest average response and pickup times; an average overall wait time of 155.7 [s] compared to the 231.4 [s] wait time of the parking method (33% and 9% improvements with respect to the Parking and Real-time only methods, respectively). Another observation is that though MDP Walk and Real-time methods could achieve promising performance, they would lead to extra repositioning mileage for each vehicle. The main reason is that most vacant vehicles under the two policies are more likely to go to the global hot spots. In contrast, the proposed repositioning method would balance lower repositioning costs and better system performance.

Results in Table I further highlight an insignificant improvement in the six evaluation metrics between Parking and Random Walk methods, specifically taking into account that Parking method does not require repositioning trips. This suggests that random cruising might not be a more productive repositioning method than parking because it will impose fuel costs on the drivers and congestion externalities to the urban network. It also should be noted that MDP walk method that incorporates global actions, see Equation 1, can bring improvements in all evaluation metrics compared to the Local MDP method that represents a local myopic historical data-based repositioning solution.

Figure 6 presents the average operational numbers of vehicles and orders (on four testing days) per minute of the six repositioning methods. In the right half of the figure, three categories are waiting (blue), matched (pink), and cancelled (teal blue) orders over the three testing hours. On the left side, there are three categories idle (cyan), repositioned (light pink), and occupied (orange) drivers over the three testing hours. The supply and demand are not uniformly distributed during the operational period, which creates the 'over-supply period' (07:00 AM to 07:30 AM with a negligible number of waiting orders and a considerable number of idle and repositioned vehicles) and the 'supply-shortage period' (08:00 AM to 09:00 AM). Note that the number of parking drivers (cyan bars) in Real-time Multi-driver is significantly more than in Real-time,

This article has been accepted for publication in IEEE Transactions on Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TITS.2022.3219381

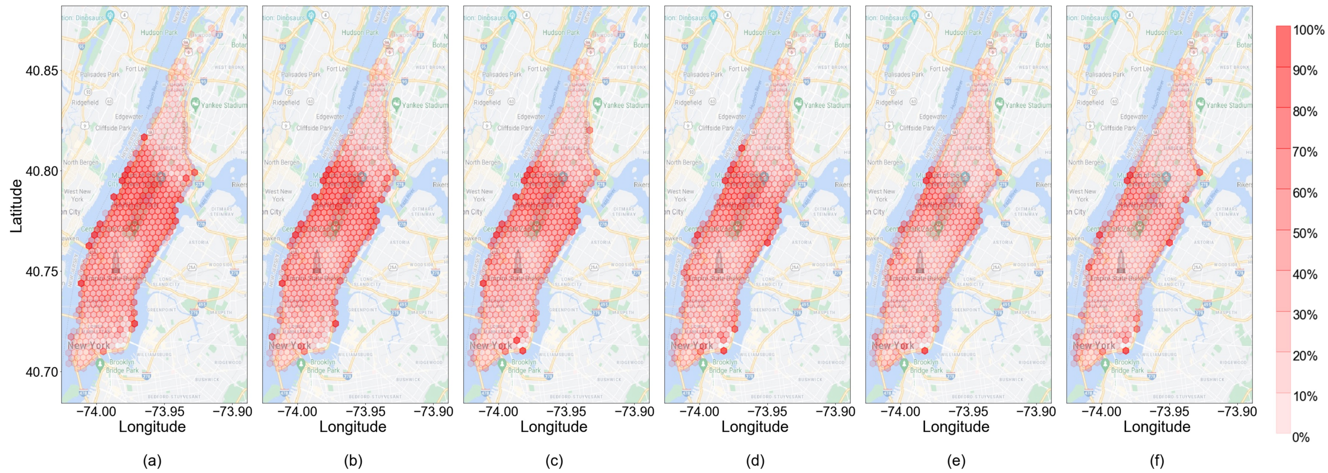


Fig. 5. Comparison of the cancellation rate of the six repositioning methods. The darker color denotes the higher cancellation rate. (a-f) correspond to Parking, Random Walk, Local MDP, MDP Walk, Real-time, and Real-time Multi-driver repositioning methods, respectively.

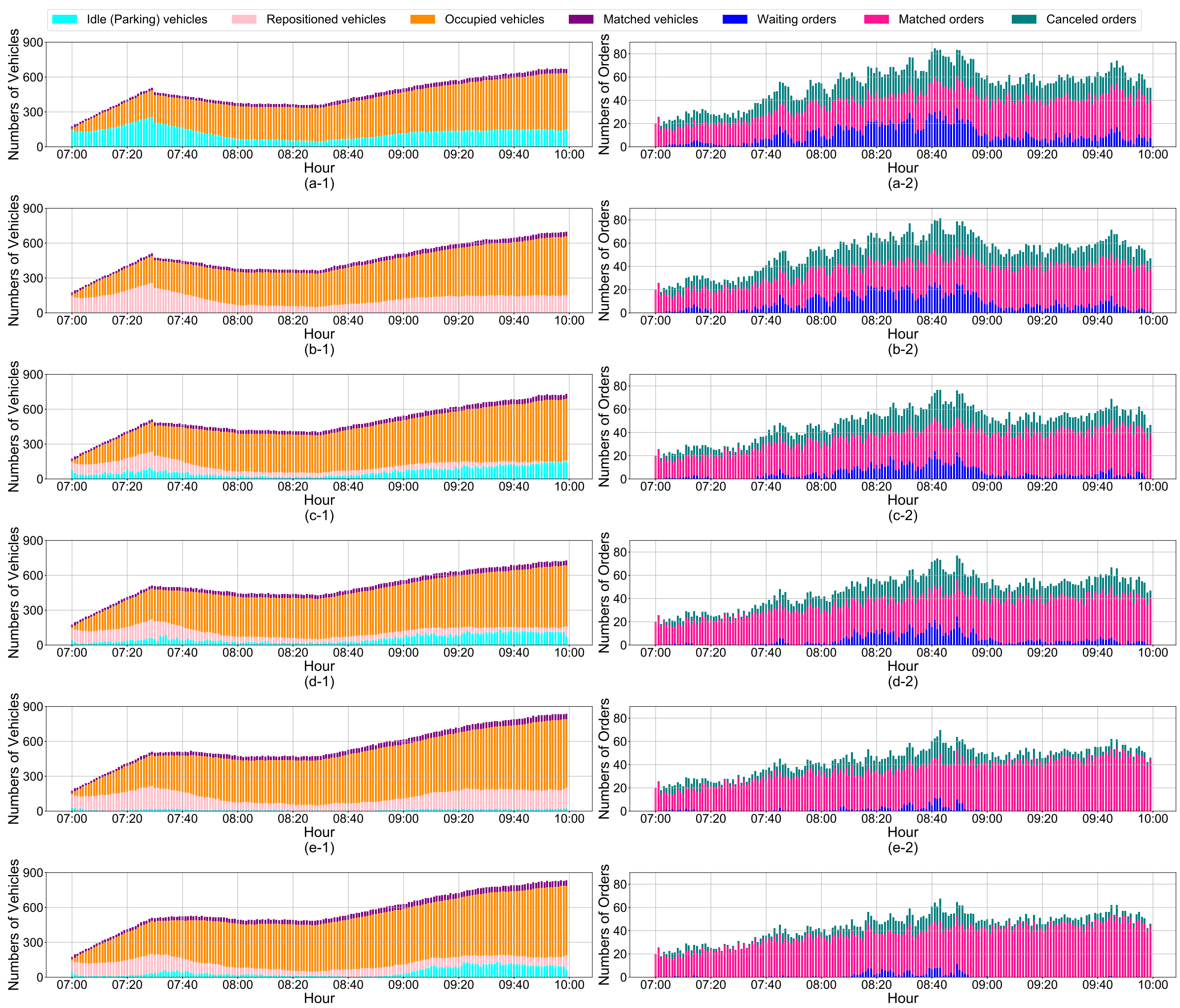


Fig. 6. Average numbers of orders and vehicles per minute of six methods within four testing days. (a-f) correspond to Parking, Random Walk, Local MDP, MDP Walk, Real-time, and Real-time Multi-driver repositioning methods, respectively.

This article has been accepted for publication in IEEE Transactions on Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TITS.2022.3219381

TABLE I

The results with different repositioning methods during 07:00 am -10:00 am. The numbers in the parentheses are average response rate and average cancellation rate of the platform. 10 weekdays training data are used to training the Q functions in single-agent MDP, and four new days are tested by different repositioning methods. (*: $\Delta = 1$ [min]; **: $\Delta = 1$ [min], $\delta = 30$ [s]; ***: $\Delta = 1$ [min], $\lambda = 10$ [s], $\delta = 30$ [s], $\hat{A} = 0.99$).

Methods	Avg. Response	Avg. Cancellation	Avg. Response time (s)	Avg. Pick-up time (s)	Avg. Occupied rate	Avg. Leaving vehicles (veh)	Avg. Repositioning distance (km)
Parking	4792.5 (62.7%)	2844.2 (37.2%)	7.4	224.0	55.9%	940.0	0.0
Random Walk	4909.5 (64.2%)	2733.5 (35.8%)	6.1	220.0	56.0%	916.0	0.3
Local MDP	5385.0 (70.4%)	2259.5 (29.6%)	3.6	204.1	58.3%	886.0	0.2
MDP Walk (*)	5587.2 (73.1%)	2057.0 (26.9%)	3.0	194.4	59.1%	889.5	0.6
Real-time (**)	6236.5 (81.6%)	1408.0 (18.4%)	1.7	169.3	60.0%	784.0	0.7
Real-time Multi-driver (***)	6506.2 (85.1%)	1138.2 (14.9%)	0.8	154.9	61.5%	762.5	0.3

which is consistent with the observation of Avg. repositioning distance in Table I. Thus, it can be observed that the Real-time Multi-driver method consistently outperforms other repositioning methods by achieving the most number of matched orders and the least number of waiting and cancelled orders. This shows considerable operation efficiency by implementing the proposed repositioning method. In summary, the proposed method brings remarkable improvements to both platform efficiency (serving more orders, reducing order cancellations, and maintaining sufficient fleet size) and users' experience (reducing passengers' waiting times and increasing drivers' occupied rate).

V. Summary and Future Research

This paper has proposed a novel method for the centralized e-hailing platform to reposition vacant vehicles to curb the network's mismatch between supply and demand. The proposed method is designed to optimize long-term operational efficiency and immediate demand satisfaction simultaneously. A single-agent MDP model achieves the former to evaluate the long-term influence of the repositioning on platform efficiency. The MDP model is trained on 10-day simulation data and aims to derive the optimal repositioning policy. Considering the fact that passengers are impatient and will cancel their order if not being matched in a reasonable time, the real-time multi-driver repositioning is addressed by a binary linear program (BLP) to prioritize the repositioning destinations based on the passengers' waiting times. The final repositioning method is the integration of the solutions of the MDP and real-time components. Through extensive numerical experiments based on field data of Manhattan, the proposed method improves both platform efficiency and users' (passengers' and drivers') experience.

Various extensions can be explored in the future. An alternative way to curb the supply and demand mismatch is dynamic fleet size management methods such as incentivizing when, where, and which vehicles join and exit the system. One possible way can be a joint spatial-temporal monetary incentive by offering a bonus to drivers who complete a repositioning instruction. There is also a strong motivation in the combined design of order dispatching and vehicle repositioning. For instance, the dispatching optimization can prioritize trip orders based on their drop-off destination as potential repositioning destinations. Another possible extension lies in adopting multi-agent reinforcement learning (MARL) and grouping drivers by their working behaviors and driving patterns. To do so, the refined analyses that consider the market-behavioural differences among various driver groups [37] are required before modelling.

Acknowledgment

All the data and codes of this work are shared on the GitHub. This research was partially funded by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) DE210100602.

Appendix

Order dispatching

For order dispatching, we assume that the e-hailing platform attempts to minimize the total pick-up time or the waiting times of passengers (deadheading of vehicles). At time k, we set $U_{o,d}$ as the reciprocal of pick-up time between unmatched order o and driver d:

$$U_{o,d} = \frac{1}{\tau(st_o, p_{d,k})} \tag{20}$$

where $\tau(st_o, p_{d,k})$ denotes the travel time from order o's origin st_o to driver d's current position $p_{d,k}$.

Assume at time k vacant vehicles are collected as set D_k and unmatched orders are collected as set O_k. The dispatching problem can be formulated as the following Binary linear program (BLP):

$$\max \sum_{d \in D_k} \sum_{o \in O_k} y_{o,d} \cdot U_{o,d} \tag{21}$$

$$\text{s.t.} \sum_{o \in O_k} y_{o,d} \leq 1; \forall d \in D_k \tag{22}$$

$$\sum_{d \in D_k} y_{o,d} \leq 1; \forall o \in O_k \tag{23}$$

$$y_{o,d} \in \{0, 1\}; \forall o \in O_k, \forall d \in D_k \tag{24}$$

where the first constraint guarantees that each driver will dispatch to at most one order or continue unmatched at this dispatching instance; the second constraint ensures that each order is assigned to at most one driver or is unserved until next decision round; and the third constraint sets the decision variables are binary.

If we abstract unmatched drivers set D_k and unmatched orders set O_k as two sets of vertices, and valid matching pairs as the set of edges, the BLP program in Equations 21-24 can be represented as a bipartite graph matching problem. In general, the initial bipartite graph is a fully connected graph where every possible edge between drivers and orders exists. To reduce computational complexity, we further introduce a 'dispatching radius' [46] to eliminate edges whose pick-up distance exceeds the dispatching radius (e.g., 2 [km]). Consequently, we can find a more compact graph and employ the Kuhn-Munkres (KM) algorithm [47] to solve it.

Parameter Estimation

Parameters θ and β in Equations 3 and 13 can be estimated respectively by fitting matching probability and answer rate using 10 weekdays training data. We have tested three repositioning methods, Parking, Random Walk, and Real-time, to estimate the parameters and calculate the R-squared and RMSE. The result of the estimation can be seen in Table II. As for the goodness-of-fit statistics, all three policies have an R-squared value higher than 0.74. It can be observed that the repositioning strategies may have only a modest effect on θ and β.

TABLE II
THE ESTIMATED PARAMETERS FROM THE TRAINING DATA

Policy	$\hat{\theta}$	R^2	$RMSE$
Parking	0.51	0.74	0.23
Random Walk	0.48	0.80	0.11
Real-time	0.44	0.77	0.19
Policy	$\hat{\beta}$	R^2	$RMSE$
Parking	0.82	0.86	0.09
Random Walk	0.89	0.96	0.06
Real-time	0.79	0.82	0.14

REFERENCES

[1] C. Smith, "110 amazing uber stats and facts (2017) by the numbers," 2018.

[2] Z. Xu, Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye, "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*, 2018, pp. 905–913.

[3] M. Hu and Y. Zhou, "Price, wage and fixed commission in on-demand matching," *Available at SSRN 2949513*, 2019.

[4] H. Hamedmoghadam, M. Ramezani, and M. Saberi, "Revealing latent characteristics of mobility networks with coarse-graining," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

[5] J. Ke, H. Yang, H. Zheng, X. Chen, Y. Jia, P. Gong, and J. Ye, "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4160–4173, 2018.

[6] L. Chen, A. H. Valadkhani, and M. Ramezani, "Decentralised cooperative cruising of autonomous ride-sourcing fleets," *Transportation Research Part C: Emerging Technologies*, vol. 131, p. 103336, 2021.

[7] J. D. Angrist, S. Caldwell, and J. V. Hall, "Uber vs. taxi: A driver's eye view," National Bureau of Economic Research, Tech. Rep., 2017.

[8] M. Nourinejad and M. Ramezani, "Ride-sourcing modeling and pricing in non-equilibrium two-sided markets," *Transportation Research Part B: Methodological*, 2019.

[9] H. Yuan, Y. Qian, R. Yang, and M. Ren, "Human mobility discovering and movement intention detection with gps trajectories," *Decision Support Systems*, vol. 63, pp. 39–51, 2014.

[10] N. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang, "Dynamic ride-sharing: A simulation study in metro atlanta," *Procedia-Social and Behavioral Sciences*, vol. 17, pp. 532–550, 2011.

[11] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[12] X. Zhan, X. Qian, and S. V. Ukkusuri, "A graph-based approach to measuring the efficiency of an urban taxi service system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2479–2489, 2016.

[13] R. Baldacci, V. Maniezzo, and A. Mingozzi, "An exact method for the car pooling problem based on lagrangean column generation," *Operations Research*, vol. 52, no. 3, pp. 422–439, 2004.

[14] D. Pelzer, J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt, "A partition-based match making algorithm for dynamic ridesharing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2587–2598, 2015.

[15] J. P. Dickerson, K. A. Sankararaman, A. Srinivasan, and P. Xu, "Allocation problems in ride-sharing platforms: Online matching with offline reusable resources," *arXiv preprint arXiv:1711.08345*, 2017.

[16] E. Özkan and A. R. Ward, "Dynamic matching for real-time ride sharing," *Stochastic Systems*, vol. 10, no. 1, pp. 29–70, 2020.

[17] G. Lyu, W. C. Cheung, C.-P. Teo, and H. Wang, "Multi-objective online ride-matching," *Available at SSRN 3356823*, 2019.

[18] Z. Wang, Z. Qin, X. Tang, J. Ye, and H. Zhu, "Deep reinforcement learning with knowledge transfer for online rides order dispatching," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 617–626.

[19] X. Tang, Z. T. Qin, F. Zhang, Z. Wang, Z. Xu, Y. Ma, H. Zhu, and J. Ye, "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, 2019, pp. 1780–1790.

[20] Y. Yang, Q. Tian, and Y. Wang, "Who is more likely to get a ride and where is easier to be picked up in ride-sharing mode?" *Journal of Management Science and Engineering*, vol. 5, no. 4, pp. 264–286, 2020.

[21] G. Jiao and M. Ramezani, "Incentivizing shared rides in e-hailing markets: Dynamic discounting," *Transportation Research Part C: Emerging Technologies*, vol. 144, p. 103879, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22002923

[22] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 37, pp. 13 290–13 294, 2014.

[23] M. M. Vazifeh, P. Santi, G. Resta, S. H. Strogatz, and C. Ratti, "Addressing the minimum fleet problem in on-demand urban mobility," *Nature*, vol. 557, no. 7706, pp. 534–538, 2018.

[24] M. Ramezani and M. Nourinejad, "Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach," *Transportation Research Part C: Emerging Technologies*, vol. 94, pp. 203–219, 2018.

[25] R.-H. Hwang, Y.-L. Hsueh, and Y.-T. Chen, "An effective taxi recommender system based on a spatio-temporal factor analysis model," *Information Sciences*, vol. 314, pp. 28–40, 2015.

[26] K. Zhang, Z. Feng, S. Chen, K. Huang, and G. Wang, "A framework for passengers demand prediction and recommendation," in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 340–347.

[27] H. WANG and Z. WANG, "Short-term repositioning for empty vehicles on ride-sourcing platforms," 2020.

[28] Q. Tang, Y. Zhang, and M. Zhou, "Vehicle repositioning under uncertainty," *Available at SSRN 3612626*, 2020.

[29] A. Braverman, J. G. Dai, X. Liu, and L. Ying, "Empty-car routing in ridesharing systems," *Operations Research*, vol. 67, no. 5, pp. 1437–1452, 2019.

[30] X. Zhou, H. Rong, C. Yang, Q. Zhang, A. V. Khezerlou, H. Zheng, M. Z. Shafiq, and A. X. Liu, "Optimizing taxi driver profit efficiency: A spatial network-based markov decision process approach," *IEEE Transactions on Big Data*, 2018.

[31] X. Yu, S. Gao, X. Hu, and H. Park, "A markov decision process approach to vacant taxi routing with e-hailing," *Transportation Research Part B: Methodological*, vol. 121, pp. 114–134, 2019.

[32] 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," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 91–113, 2020.

[33] T. Verma, P. Varakantham, S. Kraus, and H. C. Lau, "Augmenting decisions of taxi drivers through reinforcement learning for improving revenues," in *Twenty-Seventh International Conference on Automated Planning and Scheduling*, 2017.

[34] Y. Gao, D. Jiang, and Y. Xu, "Optimize taxi driving strategies based on reinforcement learning," *International Journal of Geographical Information Science*, vol. 32, no. 8, pp. 1677–1696, 2018.

[35] K. Lin, R. Zhao, Z. Xu, and J. Zhou, "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*, 2018, pp. 1774–1783.

[36] Z. Xu, D. AMC Vignon, Y. Yin, and J. Ye, "An empirical study of the labor supply of ride-sourcing drivers," *Transportation Letters*, pp. 1–4, 2020.

[37] M. Ramezani, Y. Yang, J. Elmasry, and P. Tang, "An empirical study on characteristics of supply in e-hailing markets: a clustering approach," *Transportation Letters*, pp. 1–14, 2022.

[38] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[39] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337.

[40] N. Buchholz, "Spatial equilibrium, search frictions and dynamic efficiency in the taxi industry," Working Paper, Tech. Rep., 2018.

[41] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.

[42] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[43] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*. Springer Science & Business Media, 2003, vol. 24.

[44] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.

[45] S. Mitchell. (2021, jun) Optimization with pulp. [Online]. Available: https://coin-or.github.io/pulp/

[46] H. Yang, X. Qin, J. Ke, and J. Ye, "Optimizing matching time interval and matching radius in on-demand ride-sourcing markets," *Transportation Research Part B: Methodological*, vol. 131, pp. 84–105, 2020.

[47] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

Yue Yang received the B.Sc. degree in industrial engineering from University of Science and Technology Beijing, and the M.Sc. degree in transportation engineering from Beihang University. Now he is a Ph.D. student majoring in transport engineering in the University of Sydney, TransportLab. His research interests focus on shared-mobility and ride-sourcing operation and management.

Mohsen Ramezani received the B.Sc. and M.Sc. degrees in electrical engineering (control systems) from the University of Tehran, Iran, in 2008 and 2010, respectively, and the Ph.D. degree in transportation from the École Polytechnique Fédérale de Lausanne, Switzerland, in 2014. He is currently a Senior Lecturer with TransportLab, School of Civil Engineering, The University of Sydney. His research interests include point to point transport services, traffic control, data analytics, and traffic flow theory and operations. He was the recipient of the 2015 IEEE Intelligent Transportation Systems Society 1st Prize Best Ph.D. Dissertation. He is an Australian Research Council (ARC) Discovery Early Career Research Award (DECRA) fellow.