Australasian Transport Research Forum 2021 Proceedings
8-10 December, Brisbane, Australia
Publication website: http://www.atrf.info

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

Yue Yang[1], Mohsen Ramezani[1]

[1]University of Sydney, school of Civil Engineering

Email for correspondence: mohsen.ramezani@sydney.edu.au

Abstract

Internet-based ride-sourcing platforms have become an important component of urban transportation systems in recent years. The spatial-temporal mismatch of supply (available vehicles) and demand (passenger requests) deteriorates the performance of ride-sourcing services. Hence, repositioning available vehicles to service unserved requests can be effective. In this paper, we propose a real-time repositioning method that considers both the responsiveness to immediate demand and the long-term operation efficiency simultaneously. The proposed method is achieved by integrating the solutions of two procedures: i) A single-agent Markov Decision Process (MDP) model is employed to evaluate the long-term influence of the repositioning on platform efficiency. ii) A binary nonlinear programming (BNLP) is introduced to model the multi-driver repositioning problem in real-time, which is solved by an Iterative Edge Cutting (IEC) algorithm. Numerical experiments demonstrate that the proposed method significantly outperforms several repositioning benchmarks regarding both platform efficiency, including servicing more orders and reducing order cancellations, and users' experience including reducing passengers' waiting time and increasing drivers' occupied rate.

1.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 automatically matched with passengers without random cruising on streets. E-hailing platforms continuously receive passengers' trip requests, geographical coordinates, and occupancy status of E-hailing vehicles (driven by contractors), and periodically dispatch idle drivers to serve unassigned orders (Vonolfen et al. 2016). Also, platforms might reposition the drivers who fail to receive a pick-up order to a different location for the prospect of less cruising time in the future.

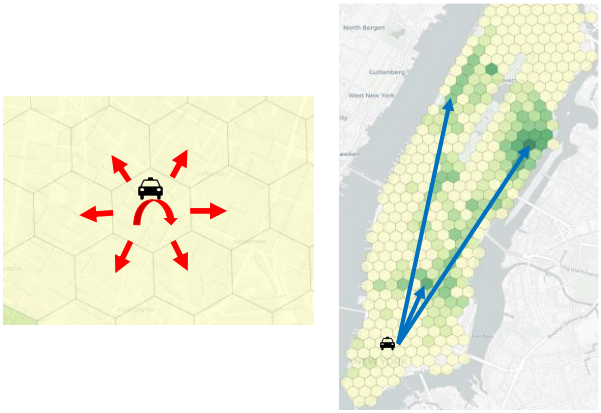
Aiming to minimize pick-up times or maximize system profit, a plethora of order dispatching algorithms have been investigated (Agatz et al. 2011, 2012). Nonetheless, once drivers are not matched in the dispatching procedure, E-hailing systems face a critical question (or an opportunity): where is the best location for idle drivers to find a passenger at subsequent times?

To design a repositioning method to tackle the above question, we model the repositioning process as a many-to-one matching problem between idle drivers and repositioning destinations. Each matched driver-destination pair can be considered as a repositioning instruction for the drivers and determined by two terms: (i) a term representing the impact of the instruction on the future efficiency and (ii) an instantaneous matching probability attained from the real-time demand-supply information. These are achieved by employing an MDP

model to evaluate the long-term influence of the repositioning on platform efficiency, developing a multi-driver optimization method to optimize the real-time demand satisfaction, and integrating the solutions of the MDP and the multi-driver optimization into the final repositioning solution.

2. Single-agent Markov Decision Process

Without loss of generality, we employ hexagonal grids $\boldsymbol{H} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_N}\}$, to represent an area unit in the digital map. Also, $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \dots, \boldsymbol{t_M}\}$ is introduced to indicate the rolling repositioning decision step. Let $\boldsymbol{\Delta}$ be the time interval between each two steps. By viewing each idle driver as an intelligent agent, we model the searching movement as a Markov Decision Process (MDP) endowed with a set of spatial actions, while the whole operating area is considered as the environment. The key elements of the MDP formulation are listed below.



(A) Neighbor actions (B) Global actions

Figure 1: The basic action setting of MDP, red arrows are neighboring actions and blue arrows represent global actions.

(1) State: $s = (h, t, \mu)$, where h and t are current hexagon and time step, $\mu = \{0,1,2\}$ indicate Searching, Picking up, and Serving.

(2) Action: For any idle drivers in States $s = (h, t, 0), \forall\, h \in H, \forall\, t \in T$, their action sets are: $A(s) = A_{\text{neighbor}}(h) \cup A_{\text{global}}(t)$, where $A_{\text{neighbor}}(h)$ is the set of neighboring hexagons of h including h itself. and $A_{\text{global}}(t)$ indicates the set of top-k hexagons with the most unserved orders in current step t (See Fig. 1).

(3) Reward: Reward function evaluates the policy and quantifies the goal of the repositioning.

$$R(s,a) = \begin{cases} \frac{\Delta}{\tau(s,a)}, & \text{if driver receives an order} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $\tau(s,a)$ is the shortest travel time from hexagon h of State s to destination a.

(4) State-Action Value, $Q(s,a)$ is the expected reward that the driver can earn being in State s after performing Action a:

$$Q(s,a) = R(s,a) + \textstyle\sum_{s'} \gamma P(s,a,s') V^*(s') \tag{2}$$

where γ is a discount factor, and $P(s,a,s')$ is the state transition probability from State s to s' by taking Action a, which can be estimated from the historical data.

(5) Optimal State Value, $V^*(s)$ is the optimal expected reward for the driver at State s:

$$V^*(s) = \max\{Q(s,a) | \forall\, a \in A(s)\}. \tag{3}$$

(6) Optimal Policy, $\pi^*(s)$ is the optimal policy that maximizes the expected reward of State s, which is

$$\pi^*(s) = \text{argmax}_{a \in A(s)} \{Q(s,a)\}. \quad (4)$$

The proposed MDP above can be solved by dynamic programming approach (Cormen et al. 2009). However, note that the deterministic policy derived from Equation 4 is optimal (and effective) when there is only one idle driver following the repositioning policy. Intuitively, given interactions and competitions among multiple idle drivers available for repositioning, careful design of multi-driver optimization based on the developed single-agent MDP is required.

3. Multi-driver Optimization

To address the repositioning among multiple drivers, an optimization model is proposed in this section. The repositioning problem between multiple drivers and a hexagon can be modeled as a many-to-one matching problem.

Suppose there are two finite sets D_t^{n}, a set of drivers with no repositioning instruction or who have arrived at the repositioned destination, and D_t^{e}, a set of idle drivers who have been repositioned and are en-route, collected at each decision step t. Here, only idle drivers in D_t^{n} are considered for the optimization model. The optimization problem at decision step t aims to maximize the estimated number of matched drivers during$(t, t+\Delta]$, so a binary nonlinear programming (BNLP) can be formulated as:

$$\max \sum_h \sum_d x_{d,h} \cdot p_{\text{match}}(h, t+\Delta) \quad (5)$$

$$\text{s.t.} \sum_{h \in A((h_{d,t},t,0))} x_{d,h} \leq 1, \forall d \in D_t^{\text{n}} \quad (6)$$

$$x_{d,h} = 0, \forall h \in H, \forall d \in D_t^{\text{n}}, \text{if } p_{\text{match}}(h, t+\Delta) \cdot \frac{\Delta}{\tau\left((h_{d,t},t,0),h\right)} < \epsilon \quad (7)$$

Equation 5 presents the objective function of the repositioning. $p_{\text{match}}(h, t+\Delta)$ is the matching probability in h at time step $t+\Delta$, which is correlated to the number of unserved orders and the number of idle drivers at step $t+\Delta$. The constraint in Equation 6 ensures each idle driver in D_t^{n} will be assigned at most one repositioning instruction, and $h_{d,t}$ is the current hexagon of driver d at time step t. By setting a threshold ϵ, Equation 7 guarantees that only a repositioning instruction with a reasonable matching probability and a proper repositioning duration can be generated in our model.

Once we achieve the optimal solution X^{opt} for the problem in Equation 5, idle drivers can be partitioned into two groups $D_t^{\text{n},1}$, a set of idle drivers assigned a repositioning instruction in X^{opt}, and $D_t^{\text{n},0}$, a set of idle drivers without a repositioning instruction. To ensure every idle driver in D_t^{n} will be assigned a repositioning instruction, MDP optimal policy is considered for drivers in $D_t^{\text{n},0}$ to generate a repositioning instruction. Consequently, the final solution can optimize the immediate demand pick-up responses and the long-term operation efficiency simultaneously.

4. Solving Approaches

To solve the problem in Equation 5, if we abstract idle drivers in $D_t^{\text{n}}$and hexagonal space H as two sets of vertices, and all valid repositioning instructions as the set of edges, the above problem can be represented as a bipartite graph. As shown in Fig. 2, green vertices are the idle drivers and blue vertices represent the hexagons to be repositioned, there is a matching

probability associated with each hexagon, which is considered as the weight of the corresponding edges linked to this hexagon. Thus, we can show the BNLP is equivalent to the maximum weight matching problem in the graph.

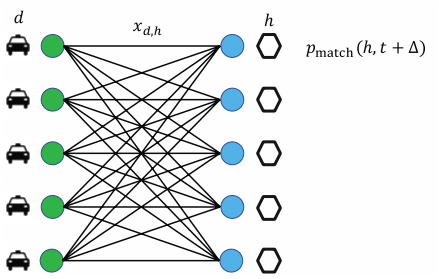


Figure 2: Graphical representation of the repositioning.

Since existing approaches (Kuhn et al. 1955) are unable to solve the nonlinear bipartite matching problem as in the graph in Fig. 2, we introduce an Iterative Edge Cutting (IEC) algorithm to achieve the optimal matching in the graph:

(1) Relax the constraints in Equations 6 and 7, and then initialize a bipartite graph that each idle driver links to all the hexagons of his action space.

(2) Remove the edges linked to the hexagons without unserved orders.

(3) Compute the matching probability on each hexagon and update the objective function.

(4) Collect all the drivers with more than one edges into set $\mathbb{D}_t^n$. Accounting for both the matching probability and the repositioning duration, the edge x_{d^*,h^*} with the following minimal criteria will be cut off.

$$d^*, h^* \leftarrow \text{argmin}_{d,h}\{p_{\text{match}}(h, t+\Delta) \cdot \frac{\Delta}{\tau\left((h_{d,t},t,0),h\right)} | \forall d \in \mathbb{D}_t^n, \text{if } x_{d,h} = 1\} \quad (8)$$

(5) Repeat steps 3 and 4 until the constraints in Equations 6 and 7 can be satisfied

5. Experiments

The real road network (6,533 nodes and 10,206 directed links) and taxi data-sets of NYC Yellow Cab are considered in the experiments. We consider the orders between 07:00 AM and 10:00 AM and initially generate 3000 vehicles randomly distributed in the road network. The order dispatching (by classical one-to-one matching method) and repositioning are determined by every 10 seconds and 30 seconds. An order will be canceled if not being matched within 90 seconds. The performance of the following benchmark repositioning methods are evaluated:

(1) **Parking**: Idle drivers park at their current location once they drop off the passengers.

(2) **Random Walk**: Idle drivers head toward a random hexagon among their neighbor hexagons.

(3) **Local Hotspot**: Idle drivers move to the hexagon with the highest demand density sequentially.

(4) **MDP Walk**: Idle drivers choose the hexagon according to the optimal policy in Equation 4.

Without any prior knowledge of the historical operation data, we employ the Random Walk as the repositioning method in the simulator to generate 14 days of training data. Subsequently, the single-agent MDP is trained from the above data by setting γ to 0.8. Further, we introduce seven new days as the test data.

Table 1: The experimental results of testing.

Methods	Avg. Response	Avg. Cancellation	Avg. Response Time	Avg. Pickup Time	Avg. Occupied rate
Parking	21026.3 (82.8%)	4101.7 (16.1%)	18.5	78.0	50.7%
Random Walk	21785.8 (85.8%)	3369.2 (13.3%)	15.6	70.4	52.7%
Local Hotspot	22469.1 (88.5%)	2741.8 (10.8%)	13.7	67.3	53.9%
MDP Walk	23373.5 (92.0%)	1815.5 (7.1%)	9.3	63.8	56.9%
Proposed method	24438.1 (96.2%)	817.3 (3.2%)	8.1	60.9	61.8%

Table 1 summarizes the numerical results of the four repositioning methods. The proposed method is designed for optimizing the operation efficiency for the central platform; not only does it achieve the best performance for the platform with 96.2% average response rate, 3.2% average cancellation rate, it also leads to an increase in passengers' service quality and drivers' working efficiency, average response time and average pick-up time of passengers are reduced to 8.1 s and 60.9 s, and average occupied rate of drivers can be increased to 61.8%. In summary, the proposed method brings remarkable improvements to both platform efficiency (serving more orders and reducing order cancellations) and users experience (reducing passenger's waiting time and increasing driver's occupied rate).

6. Summary

In this paper, we propose a novel method for the centralized E-haling platform to implement repositioning instructions to idle drivers. The proposed method is designed to optimize both the immediate demand satisfaction and the long-term operation efficiency simultaneously. To this end, a single-agent MDP model is designed to evaluate the long-term influence of the repositioning on platforms efficiency, and a binary nonlinear programming (BNLP) is also introduced for the real-time multi-driver repositioning problem, which is solved by an Iterative Edge Cutting (IEC) algorithm. The final repositioning solution is the integration of the solutions of the MDP and the BNLP. Through extensive numerical experiments based on field data of Manhattan, the proposed method improves both platform efficiency and users experience and shows a great potential to be deployed in real-world applications.

7. Reference

[1] Agatz, N., Erera, A. L., Savelsbergh, M. W., & Wang, X. (2011). Dynamic ride-sharing: A simulation study in metro Atlanta. Procedia-Social and Behavioral Sciences, 17, 532-550.
[2] Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. European Journal of Operational Research, 223(2), 295-303.
[3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.
[4] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2), 83-97.
[5] Vonolfen, S., & Affenzeller, M. (2016). Distribution of waiting time for dynamic pickup and delivery problems. Annals of Operations Research, 236(2), 359-382.