Ride-Hailing Order Matching and Vehicle Repositioning Based on Vehicle Value Function

Shun Li[1], Zeheng Zhong[1], and Bing Shi[1,2(✉)]

[1] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, China
{shunli,bingshi}@whut.edu.cn
[2] Shenzhen Research Institute, Wuhan University of Technology, Shenzhen 518000, China

Abstract. Online ride-hailing platforms, such as Uber, DiDi and Lyft, have significantly revolutionized the way of travelling and improved traffic efficiency. How to match orders with feasible vehicles and how to dispatch idle vehicles to the area with potential riding demands are two key issues for the ride-hailing platforms. However, existing works usually deal with only one of them and ignore the fact that the current matching and repositioning results may affect the supply and demand in the future since they will affect the future vehicle distributions in different zones. In this paper, we use the vehicle value function to characterize the spatio-temporal value of vehicles. At each decision-making round, we first match orders with vehicles by using bipartite graph maximum weight matching with the vehicle value function. Then we will provide idle vehicles with repositioning suggestions, where we predict the riding demand in each zone in the future, and then use a greedy strategy combined with vehicle value function to maximize social welfare. Extensive experiments based on real-world data as well the analytic synthetic data demonstrate that our method can outperform benchmark approaches in terms of the long-term social welfare and service ratio.

Keywords: Ride-hailing · Order matching · Vehicle repositioning · Value function · Demand forecasting

1 Introduction

Online ride-hailing service has played an important role in today's urban traffic system. Online ride-hailing platforms, e.g., Uber and Didi Chuxing, have greatly benefited our daily lives by allowing passengers to book a trip in advance and matching available vehicles with riding demands in real time. Order matching and vehicle repositioning are two key issues for the ride-hailing platforms, where order matching determines the matching between idle drivers (vehicles) with the riding orders, and vehicle repositioning is a strategy that can deploy idle vehicles to specific locations that are expected to generate demand in the future.

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13369, pp. 406–416, 2022.
https://doi.org/10.1007/978-3-031-10986-7_33

In this paper, we intend to address the problems of order matching and vehicle repositioning on a ride-hailing platform. Our objective is to maximize the long-term social welfare and improve the service ratio. In order to achieve these objectives, firstly, we design a vehicle value function to characterize the spatio-temporal value of vehicles. Then we transform the order matching problem into a bipartite graph maximum weight matching problem based on the vehicle value function, which is solved by using Kuhn-Munkres algorithm. Furthermore, we need to reposition idle vehicles to specific areas in order to fill the supply-demand gap in the future. We design a demand predicting method to forecast the number of orders in each area in the future. Based on the information of vehicles in the "matched" state and the "serving" state, we can know in which area these vehicle will become idle in the future. With such a demand information, we use a greedy strategy combined with the vehicle value function to maximize social welfare. The experimental results show that our approach can outperform benchmark approaches in terms of the long-term social welfare and service ratio.

The structure of this paper is as follows. In Sect. 2, we introduce the related work. In Sect. 3, we describe the basic settings of the problem. In Sect. 4, we describe the proposed algorithm in detail, and in Sect. 5, we experimentally evaluate our proposed approach. Finally, we conclude the paper in Sect. 6.

2 Related Work

For order matching/dispatching on ride-hailing platforms, extensive literature has been proposed on, such as dynamic matching [10], network flow optimization with rolling horizon [1], neural ADP for carpooling [7], and (deep) reinforcement learning [8]. For vehicle repositioning, a number of works are under the setting of grid world, and RL related methods have been proposed for this problem, e.g., DQN [2], multi-agent RL [4] and hierarchical RL [3].

To the best of our knowledge, existing works usually analyzed the order matching and repositioning problems separately, and did not consider the impacts of current decision of order matching and idle vehicle repositioning on the future matching. Therefore, in this paper we take the spatio-temporal value of vehicles and the demand information in each area into account, and consider the order matching and idle vehicle repositioning problem as a whole to maximize the long-term social welfare.

3 Problem Formulation

In this section, we first introduce how the online ride-hailing system works, and then describe the basic settings of orders and vehicles.

We assume that vehicles belong to the online ride-hailing platform, which means that vehicles are a part of the platform and the platform can determine the behavior of vehicles. Then in the ride-hailing system, there are two main roles, passengers and the platform. For the workflow of ride-hailing system, firstly, passengers submit orders to the platform according to their own demands, in

which the order information includes the maximum price they would like to pay. The platform collects the submitted order requests and then matches orders with available vehicles. After finishing order matching, the platform computes the payment of the matched orders. If there are idle vehicles left, the platform needs to provide them with repositioning suggestions to satisfy the future riding demands. The entire time is split into a set of discrete time steps $\tau = \{1, 2, ..., T\}$ with time step length Δt. In each time step, the platform will finish order matching and provide idle vehicles repositioning suggestions for this round. In the below, we give the definition of some key elements in the ride-hailing system.

Definition 1 (Order). *Order $o \in \mathcal{O}$ is defined as a tuple $(l_o^p, l_o^d, t_o^r, t_o^w)$, where l_o^p, l_o^d is the pick-up and drop-off locations of order o respectively, t_o^r means the time when the order o is raised, t_o^w is the maximum time that a passenger of order o is willing to wait for the service.*

In this paper, we assume that when the order o is not matched by the driver within t_o^w time, the order o will be automatically cancelled and the passenger is not willing to wait.

Definition 2 (Vehicle). *Vehicle $v \in \mathcal{V}$ is defined as a tuple (l_v, c_v), where l_v is the current position and c_v is the unit driving cost of vehicle v.*

Definition 3 (Social Welfare). *The long-term social welfare is the sum of the profits of the platform and passengers over the whole time step.*

$$SW = \sum_{t=1}^{T} \sum_{o \in \mathcal{O}_t^w} (val_o - C^o_{\Theta_t(o)}) \tag{1}$$

where val_o is the highest price passengers willing to pay to the platform for accessing travel service, which can be regarded as the value of this order for the passenger, $\mathcal{O}_t^w$ is the set of matched orders in round t, Θ_t is the matching result in round t, and $\Theta_t(o) = v$ denotes the matching of order o and vehicle v. $C^o_{\Theta_t(o)}$ is the cost for vehicle $\Theta_t(o)$ to complete order o.

4 Method

We design a vehicle state value function, which implies the ability of vehicles to make social welfare in different spatio-temporal states. Then, based on the vehicle value function, we design the order matching and idle vehicle repositioning algorithm.

4.1 Vehicle Value Function Design

The vehicle value function shows the potential social welfare that the vehicle can make in the future in the current spatial-temporal state. It is defined as $V(t, g, c)$, where $t \in T$ is the time step, $g \in G$ is the zone index where the vehicle is located, and c is the vehicle unit travel cost.

In a multi-round matching and repositioning process, we can capture how the current vehicle state can affect the future social welfare, i.e. the vehicle value function. This process can be considered as a sequential decision process, and thus Markov decision process(MDP) can be used to model the real-time order matching and idle vehicle repositioning process.

In the following, we define the attributes in the MDP $M = \langle S, A, P, r, \gamma \rangle$ in detail and describe the state transfer involved.

State: The state of each vehicle is defined as a tuple $s = (t, g, c) \in S$, which is the vehicle value function.

Action: The action is $a \in A = \{a_1, a_2, a_3\}$, where a_1 is to match an order with a vehicle. a_2 means that the vehicle will be stationary in a zone for a certain time, and a_3 is to reposition an idle vehicle to an adjacent zone.

Reward: The reward r is the profit of the passenger and the platform when the action is taken. The reward value r is calculated as follows:

$$r = p_o - C_{\Theta(o)} \tag{2}$$

where p_0 is the payment for an order o and $C_{\Theta(o)}$ is the cost required for the vehicle $\Theta(o)$ to complete the order o. The cumulative reward R_γ is calculated as follows:

$$R_\gamma = \sum_{t=0}^{T-1} \gamma^t \frac{r}{T} \tag{3}$$

where γ is a discount factor that decreases the impacts of the past rewards.

We obtain the vehicle value function by using value iteration. When the action performed by the agent is to match an order, the agent receives an immediate reward R_γ and performs a state transfer, and the TD update rule is:

$$V(s) \leftarrow V(s) + \alpha \left[R_\gamma + \gamma V(s') - V(s)\right] \tag{4}$$

where $s = (t, g, c)$ is the state of the vehicle at the current time step, $s' = (t + \Delta t_1, g_{l_d}, c)$ is the state of the vehicle after completing the matched order, g_{l_d} is the destination of the order, and Δt_1 is the time required to depart from the time step t to pick up the passenger and deliver the passenger to destination g_{l_d}. When the agent acts as stationary, the immediate reward of the agent is 0. The TD update rule is as follows:

$$V(s) \leftarrow V(s) + \alpha \left[0 + \gamma V(s'') - V(s)\right] \tag{5}$$

Since the agent performs a stationary action, the position of the agent does not change, i.e., $s'' = (t + 1, g, c)$. When the action performed by the agent is idle vehicle repositioning, we construct a virtual order where the payment is 0, the origin of the order is g, and the destination of the order is one of the neighboring zones of g. The TD update rule is as follows:

$$V(s) \leftarrow V(s) + \alpha \left[R_\gamma{}' + \gamma V(s''') - V(s)\right] \tag{6}$$

where $R_\gamma{}'$ is calculated by Eq. 3, and the payment for the corresponding order is 0. $s''' = (t + \Delta t_2, g', c)$ is the state of the vehicle after the repositioning is completed. and $g' \in g_{near}$ is a neighboring zone of g.

The platform first collects historical state transfer data, and then uses a dynamic programming based value iteration algorithm to backward recursively calculate the value $V(s_i)$ in each state to obtain the vehicle value function $V(s)$. The details of the algorithm are shown in Algorithm 1.

Algorithm 1: Dynamic Programming based Value Iteration Algorithm (DPVI)

Data: History state transfer tuple $D = \{(s_i, a_i, r_i, s_i{}')\}$, where each state $s_i = (t_i, g_i, c)$ consists of the vehicle's time step, geographic location index and cost

Result: Vehicle value function V

for $t = T - 1$ **to** 0 **do**
  $D_t \leftarrow \{(s_i, a_i, r_i, s_i{}')|\forall s_i = (t_i, g_i, c), t_i = t\}$;
  foreach $(s_i, a_I, r_i, s_i{}') \in D_t$ **do**
    $N(s_i) \leftarrow N(s_i) + 1$;
    $V(s_i) \leftarrow V(s_i) + \frac{1}{N(s_i)}\left(\gamma^{\Delta t(a_i)} V(s_i{}') + R_\gamma(a_i) - V(s_i)\right)$;
  end
end
return V

4.2 Order Matching and Vehicle Repositioning Algorithm

In this section we introduce the Value Function and Demand based Order Matching and Vehicle Repositioning algorithm (VFDOMVR). Overall speaking, we first complete the order matching, and then perform the demand forecasting. We then complete vehicle repositioning according to the vehicle value function and the results of demand forecasting. This algorithm is described in Algorithm 2.

Order Matching: Given the known order information and vehicle information, this problem can be transformed into a maximum weight bipartite graph matching problem (lines 2–4). Note that only edges with weight greater than 0 are added to the bipartite graph (lines 7–9). We use Kuhn-Munkres (KM) [6] algorithm to get the result of order matching (line 11).

Demand Forecasting: If we only use the value function to reposition the idle vehicles, it is likely that an excessive number of idle vehicles will be repositioned to a higher value area. Therefore, we use the SARIMAX model to predict how many vehicles will be needed in a certain area at a certain time (line 12). Through demand forecasting, we can provide guidance for future idle vehicle repositioning.

Vehicle Repositioning: We first iterate through all the vehicles in the "matched" and "serving" states to know which area these vehicles will become idle again in the future. The number of vehicles dispatched in the area should be

reduced accordingly (lines 13–17). Next, we iterate through all idle vehicles, and during the iteration we try to give repositioning suggestions for each idle vehicle (lines 18–29). We judge whether the demand for the area to be dispatched will be exceeded if this repositioning option is executed (lines 21–23). If it does not exceed, it is determined to reposition the vehicle to the area with the greatest gains, and the demand for idle vehicles at the corresponding time of the area is reduced by 1 (lines 24–28). Finally, we update the related system information (line 30).

Algorithm 2: Value Function and Demand based Order Matching Vehicle Repositioning Algorithm (VFDOMVR)

Data: The set $\mathcal{O}$ of orders, the set of $\mathcal{V}$ of vehicles, and the vehicle state value function V

Result: The set of matched orders $\mathcal{O}^w$, matching result Θ, repositioning result $\mathcal{R}$, and the social welfare results SW

1 **for** $t = 0$ **to** T **do**
2 $\quad \mathcal{O}_t \leftarrow \{o \in \mathcal{O} | t_o^r + t_o^w \leq t + \Delta t\}$;
3 $\quad \mathcal{V}_t \leftarrow select\_empty\_vehicles(\mathcal{V}, t)$;
4 $\quad$ Initialize the bipartite graph $G = (\mathcal{O}_t, \mathcal{V}_t, E)$;
5 $\quad$ **foreach** $\langle o, v \rangle \in \mathcal{O}_t \times \mathcal{V}_t$ **do**
6 $\quad\quad$ calculate the vehicle state value difference ΔV corresponding to $\langle o, v \rangle$;
7 $\quad\quad$ **if** $\Delta V > 0$ **then**
8 $\quad\quad\quad$ assign the weights of edges $\langle o, v \rangle$ to ΔV and insert into the bipartite graph;
9 $\quad\quad$ **end**
10 $\quad$ **end**
11 $\quad \mathcal{O}_t, \Theta_t \leftarrow KM(G)$;
12 $\quad \mathcal{D}_t \leftarrow Demand\_Forecasting(\mathcal{O}, \mathcal{V}, t)$;
13 $\quad \mathcal{V}' \leftarrow select\_matched\_and\_servicing\_state\_vehicles(\mathcal{V}, t, \Theta_t)$;
14 $\quad$ **foreach** $v \in \mathcal{V}'$ **do**
15 $\quad\quad t', l' \leftarrow become\_idle\_again(v)$;
16 $\quad\quad \mathcal{D}_t(t', l') \leftarrow \mathcal{D}_t(t', l') - 1$;
17 $\quad$ **end**
18 $\quad$ **foreach** $v \in \mathcal{V}_t$ *and* $v \notin \mathcal{V}'$ **do**
19 $\quad\quad$ **foreach** $g \in g_{near}$ **do**
20 $\quad\quad\quad$ calculate the time t' of the vehicle reaching the neighboring zone g;
21 $\quad\quad\quad$ **if** $\mathcal{D}_t(t', g) \leq 0$ **then**
22 $\quad\quad\quad\quad$ Continue;
23 $\quad\quad\quad$ **end**
24 $\quad\quad\quad$ calculate the difference $\Delta V'$ of the state value of the vehicle v from the current position to the neighboring zone g;
25 $\quad\quad\quad$ insert $t', g, \Delta V'$ into an Array A;
26 $\quad\quad$ **end**
27 $\quad\quad t'', g', \mathcal{R}_t(v) \leftarrow \arg\max A$;
28 $\quad\quad D_t(t'', g') \leftarrow D_t(t'', g') - 1$;
29 $\quad$ **end**
30 $\quad$ update $SW_t, SW, \mathcal{O}^w, \Theta_t, \Theta, \mathcal{R}$;
31 **end**
32 **return** $\mathcal{O}^w, \Theta, \mathcal{R}, SW$

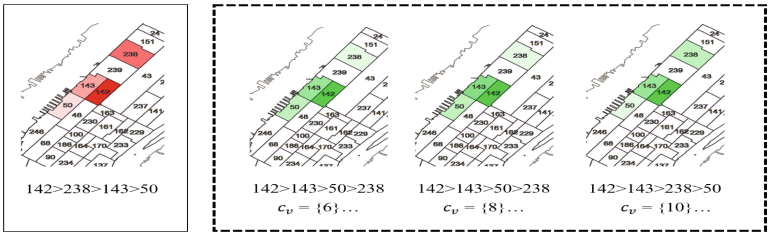
5 Experiment

In this section, we run experiments to evaluate the proposed algorithm based on the data from New York City Taxi and Limousine Comission (TLC)[1]. We use Manhattan taxi zone map provided by TLC as the map data, and collect the order data from 19:00 to 21:00 in the weekday within this area. Because we cannot find the fuel consumption of New York taxies, so we collect the related information from China Automobile Fuel Consumption Query System[2] instead. And finally we remove some unreasonable orders (e.g. orders with invalid fares, orders in isolated zones and zero trip milage).

For the experimental parameters, the length of each time step is set as 60 s. The maximum waiting time for passengers is chosen randomly from {3 min, 4 min, 5 min, 6 min, 7 min, 8 min}. The average vehicle travel speed V_{avg} is set to 7.2 mph, and for each vehicle, the unit travel cost is randomly selected from $\{6, 8, 10\} \times 2.5/6.8/1.6$\$/km. The initial location of the vehicle is randomly selected in Manhattan taxi zone map. In the experiments, we try different numbers of vehicles, which are {1500, 2000, 2500, 3000, 3500}. For each experiment, we repeat it for 10 times and then compute the average result.

5.1 Evaluation of Vehicle Value Function

The vehicle value function is the basis for the order matching and repositioning algorithm, and therefore we first analyze the effectiveness of this value function, which is shown in Fig. 1.



(a) Demand situation (b) Vehicle state values at different driving costs

Fig. 1. Effectiveness analysis of vehicle value function

The number in each block is the index of zones, where zone 142, 143 and 50 are adjacent to each other on the road network. In Fig. 1(a), the darker the color means that there are more orders in the current zone. Figure 1(b) shows the vehicle value under the vehicle travel cost $c_v = \{6, 8, 10\} \times 2.5/6.8/1.6$ respectively. The

[1] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.
[2] https://yhgscx.miit.gov.cn/fuel-consumption-web/mainPage.

darker the color indicates the higher the value of the vehicle in the current zone. Comparing Fig. 1(a) with Fig. 1(b), we can find that the riding demand situation of the zone is consistent with the vehicle value, i.e., when there are more riding orders in that zone, vehicle value in that zone is also higher.

Furthermore, we also find that although zone 238 has greater riding orders than zone 50 and 143, vehicles in zone 50 and 143 can travel to neighboring zone 142 to pick up orders, and therefore the vehicle value is higher in zone 50 and 143. We also find that as the vehicle travel cost increases, the vehicle value of zone 50 decreases. Especially when the vehicle travel cost is $c_v = 10 \times 2.5/6.8/1.6$, the vehicle value of zone 50 is lower than that of zone 238. This is because when the unit travel cost increases, the additional cost of travelling to the neighboring zone increases, which results in the loss of social welfare. This may imply that vehicles with high cost should try to serve nearby orders.

In summary, from the above analysis we can conclude that the vehicle value function can reflect the zone demand situation to a certain extent, and can reflect the ability of vehicles with different driving costs to obtain social welfare in different zones, and thus can be used in the order matching and idle vehicle repositioning to increase the social welfare.

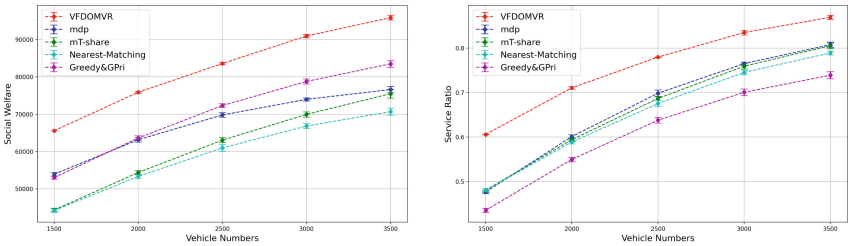
5.2 Evaluation of Order Matching and Vehicle Repositioning Algorithm

In this section, we evaluate the proposed algorithm against **mdp** [9], **mT-share** [5], **Greedy&Gpri** [11] and **Nearest-Matching** in terms of social welfare and service ratio (the ratio of the number of matched orders to the total number of orders submitted by passengers).

The experimental results are shown in Fig. 2. From Fig. 2(a), we find that as the number of vehicles increases, the social welfare increases since more orders are served. We also find that **VFDOMVR** algorithm achieves the highest social welfare.

We also look into the service ratio in Fig. 2(b). We find that the **VFDOMVR** algorithm achieves the maximum service ratio. This means the **VFDOMVR** algorithm can help the platform serve more orders.

In summary, it can be found that our **VFDOMVR** algorithm can make the highest social welfare through the above experiments. What's more, it also provides higher service ratio. Therefore, we believe that the **VFDOMVR** algorithm can help the online ride-hailing platform to make more social welfare and serve more orders.



(a) Social welfare (b) Service ratio

Fig. 2. Results of order matching and vehicle repositioning

5.3 Vehicle Repositioning Analysis

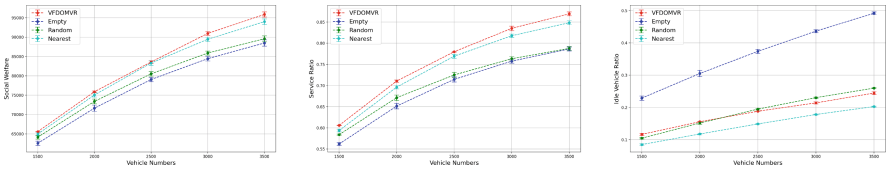
In this section, we further analyze the effectiveness of the idle vehicle repositioning algorithm. In order to evaluate the performance of the idle vehicle repositioning, we combine different benchmark repositioning algorithms such as **Empty**, **Random** and **Nearest** with the order matching of **VFDOMVR** to generate the benchmark algorithms. And the metrics are social welfare, service ratio and idle vehicle ratio (the ratio of the number of idle vehicles to the total number of vehicles).

The experimental results are shown in Fig. 3. From Fig. 3(a), we find that **VFDOMVR** algorithm achieves the largest social welfare. As the number of vehicles increases, the social welfare obtained by all algorithms increases.

From Fig. 3(b), we find that **VFDOMVR** algorithm achieves the maximum service ratio. This means that after using the proposed repositioning algorithm, the platform can serve more orders, and thus can achieve the maximum social welfare.

From Fig. 3(c), we can find that the **Nearest** dispatching algorithm achieves the minimum idle vehicle ratio, followed by **VFDOMVR** algorithm, **Random** dispatching algorithm and **Empty** dispatching algorithm. Among them, the results of **Nearest** dispatching algorithm, **VFDOMVR** algorithm and **Random** dispatching algorithm are close, while the results of **Empty** dispatching algorithm are obviously much higher. It indicates that our repositioning algorithm can effectively improve the utilization of idle vehicles.

In summary, we find that the repositioning of idle vehicles in the **VFDOMVR** algorithm can utilize idle vehicles and satisfy more order requests, thus improving social welfare. This is mainly because the repositioning of idle vehicles in the **VFDOMVR** algorithm takes the spatio-temporal value of vehicles into account and dispatches vehicles to zones where more vehicles are needed, and thus can increase service ratio and the social welfare.



(a) Social welfare (b) Service ratio (c) Idle vehicle ratio

Fig. 3. Results of vehicle repositioning

6 Conclusion

In this paper, we focus on the problem of real-time order matching and idle vehicle repositioning to maximize the long-term social welfare. We need to consider the impacts of current decision on the future rounds. In more detail, we consider the spatio-temporal value of vehicle, and use the vehicle value function to transform order matching into a maximum weight bipartite graph matching problem. For idle vehicle repositioning, we use vehicle value function to learn which area is more profitable to move to, and use demand forecasting to avoid too many vehicles being moved to the same area. In order to verify the effectiveness of the proposed algorithm, we further carry out experimental analysis based on the taxi data in Manhattan, and evaluate the **VFDOMVR** algorithm against some typical benchmark algorithms. The results show that the **VFDOMVR** algorithm can help online ride-hailing platforms to dispatch idle vehicles efficiently, improve the utilization of idle vehicles, and increase the service ratio and social welfare.

Acknowledgement. This paper was funded by the Shenzhen Fundamental Research Program (Grant No. JCYJ20190809175613332), the Humanity and Social Science Youth Research Foundation of Ministry of Education (Grant No. 19YJC790111), the Philosophy and Social Science Post-Foundation of Ministry of Education (Grant No.18JHQ0 60) and the Fundamental Research Funds for the Central Universities (WUT: 202 2IVB004).

References

1. Bertsimas, D., Jaillet, P., Martin, S.: Online vehicle routing: the edge of optimization in large-scale applications. Oper. Res. **67**(1), 143–162 (2019)
2. Holler, J., et al.: Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 1090–1095. IEEE (2019)
3. Jin, J., et al.: CoRide: joint order dispatching and fleet management for multi-scale ride-hailing platforms. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1983–1992 (2019)
4. Lin, K., Zhao, R., Xu, Z., Zhou, J.: 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, pp. 1774–1783 (2018)

5. Liu, Z., Gong, Z., Li, J., Wu, K.: Mobility-aware dynamic taxi ridesharing. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 961–972. IEEE (2020)
6. Munkres, J.: Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**(1), 32–38 (1957)
7. Shah, S., Lowalekar, M., Varakantham, P.: Neural approximate dynamic programming for on-demand ride-pooling. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 507–515 (2020)
8. Tang, X., et al.: 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, pp. 1780–1790 (2019)
9. Xu, Z., et al.: 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, pp. 905–913 (2018)
10. Yan, C., Zhu, H., Korolko, N., Woodard, D.: Dynamic pricing and matching in ride-hailing platforms. Nav. Res. Logist. (NRL) **67**(8), 705–724 (2020)
11. Zheng, L., Cheng, P., Chen, L.: Auction-based order dispatch and pricing in ridesharing. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 1034–1045. IEEE (2019)