A Deep Reinforcement Learning Based Dynamic Pricing Algorithm in Ride-Hailing

Bing Shi[1,2](✉), Zhi Cao[1], and Yikai Luo[1]

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

Abstract. Online ride-hailing has become one of the most important transportation ways in the modern city. In the ride-hailing system, the vehicle supply and riding demand is different in different regions, and thus the passengers' willingness to take a riding service will change dynamically. Traditional pricing strategies cannot make reasonable decisions to set the riding prices with respect to the dynamical supply and demand in different regions, and they cannot make adaptive responses to the real-time unbalanced supply and demand. In addition, the ride-hailing platform usually intends to maximize the long-term profit. In this paper, we use deep reinforcement learning to design a multi-region dynamic pricing algorithm to set the differentiate unit price for different regions in order to maximize the long-term profit of the platform. Specifically, we divide the ride-hailing area into several non-overlapping regions, and then propose a model to characterize the passenger's price acceptance probability. We further model the pricing issue as a Markov decision-making process, and then use deep reinforcement learning to design a multi-region dynamic pricing algorithm (**MRDP**) to maximize the platform's long-term profit. We further run extensive experiments based on realistic data to evaluate the effectiveness of the proposed algorithm against some typical benchmark approaches. The experimental results show that **MRDP** can set the price effectively based on supply and demand to make more profit and can balance the supply and demand to some extent.

Keywords: Ride-hailing · Dynamic pricing · Supply and demand · Long-term profit · Reinforcement learning

1 Introduction

Nowadays, various online ride-hailing platforms have emerged, such as DiDi and Uber. In the ride-hailing system, how to set the riding service price is one of the most important issues. It can affect the choice of passengers, thereby affecting the platform's profit and the supply and demand in the future. Currently, the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13246, pp. 489–505, 2022.
https://doi.org/10.1007/978-3-031-00126-0_36

traditional ride-hailing pricing algorithm is based on the combination of basic price and unit price. The online ride-hailing platform set the price by taking into account the order completion time and other factors. These ride-hailing platforms use dynamic pricing to control supply and demand at different locations and times. However, these pricing strategies usually use a uniform way to set the service price for the whole area, and do not consider the supply and demand diversity in different sub-regions. Furthermore, the supply and demand dynamically changes over the time. The current pricing strategy may not be able to adapt the changed supply and demand. Therefore, the ride-hailing system needs to design an effective pricing strategy to set the price differently according to the real-time supply and demand in different sub-regions in order to maximize the long-term profit.

In the ride-hailing market, the supply and demand in different regions are different. Therefore, we first divide the entire area into several non-overlapping sub-regions. The multi-region status reflect the difference between supply and demand more precisely. Secondly, the supply and demand in the same region are dynamically changing. Passengers may have dynamic acceptance probability of the service price. For example, during the peak time, passengers may have a stronger riding demand and are more likely to accept higher price. In contrast, during the off peak time, higher price may cause passengers to give up their ride plans, which will damage the platform's profit. Therefore, the platform should adopt a reasonable dynamic pricing strategy according to different supply and demand. Finally, the platform needs to maximize the long-term profit, rather than the short-term profit.

Although there exist some related works about pricing in the ride-hailing [8,9,13], they did not consider all the above factors. In this paper, we design a dynamic pricing algorithm based on the real-time supply and demand states of different regions to maximize the long-term profit of the platform. The main contributions of this paper are as follows. Firstly, we divide the whole ride-hailing area into several non-overlapping sub-regions, and then collect the supply and demand states in different regions for pricing. Secondly, considering the dynamic change of supply and demand status and passengers' willingness to take a riding service, we propose a multi-region dynamic pricing problem. Since this problem is a sequential decision problem, we model it as a Markov decision process and then propose a multi-region dynamic pricing algorithm (**MRDP**) based on deep reinforcement learning. The algorithm maximizes the platform's long-term profit by taking into account passengers' acceptance probability of prices based on the real-time spatial and temporal distribution information. Finally, we run experiments based on Chengdu ride-hailing order data to evaluate the proposed algorithm. The result shows that **MRDP** can balance the supply and demand, and can make reasonable pricing based on the real-time supply and demand status, which will bring higher profit to the platform.

The rest of this paper is structured as follows. In Sect. 2 we will introduce the related work, in Sect. 3 we will introduce the basic settings of this paper, in

Sect. 4 we will introduce the dynamic pricing algorithm designed in this paper. Finally we will give experimental analysis and summary in Sects. 5 and 6.

2 Related Work

There exist a number of works about pricing in the ride-hailing. In the dynamic pricing of online ride-hailing, Gan et al. [8] proposed a pricing method to incentivize drivers in order to solve the issue that most taxi drivers deliberately avoid providing riding service in the peak time. Chen et al. [5] proposed a dynamic pricing strategy in the intelligent transportation platforms to minimize traffic congestion. Asghari et al. [3] adjusted the prices by considering the future demand of the road network to increase the platform's profit while lowering the prices. However, they did not consider the impact of pricing on the future, and could not maximize the platform's long-term profit. Chen et al. [6] proposed a pricing method for the ride-sharing system, which provides passengers with more choices. Chen et al. [4] proposed a joint framework to optimize both pricing and matching strategies at the same time. However, they only considered discrete pricing action.

There also exist some works on combining mechanism design with pricing strategies. Most of these works focused on preventing strategic behaviors of participants that lead to loss of platform profit. Asghari et al. [1] studied the real-time ride sharing problem and designed a framework based on distributed auctions to maximize the platform's profit without affecting the quality of service. Then Asghari et al. [2] proposed a pricing model based on Vickery auctions to ensure drivers bidding truthfully without sacrificing any profit. Zhang et al. [15] designed a discounted trade reduction mechanism for dynamic pricing in the ride-sharing system. Zheng et al. [16] considered the order price and proposed a constrained optimization problem, which takes the profit of the platform as the optimization goal and controls the detour distance and waiting time of the driver. In order to realize self-motivated bonus bidding of users, Zheng et al. [17] designed an auction mechanism that requires users to submit the information truthfully.

To the best of our knowledge, existing works did not consider the difference of supply and demand in different regions, and did not consider the dynamic changes of supply and demand over the time and the dynamic acceptance probability of the riding price. In this paper, we consider the above factors to design a multi-region dynamic pricing algorithm to maximize the profit of the platform.

3 Basic Settings

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

Figure 1 show how the online ride-hailing system works. First, the passenger rises the riding demand, and then the platform sets the price for the riding service. The passenger decides whether accepting the service price or not. If so, the riding demand becomes an order, and then the platform matches the order with the idle vehicles.

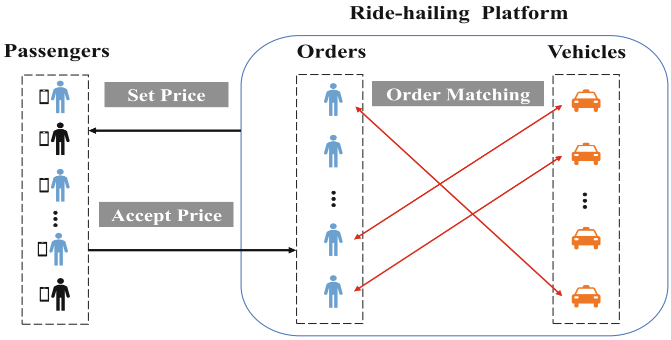


Fig. 1. A ride-hailing system

3.1 Symbols and Definitions

In this paper, we divide the entire online ride-hailing time period into several time steps $\mathcal{T} = \{1, 2, \cdots, T\}$ with the length of each time step denoted as Δt. The relevant definitions are given below.

The entire online ride-hailing area is divided into non-overlapping and interconnected rectangular regions, numbered as 1, ..., N. A riding demand $r_i \in R$ is expressed as a tuple $r_i = \left(l^s_{r_i}, l^e_{r_i}, g_{r_i}, t_{r_i}, val_{r_i}\right)$, $l^s_{r_i}$ and $l^e_{r_i}$ represents the pick-up and drop-off location of r_i respectively, g_{r_i} represents the pick-up region, t_{r_i} is the time when r_i is submitted, val_{r_i} is the maximum unit price that the passenger can accept for this trip. In the following, passenger r_i and riding demand r_i have the same meaning.

In addition, we set f_{r_i} as a status identifier, where $f_{r_i} = 0$ means r_i does not accept the riding price provided by the platform, $f_{r_i} = 1$ means that r_i accepts the riding price, $f_{r_i} = 2$ means that the platform has arranged a vehicle to serve the riding demand. val_{r_i} is private to the passenger and cannot be obtained directly by the platform. We assume that val_{r_i} is uniformly distributed within $[p_{\min}, p_{\max}]$ [11]. The unit price in region g is p_g, and the probability that the passenger r_i in region g accepts the platform price p_g is $G_{r_i}(p_g) = 1 - F_{r_i}(p_g)$, where $F_{r_i}(p_g) = P(val_{r_i} \leq p_g)$ is the cumulative distribution function [3,11,13].

Definition 1 (Order). *When the passenger r_i accepts the platform pricing, the riding demand is converted into an order. An order $o_i \in O$ is expressed as $o_i = \left(r_i, t^w_{r_i}, pay_{r_i}\right)$, where r_i is the riding demand information $\left(l^s_{r_i}, l^e_{r_i}, g_{r_i}, t_{r_i}, val_{r_i}\right)$, $t^w_{r_i}$ indicates the maximum waiting time for r_i, pay_{r_i} is the price of the order.*

The price for each order is the unit price multiplied by the shortest travelling distance, that is, $pay_{r_i} = p_{g_{r_i}} \times \mathrm{dis}(l^s_{r^s_i}, l^e_{r_i})$, $\mathrm{dis}(l^s_{r^s_i}, l^e_{r_i})$ is the shortest distance between the pick-up and drop-off location of the order.

Definition 2 (Vehicle). *The vehicle $v_i \in V$ is represented as a four-tuple $v_i = (g_{v_i}, l_{v_i}, d_{v_i}, s_{v_i})$, g_{v_i} represents the current region of the vehicle, l_{v_i} indicates the current location of the vehicle, d_{v_i} is the driver's fuel consumption cost.*

We also set an identifier s_{v_i} for vehicle status, where $s_{v_i} = 1$ means that the vehicle is serving other orders, $s_{v_i} = 0$ means that the vehicle is idle. In this paper, the vehicle belongs to the platform, that is, the vehicle will not have spontaneous behavior, which makes it easier for the platform to manage and dispatch, thus improving the efficiency of the platform [14].

Definition 3 (Platform profit). *As we assume that the vehicles are managed by the platform, the profit of the platform is the amount paid by all passengers who have been served minus the cost of the vehicle during the entire time period.*

$$EP = \sum_{\mathcal{T}=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{M^i_{\mathcal{T}}} G\left(p^{\mathcal{T}}_{g_i}\right) \mathbb{I}\left(f_{r_j} = 2\right) \left(p^{\mathcal{T}}_{g_i} \times \mathrm{dis}\left(l^s_{r_j}, l^e_{r_j}\right) - C_{r_j}\right) \tag{1}$$

Where $M^i_{\mathcal{T}}$ is the total number of riding demand in the i-th region at time $\mathcal{T}$, $p^{\mathcal{T}}_{g_i}$ is the unit price in region g_i at $\mathcal{T}$-th time step, and C_{r_j} is the cost of vehicle serving for the riding demand r_j.

3.2 Problem Formulation

Similar to the existing work [16,17], we consider the ride-hailing process in the multiple time steps.

In reality, the platform does not know the acceptance probability of passengers in advance. The acceptance probability should be estimated. Secondly, only the passengers who accept the price will contribute to the total profit of the platform. The decisions of passengers are unknown before the platform determines the unit price, and therefore the platform needs to estimate the expectation of passengers on the unit price. Then the pricing will affect the future vehicle supply, and the supply and demand in turn determines the unit price in each region. Therefore the affections between regions should also be considered. Finally the platform always wishes to find a set of optimal unit price to maximize its total profit. The goal stated above can be formalized to the following problem.

Definition 4 (Multi-region dynamic pricing problem). *In a period of time, given the real-time riding demand set R_t and the idle vehicle set V_t in a continuous time slot, the platform will set the unit price $P_t = \left(p^1_t, p^2_t, \ldots, p^N_t\right)$ for each region at each time step according to the real-time supply and demand status to maximize the platform's profit EP over the entire time period.*

4 The Algorithm

We first divide the ride-hailing market into N region, and the platform performs pricing and matching in each region. In this section, we introduce the pricing and matching algorithms.

4.1 Multi-region Dynamic Pricing Algorithm

At each time step, the platform sets unit price for each region based on the dynamic supply and demand information. Pricing will affect the matching results of the platform and thus affect the future supply and demand in different regions. Therefore, this multi-region dynamic pricing problem is a sequential decision-making problem. We model it as a Markov decision process, and then use reinforcement learning to address it.

This Markov decision process can be described as a tuple (S, A, P, r, γ), where S is a set of states, A is a set of actions, P is a transition probability function, r is the immediate reward and γ is a discount factor that decreases the impact of the past reward. We now describe them in details.

State: $s_t = (v_t, c_t, a_{t-1}, m_{t-1}, e_{t-1}) \in S$, where $v_t = \left(v_t^1, v_t^2, \ldots, v_t^N\right)$ is the number of idle vehicles in each region at time step t, $c_t = \left(c_t^1, c_t^2, \ldots, c_t^N\right)$ is the number of riding demands in each region, $a_{t-1} = \left(p_{t-1}^1, p_{t-1}^2, \ldots, p_{t-1}^N\right)$ is the unit price of each region in the last time step, $m_{t-1} = \left(m_{t-1}^1, m_{t-1}^2, \ldots, m_{t-1}^N\right)$ is the number of orders successfully matched in each region in the last time step, $e_{t-1} = \left(e_{t-1}^1, e_{t-1}^2, \ldots, e_{t-1}^N\right)$ is the profit made in each region in the last time step.

Action: $a_t = \left(p_t^1, p_t^2, \ldots p_t^N\right) \in A$ where p_t^i is the unit price set in region i at time step t.

Reward: $r_t\left(s_t, a_t\right) = \mu_1 P_t + \mu_2 ratio$, where μ_1 and μ_2 are weights, $P_t = \sum_{i=1}^{n} pay_t^i$ is the platform's profit and $ratio$ is the ratio of the actual number of orders being served to the maximum possible number of orders being served, which is:

$$ratio = \begin{cases} 0 & \text{if } v_t^i = 0 \text{ or } c_t^i = 0 \\ \frac{1}{N} \sum_{i=0}^{N} \frac{m_t^i}{\min\left(v_t^i, c_t^i\right)} & \text{if } v_t^i \neq 0 \text{ and } c_t^i \neq 0 \end{cases} \tag{2}$$

So $r_t\left(s_t, a_t\right) = \mu_1 P_t + \mu_2 ratio$, where μ_1 and μ_2 are weighting factors.

The reason of considering $ratio$ is as follows. When only considering to maximize the platform's profit, it may happen that the platform increases the price (which causes less served orders) to make more profit. Therefore, we take the $ratio$ into account in the reward, to make more the full utilization of idle vehicles.

The pricing action can be regarded as a continuous action. Therefore, we use DDPG (deep deterministic policy gradient) to design a multi-region dynamic pricing algorithm (**MRDP**), which is shown in Algorithm 1:

Algorithm 1 takes the riding demand spatio-temporal information and initial vehicle distribution information as input. A total of κ rounds of training (line 4) will be experienced. In each training, there will be T time steps (line 6). When the

Algorithm 1. Multi-region Dynamic Pricing Algorithm (**MRDP**)

Input:
Riding demand spatio-temporal distribution, initial vehicle distribution
Output:
Platform's pricing strategy π

```
1: Initialize the experience pool 𝒟 of the ride-hailing platform;
2: Initialize the Critic network 𝒬(s, a | θ^𝒬) and Actor network μ(s | θ^μ);
3: Initialize the target network 𝒬′, μ′, and set the weight θ^{𝒬′} ← θ^𝒬, θ^{μ′} ← θ^μ;
4: for κ = 1 to K do
5:    Initialize the initial state s_1 and the random noise parameter 𝒩;
6:    for t = 1 to T do
7:       The platform observes the state s_t and selects the action a_t = μ(s_t | θ^μ) + 𝒩_t
         and executes it to get rewards r_t and transfer to the next state s_{t+1};
8:       Store the state transition tuple (s_t, a_t, r_t, s_{t+1}) into 𝒟;
9:       Randomly select a set of samples (s_χ, a_χ, r_χ, s_{χ+1}) from 𝒟 for training;
10:      Set y_χ = r_χ + γ𝒬′(s_{χ+1}, μ′(s_{χ+1} | θ^{μ′}) | θ^{𝒬′});
11:      Update Critic by minimizing the loss ℒ = (1/x) Σ_x (𝒴_χ − 𝒬(s_χ, a_χ | θ^𝒬))²;
12:      Use the sample policy gradient to update the Actor policy:
         ∇_{θ^μ} J ≈ (1/X) Σ_χ ∇_a Q(s, a | θ²)|_{s=s_χ, a=μ(s_χ)} ∇_{θ^μ} μ(s | θ^μ)|_{s=s_χ};
13:      Update the target network: θ^{𝒬′} ← υθ^𝒬 + (1 − υ)θ^{𝒬′}, θ^{μ′} ← υθ^μ + (1 − υ)θ^{μ′};
14:      Set s_t = s_{t+1};
15:   end for
16: end for
```

platform observes the state s_t, it will add noise to the currently learned strategy as the unit price of each region for the next time step. After the matching is completed, we can calculate the real-time reward r_t of the action, and move to the next state s_{t+1} (line 7), and then store this information in the memory bank $\mathcal{D}$ as training data. We will take a certain number of samples from the memory bank for training each time. After continuous repeated learning and training, a convergent action strategy can be obtained. In the experiment, the experience pool stores the interaction information between the platform and passengers at each time step. **MRDP** can continuously train through the data information collected in the experience pool to learn the passengers acceptance probability under different supply and demand conditions. We treat the platform as an agent. Since the platform can obtain all the information at each time step, the agent will also consider the affections between regions and the impact of current pricing on the future supply and demand. Therefore, we can solve the difficulties mentioned above by using **MRDP**.

4.2 Order Matching Algorithm

In this paper, the matching between passengers and vehicles can be modeled as a bipartite graph maximum weighted matching problem, in order to maximize the platform's profit, i.e.

Algorithm 2. Order Matching Algorithm

Input:
The order set O_t and the idle vehicle set V_t of each region in each time step
Output:
Matching result M_t^r, platform profit P_t^{max}

1: Initialize $M_t^r \leftarrow \emptyset$;
2: **for** $g = 1$ to N **do**;
3: Initialize the bipartite graph G_g;
4: **for** $<o_i, v_j> \in O_t^g \times V_t^g$ **do**
5: **if** $current\_time - t_{o_i} \leq t_i^w$ **then**
6: $w_{ij} = p_{g_{o_i}} \times \text{dis}\left(l_{o_i}^s l_{o_i}^e\right) - d_{v_j} \times \left(\text{dis}\left(l_{o_i}^s l_{o_i}^e\right) + \text{dis}\left(l_{v_j}, l_{o_i}^s\right)\right)$;
7: **if** $w_{ij} \geq 0$ **then**
8: Set the weight of edge $<o_i, v_j>$ to w_{ij} and add this edge to G_g;
9: **end if**
10: **end if**
11: **end for**
12: $M_t^g, P_t^g \leftarrow KM\left(G_g\right), M_t^r \leftarrow M_t^r \cup M_t^g$;
13: **for** $<o_i, v_j> \in M_t^g$ **do**
14: Update $f_{o_i} = 2$ and $s_{v_j} = 1$
15: **end for**
16: **end for**
17: return $M_t^r, \sum_{g=1}^N P_t^g$;

$$\underset{x_{ij}}{\text{argmax}} \sum_{i=1}^{P} \sum_{j=1}^{Q} x_{ij} w_{ij}, \quad \text{s.t. } \forall i, \sum_{i=0}^{P} x_{ij} \leq 1; \forall j, \sum_{j=0}^{Q} x_{ij} \leq 1$$

$$x_{ij} = \begin{cases} 1 & \text{if order } r_i \text{ is matched with vehicle } v_j \\ 0 & \text{if order } r_i \text{ is not matched with vehicle } v_j \end{cases}$$

where P is the number of orders and Q is the number of idle vehicles, w_{ij} is the profit made in the matching of order r_i with vehicle v_j, and thus $w_{ij} = pay_{r_i} - d_{v_j} \times (\text{dis}(l_{r_i}^s, l_{r_i}^e) + \text{dis}(l_{v_j}, l_{r_i}^s))$. We can use Kuhn-Munkre algorithm [10] to solve this bipartite graph maximum weighted matching problem, which is shown in Algorithm 2.

In Algorithm 2, the platform will collect all the riding demand information of each region in the time slot t. For each riding demand r in the region g, if $f_r = 1$, the riding demand will be added to the order set O_t^g, and if $s_v = 0$, the vehicle will be added to the idle vehicle set V_t^g. At the beginning, the bipartite graph is initialized for each region, and then for each matching pair $<o_i, v_j>$ in the region, we will determine whether the current time exceeds the maximum waiting time of order o_i (line 5), and then calculate the platform profit when matching o_i with v_j (line 6). If the profit is non-negative, set the weight of $<o_i, v_j>$ and add to the bipartite graph. Finally, the Kuhn-Munkres algorithm [10] is used to obtain the optimal solution of the bipartite graph to maximize the platform profit in the current time slot (line 12), where M_t^g and P_t^g is the matching result and the profit in region g. The platform will update the status information of vehicles and orders (lines 13–15).

5 Experimental Analysis

In this section, we run experiments to evaluate the proposed algorithm based on the Chengdu Didi ride-hailing data[1]. We select the map near the downtown of Chengdu for the experiment, and collect the order data from 13:00 to 17:00 within this area, including the demand initiation time, starting location and travel distance. After excluding abnormal order data, a total of 31283 demand data were obtained. The heat map of riding demand during this period is shown in Fig. 2. It can be seen that the number of riding demands in the central area is large.

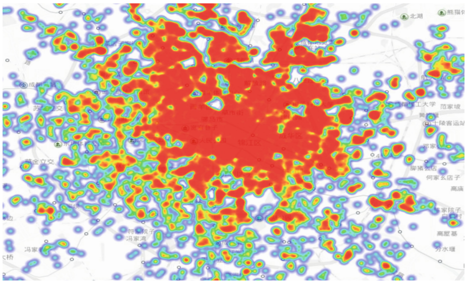


Fig. 2. User demand heat map

Firstly, a rectangular area is selected on the map with the longitude range from 104.03° E to 104.12° E and the latitude range from 30.62° N to 30.71° N. Then the area is divided into 4×4 rectangular regions. We set the length of each time step as 60 s, and vehicles are randomly distributed over 16 regions in the beginning. For each riding demand, the maximum accepted price is uniformly distributed in $[p_{min}, p_{max}]$, where p_{min} is set to $10 + 2 \times \text{dis}(l_r^s, l_r^e)$, and p_{max} is set to 1.5 times p_{min}. That is, for each demand r, its maximum accepted unit price is $val_r \sim U\left(\frac{10+2\times dis(l_r^s,l_r^e)}{dis(l_r^s,l_r^e)}, \frac{15+3\times dis(l_r^s,l_r^e)}{dis(l_r^s,l_r^e)}\right)$. The maximum waiting time is randomly selected within$[\Delta t, 2\Delta t]$. The vehicle fuel consumption v_r cost is randomly selected from $\{1.4, 1.5, 1.6, 1.7\}$CNY/km. In the experiment, the unit pricing range is $[4, 7]$ (CNY/km). The experimental parameter settings are shown in Table 1.

[1] https://outreach.didichuxing.com/research/opendata/.

Table 1. Experimental parameters

Parameter	Value
Length of each time step $\Delta t(s)$	60
Map longitude range	$[104.03°E, 104.12°E]$
Map latitude range	$[30.62°N, 30.71°N]$
Number of vehicles $\|V\|$	$600, 800, 1000, 1200, 1400$
Maximum waiting time	$[\Delta t, 2\Delta t]$
The unit cost of the vehicle c_v (CNY/km)	$1.4, 1.5, 1.6, 1.7$

5.1 Benchmark Approaches and Metrics

We evaluate the proposed multi-region dynamic pricing algorithm against the following benchmark approaches.

FIX (Fixed Pricing). Traditional ride-hailing usually uses a fixed pricing algorithm, where the unit price does not change over time [7]. This is a kind of uniform pricing algorithm for all regions. In this paper, we consider a fixed pricing algorithm, which sets a fixed unit price for all regions over the whole time. Specifically, we repeat the experiment to explore in the whole pricing domain to find the unit price that maximizes the platform's profit, which is:

$$p_{fix} = \underset{p}{\text{argmax}} G(p) \sum_{\tau=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{m_i} \mathbb{I}(f_{r_j} = 2)(p \times \text{dis}(l^s_{r_j}, l^e_{r_j}) - C_{r_j}) \tag{3}$$

SDE. This is also multi-region pricing algorithm that dynamically sets unit price based on the different supply and demand. Specifically, according to [13], when the number of vehicles is greater than the number of riding demands, we set $p_b = \underset{p}{\text{argmax}} pG(p)$, i.e. the platform can make the largest profit. The following pricing rules are designed:

$$p = \begin{cases} p_b & if\,|V_t| \geq |M_t| \\ p_b\left(1 + 2e^{|V_t|-|M_t|}\right) & if\,|V_t| < |M_t| \end{cases}$$

where at each time step, the platform determines the unit price according to the supply and demand of each region. The passengers acceptance probability of unit price per kilometer is obtained by studying the historical order data, as shown in Fig. 3, and then we calculate the value of p_b is about 4.8. From Fig. 4, when p is about equal to 4.8, the value of $pG(p)$ reaches the maximum.

GREEDY (Greedy Pricing). Existing works show that greedy algorithms can perform well in the crowdsourcing problems [12]. Therefore, in this paper we also design a greedy based multi-region pricing algorithm, which tries to maximize the platform's profit at the current time step. Similar to the **SDE** algorithm, when the number of vehicles is greater than the number of demands,

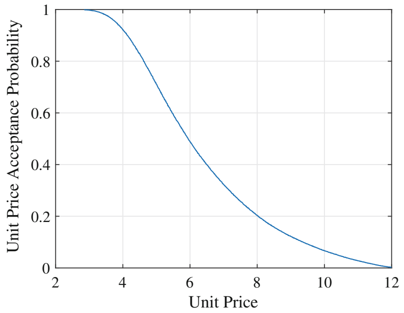


Fig. 3. Passenger acceptance probability of unit price

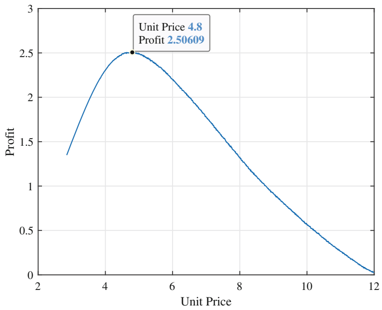


Fig. 4. Total profit when vehicles are sufficient

p_b is the optimal pricing. When the number of vehicles is less than the number of riding demands, we can combine the passenger's price acceptance probability model learned from historical data, and calculate the optimal unit price under the current state based on this model.

In order to evaluate the performance of **MRDP**, we consider the following metrics.

- **Platform profit.** The platform's profit refers to the sum of the actual profit of the platform over all time steps.
- **The number of served orders.** The number of served orders refers to the sum of the number of orders successfully served over all time steps, which can evaluate the effectiveness of the pricing.
- **Average order profit.** The average order profit is the ratio of total profit of the platform to the number of served orders over all time steps. This metrics can evaluate the ability of the platform making profit on each individual order.
- **Market supply and demand changes.** We use the number of idle vehicles in the market minus the number of riding demands to represent the difference between the supply and demand, that is, $\sum_{i=1}^{N}(|V_t| - |R_t|)$, from which we can investigate the impact of pricing strategies on the market supply and demand.

5.2 Experimental Results

The experiments are run on a machine with AMD Ryzen7 4800H processor. In the experiment, we increase the number of vehicles from 600 to 1400 with step 200. For each number of vehicles, the experiment is repeated for 10 times, and then we compute the average results for analysis.

Firstly, the platform's profit of four different algorithms are shown in Fig. 5. We find that as the number of vehicles increases, the platform's profit is also rising. This is because more orders will be served when more vehicles participate, and thus the platform's profit is increased. The supply of vehicles shows a trend from shortage of supply to saturation. We can find that **MRDP** can outperform all other three algorithms. **GREEDY** can perform better than another two algorithms. In more details, we find that multi-region based pricing algorithms (**MRDP**, **SDE** and **GREEDY**) generally can bring more profit to the platform than the **FIX** pricing algorithm, which sets a uniform and fixed price for all regions. Note that when the supply is insufficient, the profit of **SDE** is a little lower than **FIX**. While the supply is gradually saturated, the profit of **SDE** is greatly improved compared with **FIX**. After the gradual saturation of vehicle supply, the performance of **SDE** algorithm is better than **FIX** since it can maximize the profit in the saturation situation [13]. In summary, **MRDP** can achieve the maximum profit with respect to the dynamic supply and demand compared to other algorithms.

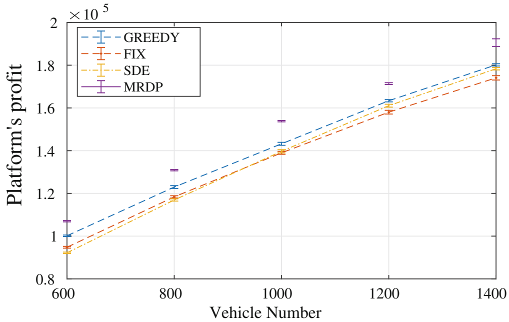


Fig. 5. Platform's profit

From Fig. 6, we find that the number of served orders using the **MRDP** algorithm and the **GREEDY** algorithm are similar, and both are significantly higher than another two algorithms. In combination with Fig. 5, we can see that the **MRDP** algorithm can make more profit than **GREEDY**. The multi-region based pricing algorithms can serve more passengers than the **FIX** pricing algorithm, and brings more profit. Then we can find that differentiated pricing in different regions has greatly improved the efficiency of platform, that is, each vehicle can serve more passengers on average.

From Fig. 7, with the increased number of vehicles, the average profit of a vehicle to complete an order decreases. This is because as the number of vehicles increases, the supply of vehicles in some regions is becoming too sufficient. Therefore the platform decreases the unit price so that more passengers are willing to accept the price. Thus the profit of each single order may decrease. However, the overall profit of the platform increases (see Fig. 5). Furthermore, from Figs. 5 and 6, we find that the **FIX** algorithm has the highest average order profit, which can be concluded that the **FIX** algorithm directly increases the platform profit by increasing the unit price, while the **SDE** algorithm will lose many orders due to high price when there are fewer vehicles. The **GREEDY**

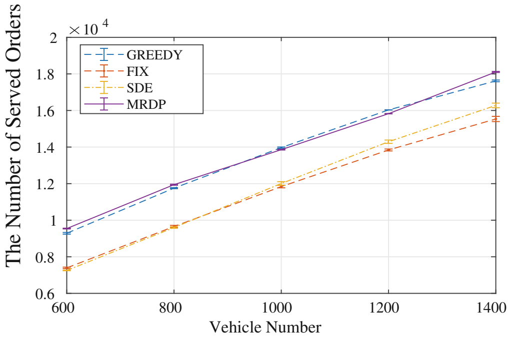


Fig. 6. The number of served orders

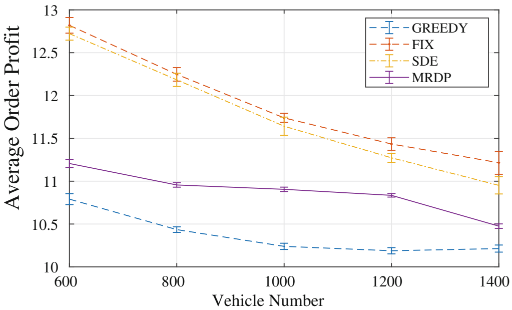
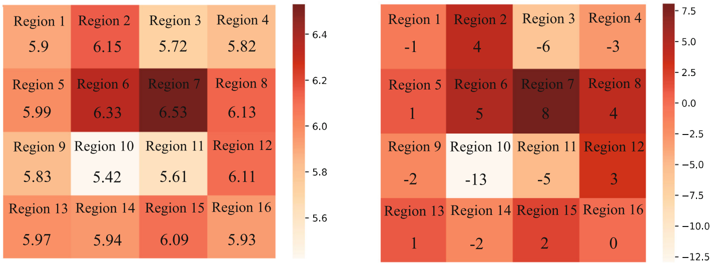


Fig. 7. Average order profit

algorithm and **MRDP** algorithm can set unit price according to the current supply and demand state, and thus can make profit while serving more orders. The **MRDP** algorithm has learned the passenger's acceptance of the unit price, and within the acceptable range of the passenger, it can make higher profit for each order than **GREEDY**, and it will not lose the order due to high pricing.

The market supply and demand changes and the thermal diagram of the prices set by **MRDP** algorithm at a certain time are shown in Fig. 8. The value in each block in Fig. 8(a) is the unit price in each region, and the value in each block in Fig. 8(b) is the difference between the number of riding demands and the number of idle vehicles. We can find that the unit price of each region is consistent with the supply and demand status of each region. For example, we can see that the difference between the number of riding demands and idle vehicles in region 10 is -13. At this time, the supply of vehicles is very sufficient, and the vehicles do not have enough riding demands to serve. Therefore, **MRDP** decreases the unit price to avoid losing orders. The difference between the number of riding demands and the number of idle vehicles in region 7 is 8. At this time, idle vehicles are not enough to meet all the riding demands. Therefore, **MRDP** increases the unit price.

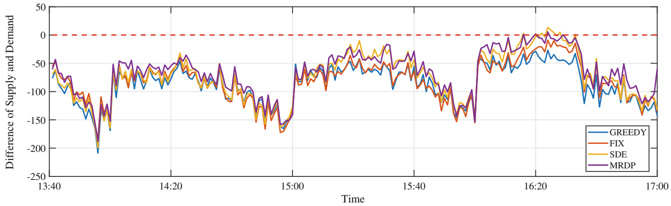


(a) Unit pricing heat map

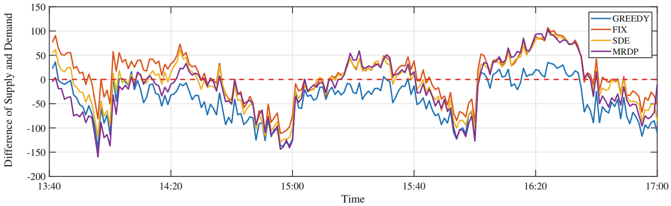
(b) Heat map of supply and demand in ride-hailing market

Fig. 8. The unit pricing heat map and the supply and demand status

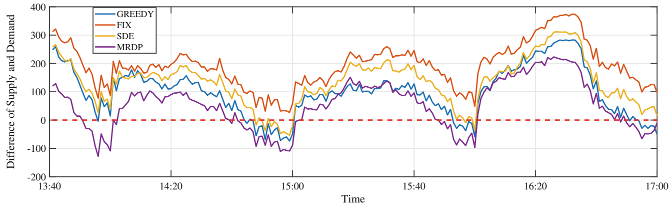
Furthermore, by observing the difference between the number of vehicles and demands at each time step, we can obtain the change curve of supply and demand in the ride-hailing market at 13:40–17:00 when the platform adopts different pricing algorithms with 600, 1000 and 1400 vehicles respectively. This is shown in Fig. 9. When the difference between supply and demand is close to 0, it means that the market supply and demand are balanced. From Fig. 9(a), we can find the market is basically in a state of short supply. However, the **MRDP** supply-demand difference curve is closer to 0, showing a better performance. In Fig. 9(b), the market is relatively stable. The **GREEDY** supply and demand difference curve seems to be more balanced than **MRDP** from 15:00 to 16:30. According to Fig. 6, it can be found that the number of served orders for **GREEDY** is slightly less than that **MRDP**, and most of the supply and demand difference curve of **MRDP** is above **GREEDY**, indicating that **MRDP** use fewer vehicles to achieve better platform profits than **GREEDY**. In Fig. 9(c), the vehicle supply in the entire market is relatively sufficient, and the supply is greater than demand at most of the time. **MRDP** can reach more balanced supply and demand, and and can serve more orders when the supply of vehicles is sufficient (see Fig. 6). In all these cases, we can find that the overall supply and demand of **MRDP** are relatively balanced during the entire time steps, which may imply that the **MRDP** algorithm can help to balance market supply and demand to a certain extent.



(a) Number = 600



(b) Number = 1000



(c) Number = 1400

Fig. 9. Market supply and demand curve under different number of vehicles

6 Conclusion

In this paper, we propose a multi-region dynamic pricing algorithm (**MRDP**) based on deep reinforcement learning by considering the real-time supply and demand status of different regions in order to maximize the ride-hailing platform's profit. In order to evaluate the effectiveness of **MRDP**, we run experimental analysis based on real Chengdu ride-hailing order data. We evaluate **MRDP** with three typical benchmark algorithms. The experimental results show that **MRDP** can outperform other algorithms. We can find that **MRDP** can set the unit price dynamically for each region according to the real-time change of market supply and demand status to maximize the profit, and it can also balance the market supply and demand to some extent.

Acknowledgment. This paper was funded by 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. 18JHQ060) and Shenzhen Fundamental Research Program (Grant No. JCYJ20190809 175613332).

References

1. Asghari, M., Deng, D., Shahabi, C., Demiryurek, U., Li, Y.: Price-aware real-time ride-sharing at scale: an auction-based approach. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10 (2016)
2. Asghari, M., Shahabi, C.: An on-line truthful and individually rational pricing mechanism for ride-sharing. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10 (2017)
3. Asghari, M., Shahabi, C.: Adapt-pricing: a dynamic and predictive technique for pricing to maximize revenue in ridesharing platforms. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 189–198 (2018)
4. Chen, H., et al.: InBEDE: integrating contextual bandit with TD learning for joint pricing and dispatch of ride-hailing platforms. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 61–70 (2019)
5. Chen, L., Shang, S., Yao, B., Li, J.: Pay your trip for traffic congestion: dynamic pricing in traffic-aware road networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 582–589 (2020)
6. Chen, L., Gao, Y., Liu, Z., Xiao, X., Jensen, C.S., Zhu, Y.: PTRider: a price-and-time-aware ridesharing system. Proc. VLDB Endow. **11**(12), 1938–1941 (2018)
7. Chen, M., Shen, W., Tang, P., Zuo, S.: Dispatching through pricing: modeling ride-sharing and designing dynamic prices. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019, pp. 165–171 (2019)
8. Gan, J., An, B., Wang, H., Sun, X., Shi, Z.: Optimal pricing for improving efficiency of taxi systems. In: 23rd International Joint Conference on Artificial Intelligence, pp. 2811–2818 (2013)
9. Liu, J.X., Ji, Y.D., Lv, W.F., Xu, K.: Budget-aware dynamic incentive mechanism in spatial crowdsourcing. J. Comput. Sci. Technol. **32**(5), 890–904 (2017)
10. Munkres, J.: Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**(1), 32–38 (1957)
11. Schröder, M., Storch, D.M., Marszal, P., Timme, M.: Anomalous supply shortages from dynamic pricing in on-demand mobility. Nat. Commun. **11**(1), 1–8 (2020)
12. Tong, Y., She, J., Ding, B., Chen, L., Wo, T., Xu, K.: Online minimum matching in real-time spatial data: experiments and analysis. Proc. VLDB Endow. **9**(12), 1053–1064 (2016)
13. Tong, Y., Wang, L., Zhou, Z., Chen, L., Du, B., Ye, J.: Dynamic pricing in spatial crowdsourcing: a matching-based approach. In: Proceedings of the 2018 International Conference on Management of Data, pp. 773–788 (2018)

14. 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)
15. Zhang, J., Wen, D., Zeng, S.: A discounted trade reduction mechanism for dynamic ridesharing pricing. IEEE Trans. Intell. Transp. Syst. **17**(6), 1586–1595 (2015)
16. Zheng, L., Chen, L., Ye, J.: Order dispatch in price-aware ridesharing. Proc. VLDB Endow. **11**(8), 853–865 (2018)
17. 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 (2019)