An Incentive-Compatible and Efficient Mechanism for Matching and Pricing in Ride-Sharing

Bing Shi[1,2](✉), Xizi Huang[1], and Zhi Cao[1]

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

Abstract. As a novel and economic transportation way, ride-sharing has attracted more and more passengers and drivers to participate. How to match passengers with drivers efficiently has become a key issue. Specifically, drivers are usually heterogeneous with different costs, and they may behave strategically (e.g. reveal their private cost information untruthfully) in order to make more profits. Drivers' strategic behavior may lead to inefficient matching, and thus result in the loss of social welfare of ride-sharing platform and drivers. In this paper, we intend to solve this issue by designing an incentive-compatible and efficient mechanism, which can match passengers with drivers and determine the payments to drivers in order to maximize the social welfare while ensuring drivers reveal their cost information truthfully. Specifically, we design an order matching algorithm with a branch and bound based route planning algorithm to accelerate the matching process. Meanwhile, we compute the payments to drivers based on the second pricing mechanism. In so doing, we propose a second pricing based ride-sharing mechanism (SPRM), which satisfies incentive compatibility, individual rationality, budget balance and computational efficiency. We further run extensive experiments to evaluate our mechanism based on the real Manhattan taxi order data and vehicle fuel consumption data. The experimental results show that SPRM can guarantee drivers' profits and improve the ratio of drivers' participation and the ratio of served orders, and eventually achieve greater social welfare than two typical benchmark approaches, GPri and ND.

Keywords: Ride-sharing · Mechanism design · Incentive compatibility · Order matching · Pricing

1 Introduction

The increased amount of vehicles results in serious traffic congestion in the urban transportation system [4]. However, these vehicles are not utilized well when providing riding service. Actually, a vehicle usually carries only 1.6 passengers on average [5]. To relieve the traffic pressure, crowdsourcing based ride-sharing

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13370, pp. 191–204, 2022.
https://doi.org/10.1007/978-3-031-10989-8_16

methods are proposed, and more and more companies (e.g. Didi Chuxing and Uber) have entered this market to increase vehicle seat usage. As more and more passengers and drivers participate in such a business, the ride-sharing platform needs to match riding orders with drivers efficiently and determine the payments to drivers to incentivize them to provide the riding service.

In this paper, we assume that the platform charges up-front fares to passengers when they submit the riding orders. Note that it could be problematic to require passengers to bid for vehicles and this kind of mechanism may not work well in the practical application [10]. Therefore, we adopt the up-front fares and do not consider the strategic bidding behavior of passengers. The platform determines the up-front fares according to the pick-up and drop-off locations, current demand and supply and so on. Passengers then determine whether to accept the up-front fares or not. Then the same as some existing works [14,17], we assume that the ride-sharing platform adopts an auction mechanism to determine the matching between the orders and drivers. Specifically, the platform publishes the orders to available drivers, and then drivers compete with each other by bidding their costs for serving orders. The cost depends on several factors, such as fuel cost, vehicle wear, maintenance cost and so on. The platform then matches orders with drivers and makes the payments to drivers. However, drivers are usually self-interested, and they may untruthfully report their costs to make more profits. Drivers untruthfully revealing their cost information may lead to inefficient order matching, and then reduce the overall social welfare of drivers and the platform. Moreover, the mechanism may need to match a large number of incoming orders with thousands of feasible vehicles. Therefore, the mechanism should be computationally efficient, Therefore, we need to design an efficient mechanism that incentivizes drivers to bid truthfully to maximize the social welfare.

In more detail, we design an efficient mechanism to match orders with drivers to maximize the social welfare of the platform and drivers. This paper advances the state of art in the following ways. We design an approximated order matching algorithm with a branch and bound based route planning algorithm to accelerate the matching process, and then compute the payments to the drivers based on the second pricing mechanism. In so doing, we design a second pricing based ride-sharing mechanism, which is named SPRM. We prove that this mechanism can satisfy incentive compatibility, individual rationality, budget balance[1] and computational efficiency. We run extensive experiments to evaluate our mechanism based on the real Manhattan taxi order data and vehicle fuel consumption data. We find that SPRM can make greater social welfare than two typical benchmark approaches, GPri [17] and Nearest Dispatching (ND). The experimental results also show that SPRM can guarantee the drivers' profits and improve the ratio of drivers' participation and the ratio of served orders.

[1] Incentive compatibility means that drivers have no incentive to strategically manipulate their costs to increase their profits. Individual rationality means that the driver's profits are not negative. Budget balance means that the profits of the platform are not negative.

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

2 Related Work

There exist plenty of works on investigating the issues of the crowdsourcing based ride-sharing systems, such as minimizing the total traveling distance [3,8], maximizing the ratio of served orders [13,15], maximizing the platform's profits [16] and so on. Furthermore, there exist some works using auction based methods to match orders with drivers. In [1], the ride-sharing platform proposes an auction-based mechanism to maximize its own profits. In [6], the authors propose a heuristic algorithm to determine ride-sharing travel plan dynamically and use VCG mechanism to determine the payments of participants. In [7], a second-price sealed auction mechanism is proposed to do the order matching to increase the ratio of served orders and minimize the traveling distance. In [17], an auction-based method is proposed to balance the supply and demand of drivers and passengers. Furthermore, there exist some works on investigating travel expenses of passengers [2], such as that in [10], a pricing algorithm is proposed for passengers based on their riding demands, and an online mechanism is proposed to encourage passengers to report information truthfully.

However, to the best of our knowledge, existing works usually ignore the profits of vehicle drivers, which may result in drivers refusing to provide the ride-sharing service if they make a loss in this business. Therefore in this paper, we propose an incentive-compatible mechanism to maximize the social welfare of the platform and the drivers, in order to ensure that both sides are willing to participate in the ride-sharing business.

3 Basic Settings

In this section, we first describe how the ride-sharing system works in this paper, and then give the relevant symbols and definitions. Finally, we provide a formal definition of the problem.

We first introduce how the online ride-hailing system works. When a passenger sends the riding orders to the platform, the platform computes the up-front fare charged to the passenger. Then the platform puts the orders where the passengers have accepted the fares into the order pool and publish these orders to the online drivers. The driver will bid on the feasible orders, and then the platform matches the orders according to the drivers' bids and compute the payments to drivers.

3.1 Symbols and Definitions

We model the order matching process as a one-side reverse auction which runs in a set of time slot $\mathcal{T} = \{1, 2, \cdots, T\}$. Drivers will run on a road network to serve orders following travel plans. The related concepts are defined as follows.

Definition 1 (Order). *Order $o \in \mathcal{O}$ is defined as a tuple $(l_o^s, l_o^e, t_o^s, \overline{wt}_o, \overline{dr}_o, n_o, f_o)$, where $\mathcal{O}$ represents the set of orders. l_o^s and l_o^e represent the departure and destination of order o respectively, t_o^s is the earliest time when order o leaves l_o^s, $\overline{wt}_o$ is the maximum time that order o is willing to wait after t_o^s, and $\overline{dr}_o$ indicates the maximum detour ratio that passengers can tolerate for order o, n_o is the number of passengers in order o, f_o is the up-front fare charged to passengers by the platform.*

We use $dis(l_o^s, l_o^e)$ to represent the shortest distance between l_o^s and l_o^e. Note that passengers will withdraw their orders if the actual departure time $\overline{t_o^s}$ is more than $t_o^s + \overline{wt}_o$. Furthermore, order o has a certain detour tolerance where passengers hope that the trip distance cannot exceed $(1 + \overline{dr}_o) \cdot dis(l_o^s, l_o^e)$.

Definition 2 (Driver). *Driver $d \in \mathcal{D}$ is defined as a tuple $(\theta_d, \mathcal{O}_d)$, where $\theta_d = (l_d^s, l_d^e, [t_d^s, t_d^e], c_d, n_d)$ is the driver's type, which is the private information known by himself. In more detail, l_d^s and l_d^e represent the driver's origin and destination respectively. $[t_d^s, t_d^e]$ is the driving time interval, where t_d^s and t_d^e represent the earliest time to leave l_d^s and the latest time to reach l_d^e. c_d is the unit cost, and n_d is the maximum number of passengers that the vehicle can carry, $\mathcal{O}_d$ represents the set of orders allocated by the platform to driver d, where $\mathcal{O}_d \subseteq \mathcal{O}$. Furthermore, we use $\Theta = \{\theta_1, \theta_2, \cdots, \theta_{|D|}\}$ to denote the set of private types of all drivers, and use Θ_{-d} to denote the set of private types of all drivers excluding driver d. We use $\overline{V}$ to represent the average driving speed.*

Definition 3 (Travel Plan). *tp_d is defined as the travel plan of driver d, which is a sorted sequence of nodes $l_d^1 \to l_d^2 \to \cdots \to l_d^{|tp_d|}$, where $l_d^k (k = 1, 2, 3, ..., |tp_d|)$ is the origin or destination of the order allocated to driver d (that is $l_d^k \in \{l_o^s \mid o \in \mathcal{O}_d\} \cup \{l_o^e \mid o \in \mathcal{O}_d\}$).*

Note that the travel plan is dynamically changed when a new order is inserted. We use $dis_{tp_d}(l_u, l_v)$ to represent the actual traveling distance between l_u and l_v on the travel plan t_p, and $dis_d(tp_d)$ is the traveling distance of driver d at location l_d following travel plan tp_d:

$$dis_d(tp_d) = dis(l_d, l_d^1) + \sum_{k=1}^{|tp_d|-1} dis(l_d^k, l_d^{k+1}) \tag{1}$$

Now $dr_{tp_d}(o)$ is the actual detour ratio of order o under travel plan tp_d:

$$dr_{tp_d}(o) = dis_{tp_d}(l_o^s, l_o^e)/dis(l_o^s, l_o^e) - 1 \tag{2}$$

According to the above definition, if the driver's travel plan is feasible, the following conditions should be satisfied:

(1) $\forall o \in \mathcal{O}_d$, we have $l_o^s \prec l_o^e$ for l_o^s and l_o^e in tp_d.
(2) $\forall o \in \mathcal{O}_d$, we have $dis_{tp_d}(l_d^s, l_o^s) \leq (\overline{t_o}^s - t_d^s) \cdot \overline{V}$.
(3) $\forall o \in \mathcal{O}_d$, we have $dr_{tp_d}(o) \leq \overline{dr}_o$.
(4) $\forall o \in \mathcal{O}_d$, we have $dis_{tp_d}(l_d^s, l_d^e) \leq (t_d^e - t_d^s) \cdot \overline{V}$.
(5) The number of passengers in the vehicle does not exceed n_d.

The above condition 1 means that the driver needs to pick up and drop off passengers according to the origin and destination of the order. Condition 2 and 3 mean that the travel plan should satisfy the order's detour ratio and waiting time. Condition 4 means that the driver must arrive at the destination on time. Condition 5 means that the number of passengers in the vehicle must meet the seat limit. The travel plan that satisfies the above five conditions is called a feasible plan. We use $\mathcal{F}_d$ to represent the set of all feasible plans of the driver.

3.2 Social Welfare

In the ride-sharing system, driver d may not strategically report the type information such as departure l_d^s, destination l_d^e, travel time $[t_d^s, t_d^e]$, and vehicle seat number n_d since the platform can use some technical manners to prevent such misreporting (e.g. acquiring vehicle's location using GPS, computing the remaining available seat capacity according to vehicle type and current served orders). However, the driver can misreport the unit cost c_d to make more profits since the ride-sharing platform cannot verify the driver's cost information.

According to Eq. 1, the cost of driver d following travel plan tp_d is $cost_d(tp_d) = c_d \cdot dis_d(tp_d)$, and its profit is:

$$u_d = p_d - cost_d(tp_d) \tag{3}$$

where p_d is the payment received from the platform. The platform makes profits through the difference between the passengers' up-front fares and the payments to drivers:

$$\mathcal{U}^P = \sum_{d\in\mathcal{W}} (\sum_{o\in\mathcal{O}_d} f_o - p_d) \tag{4}$$

where $\mathcal{W} \subseteq \mathcal{D}$ is the set of drivers serving orders. Now, the social welfare of the platform and the drivers is:

$$SW = \mathcal{U}^P + \sum_{d\in\mathcal{W}} u_d = \sum_{d\in\mathcal{W}} (\sum_{o\in\mathcal{O}_d} f_o - cost_d(tp_d)) \tag{5}$$

3.3 Problem Formulation

Based on the above description, we give the problem formulation of this paper.

Definition 4 (Order Matching for Maximizing Social Welfare). *Given the set of orders $\mathcal{O}$ and the set of drivers $\mathcal{D}$, the platform determines the order matching to maximize the social welfare SW, while ensuring that the driver $d \in \mathcal{W}$ who gets the order can successfully serve the order, i.e. the travel plan of each driver is feasible ($tp_d \in \mathcal{F}_d \neq \varnothing$).*

Definition 5 (Driver Pricing). *The platform needs to determine the payment to driver $d \in \mathcal{W}$ according to the order matching result, the set of orders $\mathcal{O}$ and the set of private types $\hat{\Theta}$ submitted by all drivers, satisfying incentive compatibility, individual rationality and budget balance.*

4 Mechanism Design

We now design a Second-Pricing based Ride-sharing Mechanism (SPRM) to do the order matching and pricing, which is shown in Algorithm 1. Line 2 in Algorithm 1 uses the order matching algorithm to allocate the orders to drivers. For each matched pair (o, d^*) in $\mathcal{M}$, SPRM uses the second price pricing algorithm to compute the payments to driver d^* in line 3 to 5. In the following, we will introduce the order matching algorithm and driver pricing algorithm in detail.

4.1 Order Matching Algorithm

The order matching algorithm of SPRM mechanism is shown in Algorithm 2. Specifically, line 2 sorts the orders according to the descending up-front charged fares, and in lines 3 to 6, the sorted orders are matched one by one. In the matching process, the driver with the lowest additional travel cost is selected to ensure the maximum social welfare.

Algorithm 1. Second-Pricing based Ride-sharing Mechanism

```
Input:
  the set of orders $\mathcal{O}$, the set of drivers $\mathcal{D}$
Output:
  the set of drivers $\mathcal{W}$ who obtain the orders, and the corresponding set of payments
  $\mathcal{P}$
1: Initialize:$\mathcal{W} \leftarrow \varnothing, \mathcal{P} \leftarrow \varnothing$;
2: $\mathcal{M}$ =Order_Matching($\mathcal{O}, \mathcal{D}$);
3: foreach matched pair $(o, d^*) \in \mathcal{M}$ do
4:    Second_pricing($(o, d^*), \mathcal{D}, \mathcal{W}, \mathcal{P}$);
5: end
6: return: $\mathcal{W}, \mathcal{P}$;
```

Algorithm 2. Order Matching Algorithm: Order_Matching($\mathcal{O}, \mathcal{D}$)

```
Input:
  the set of orders $\mathcal{O}$, the set of drivers $\mathcal{D}$
Output:
  the set of matched pairs $\mathcal{M}$
1: Initialize: $\mathcal{M} \leftarrow \varnothing$;
2: Sort the orders in $\mathcal{O}$ in descending up-front fares to get $\mathcal{O}'$;
3: foreach $o \in \mathcal{O}'$ do
4:    $d^*, \Delta cost_{o,d^*} \leftarrow$ Select_Best_Driver $(o, \mathcal{D})$;
5:    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(o, d^*)\}$;
6: return: $\mathcal{M}$;
```

Driver Selection Algorithm. In the order matching, we select the driver with the lowest additional travel cost from the feasible set of drivers to server the order. For serving order o, the additional travel cost for driver d is:

$$\Delta \cos t_{o,d} = \hat{c}_d \cdot (dis_d(tp_d \cup \{l_o^s, l_o^e\}) - dis_d(tp_d)) \tag{6}$$

where $tp_d \cup \{l_o^S, l_o^e\}$ is the new travel plan after adding the departure and destination of order o into the original travel plan of driver d.

The driver selection algorithm is shown in Algorithm 3. For a certain order $o \in \mathcal{O}$, line 3 will determine whether driver $d \in \mathcal{D}$ can pick up the passengers on time and whether the driver can arrive at the destination on time. Then line 5 will reschedule the route according to the departure and destination of order o and the driver's current travel plan tp_d. The route planning algorithm adopts a branch and bound based method, which will be introduced in the below. Lines 6 and 7 can find the best driver d^* matched with order o.

In the above route planning problem, we design a branch and bound based route planning algorithm to reschedule the travel plan. The algorithm will recurse all possible travel plans, and we will use pruning to eliminate nonviable branches. There are pruning operations at each recursion step to ensure that the current travel plan is feasible to reduce the searching cost and accelerate the route planning. The travel cost pruning is used to eliminate the recursive branch whose current cost has exceeded the optimal cost. The travel plan feasibility pruning use the five conditions of a feasible plan to prun the current travel plan by selecting the branches that can form a feasible plan. The time complexity is equivalent to the time complexity of recursive searching, with $O(\ell^2)$ in the worst case, where $\ell = \max_{d \in \mathcal{D}_t}\{|tp_d|\}$.

Algorithm 3. Driver Selection Algorithm: Select_Best_Driver$(o, \mathcal{D})$

```
Input:
  order $o$, the set of orders $\mathcal{D}$
Output:
  driver $d^*$ with the lowest additional travel cost $\Delta cost^*$
 1: $d^* \leftarrow NULL, \Delta cost^* \leftarrow \infty$;
 2: foreach $d \in \mathcal{D}$ do
 3:    if $dis(l_d^s, l_o^s) \leq (\bar{t}_o^s - t_d^s) \cdot \overline{V}$ and
       $dis(l_d^s, l_o^s) + dis(l_o^s, l_o^e) + dis(l_o^e, l_d^e) \leq (t_d^e - t_d^s) \cdot \overline{V}$ then
 4:       $tp^* \leftarrow \varnothing, cost^* \leftarrow \infty, S \leftarrow (\varnothing, tp_d, \{l_o^s, l_o^e\}, l_d, t, n_d, 0)$;
 5:       Use the Branch and Bound based Route Planning Algorithm to reschedule
          the travel plan;
 6:       if $tp_d^* \neq \varnothing$ and $\Delta cost^* > cost_d(tp_d^*) - cost_d(tp_d)$ then
 7:          $d^* \leftarrow d, \Delta cost^* \leftarrow cost_d(tp_d^*) - cost_d(tp_d)$;
 8:       end
 9:    end
10: end
11: return: $d^*, \Delta cost^*$;
```

4.2 Driver Pricing Algorithm

After matching orders with drivers, the platform needs to compute the payments to drivers, which is based on second pricing, as shown in Algorithm 4.

For a matching pair (o, d^*), the platform will temporarily delete the driver d^* with the lowest additional travel cost, and find driver d' with the lowest additional travel cost from the remaining drivers in line 1 of Algorithm 4, and then the platform will use $\Delta cost_{o,d^*} \leq f_o$ to determine whether the driver exists (line 2). If it exists, $\Delta cost_{o,d^*}$ is taken as the payment to driver d^*. Furthermore, the platform will allocate order o to driver d^*. Finally we update the driver's travel plan tp_{d^*} and the current number of passengers n_{d^*}.

4.3 Theoretical Analysis

SPRM mechanism can satisfy the typical economic properties: incentive compatibility, individual rationality and budget balance, and can guarantee computational efficiency.

Theorem 1. *SPRM can guarantee incentive compatibility.*

Proof. We use Myerson theorem [9] to prove that SPRM can guarantee incentive compatibility. According to Myerson's theorem, a mechanism is incentive compatible if and only if the mechanism can satisfy the following two conditions:

1) The order matching algorithm is monotonic: If the private cost information $\hat{c}_d$ revealed by driver d can help him to get the order, driver d can get the order with the lower cost information $\hat{c}'_d$ (i.e. $\hat{c}'_d \leq \hat{c}_d$).

Algorithm 4. Pricing Algorithm: Second_pricing($(o, d^*), \mathcal{D}, \mathcal{W}, \mathcal{P}$)

```
Input:
   a matched pair $(o, d^*) \in \mathcal{O} \times \mathcal{D}$, the set of drivers $\mathcal{D}$, the set of drivers $\mathcal{W}$ who obtain
   orders, and the set of payments $\mathcal{P}$
Output:
   the updated set of drivers $\mathcal{W}$ who obtain orders, and the updated set of payments $\mathcal{P}$
 1: $d', \Delta cost_{o,d'} \leftarrow Select\_Best\_Driver(o, \mathcal{D} \backslash \{d^*\})$;
 2: if $\Delta cost_{o,d^*} \leq f_o$ and $\Delta cost_{o,d'} \leq f_o$ then
 3:    if $d^* \in \mathcal{W}$ then
 4:       $p_{d^*} \leftarrow p_{d^*} + \Delta cost_{o,d'}$;
 5:    else
 6:       $p_{d^*} \leftarrow \Delta cost_{o,d'}$;
 7:       $\mathcal{W} \leftarrow \mathcal{W} \cup \{d^*\}, \mathcal{P} \leftarrow \mathcal{P} \cup \{p_{d^*}\}$;
 8:    end
 9:    $\mathcal{O}_{d^*} \leftarrow \mathcal{O}_{d^*} \cup \{o\}, n_{d^*} \leftarrow n_{d^*} - n_o$, update $tp_{d^*}$;
10: end
11: return: $\mathcal{W}$, $\mathcal{P}$;
```

2) The payments to driver is critical price: The payment to the driver is the maximum payment. In other words, if driver d reveals the private cost information $\hat{c}_d$ untruthfully, and his additional travel cost is more than p_d, he will not obtain the order.

We now prove that our mechanism can satisfy the above two conditions. Obviously, the order matching algorithm is monotonic since the driver selection algorithm chooses the driver with the lowest additional travel cost. When driver d reveals lower cost information, driver d will be the optimal driver again. Then the payment to the driver calculated by the second-price pricing is the critical price. When the cost $\hat{c}_d$ revealed by the driver is too high to make $\Delta cost_{o,d'} < \Delta cost_{o,d^*}$, the driver selection algorithm will choose driver d' instead of driver d, which causes driver d to be unmatched with order o. According to Myerson's theorem, SPRM can guarantee incentive compatibility. Theorem 1 is proved.

Theorem 2. *SPRM can guarantee individual rationality.*

Proof. We need to prove that the profit of driver d is non-negative. Due to the monotonicity of the driver selection algorithm, there is $\Delta cost_{o,d'} \geq \Delta cost_{o,d^*}$ in each round of order matching, where the driver's profit is $u_{d^*} = p_{d^*} - cost_{d^*}(tp_{d^*}) = \sum_{o \in \mathcal{O}_d}(\Delta cost_{o,d'} - \Delta cost_{o,d^*}) \geq 0$. Theorem 2 is proved.

Theorem 3. *SPRM can ensure the budget balance of the platform.*

Proof. We now prove that the profit of the platform is non-negative. In line 6 of Algorithm 4, we have $\Delta cost_{o,d'} \leq f_o$, and the platform's profit is $\mathcal{U}^p = \sum_{d \in \mathcal{W}}(\sum_{o \in \mathcal{O}_d} f_o - p_d) = \sum_{d \in \mathcal{W}} \sum_{o \in \mathcal{O}_d}(f_o - \Delta cost_{o,d'}) \geq 0$. Theorem 3 is proved.

Theorem 4. *SPRM satisfies computational efficiency.*

Proof. We now prove SPRM can be executed within a polynomial time. During each round of order matching, the worst time complexity of the driver selection algorithm for a single order is $O(|\mathcal{D}| \cdot \ell^2)$. Then the total time complexity is $O(|\mathcal{O}| \cdot |\mathcal{D}| \cdot \ell^3)$. Theorem 4 is proved.

5 Experimental Analysis

In this section, we evaluate our mechanism based on the Manhattan taxi order data set, which has been widely used by related works [1,11–13].

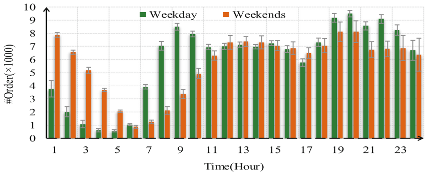

Fig. 1. Orders on weekdays and weekends

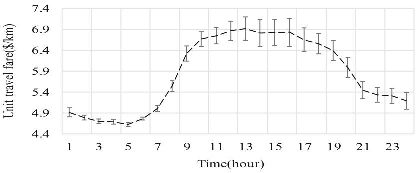

Fig. 2. Unit travel fare

5.1 Data Set and Characteristics

In this section, we introduce the data set and describe how to extract the characteristics of the data, which will be used to generate data for running experiments.

Order Data Set. We collect a data set of taxi orders on Manhattan Island in June 2016 from New York City Taxi and Limousine Commission.[2] Each order data includes the latitude and longitude of the departure and the destination, the departure time, the travel time, the fare and the travel distance.

Fuel Consumption Data Set. We collect the urban vehicle fuel consumption data of type M1 and M2 from China Automobile Fuel Consumption Query System[3] of the Ministry of Public Information of China.

Order Data Cleaning. We find that the number of orders on weekdays and weekends varies greatly over time, as shown in Fig. 1. For consistency, we use the order data of weekdays to extract the traveling characteristics.

Extracting Passenger Pricing Rules. We compute the average unit travel fare per hour, where $\overline{f}^k$ is the average unit travel fare in k-th hour. The dynamics of unit travel fare is shown in Fig. 2.

Clustering Road Network Nodes. We use K-Means algorithm to cluster 4573 nodes on Manhattan network into 40 regions according to the frequency of departure and destination in the order data set. We use $R^s = \{R^s_1, R^s_2, \cdots, R^s_{40}\}$ to represent 40 regions of the origin of the order data, and $R^e = \{R^e_1, R^e_2, \cdots, R^e_{40}\}$ to represent 40 regions of the destination of the order data.

5.2 Experiment Settings

Data Generation and Experimental Parameters. We select the peak period from 19:00 to 20:00 on weekdays as the time period We use time slot $\Delta t = 5\,\text{min}$ to divide one hour into 12 time slots, and each order contains only one passenger. We set the average vehicle speed to be 7.2 mph. We randomly selects a node from R^s_u and R^e_v respectively as the departure node l^s_o and destination node l^s_e of order o, and then uses $f_o = \overline{f}^{20} \cdot dis(l^s_o, l^e_o)$ to calculate the trip

[2] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[3] http://chaxun.miit.gov.cn/asopCmsSearch.

fare of order o. In addition, the maximum waiting time $\overline{wt_o}$ is randomly selected from {5 min, 10 min, 15 min, 20 min}, and the detour ratio $\overline{dr_o}$ is randomly selected from {0.25, 0.50, 0.75, 1.00}. Since the unit cost and fuel consumption data are not in the same magnitude, we use a factor 2.5/6.8/1.6[4] to scale the fuel consumption data to the taxi rate in New York City. Besides, we randomly select the time slot as the driver's departure time, randomly select a value from [10 min, 60 min] as the length of the travel time interval, and randomly select the node as the driver's departure point. The capacity of vehicle seats is set to 4.

Benchmark Approaches and Metrics. In this paper, we choose two typical benchmark approaches, which are GPri [17] and Nearest Dispating (ND). GPri uses a greedy algorithm to match orders and uses a uniform pricing rule for drivers. The nearest dispatching (ND) algorithm is adopted by major ride-sharing platforms such as Uber, which selects the nearest available driver to serve the new order. We run experiments to evaluate our mechanism against these two approaches in terms of the following metrics.

- **Ratio of participated drivers** which refers to the ratio of the number of drivers who serve orders to the total number of drivers.
- **Average profits of participated drivers** which refer to the ratio of the cumulative profits of drivers serving orders to the total number of such drivers.
- **Ratio of served orders** which refers to the ratio of the number of served orders to the total number of orders.
- **Average order payments** which refer to the ratio of the total payments of the matched orders to the number of orders.
- **Average random travel distance** which refers to the ratio of the total driving distance of drivers who are currently not serving any orders to the total number of drivers.
- **Social welfare** which refers to total profits of the platform and drivers.

5.3 Experimental Results

The experiments are run on a machine with AMD Ryzen7 4800H processor. The experiment increases the number of drivers from 500 to 2500. For each number of drivers, the experiment is repeated 10 times and we compute the average results.

We first analyze the ratio of participated drivers. According to Fig. 3, we find that the pricing rules of GPri and ND will cause a large number of drivers to refuse to provide services. Compared with SPRM, the participation ratio is reduced by about 20% and 11% respectively. When the total number of vehicles increases, the number of drivers serving orders has increased from 450 to 1000, and then become saturated. Because more vehicles are involved in the business, but with limited incoming orders, the driver maybe not willing to participate since it cannot get enough orders. From Figs. 4 and 5, we find that the average

[4] The reason of 2.5/6.8/1.6 being used is because (1) The basic unit rate in New York City is 2.5 dollars per mile; (2) The average fuel consumption of New York City taxis is 6.8 liters/100 km; (3) 1.6 is the factor converting from miles to kilometers.

profits and ratio of served orders who provide ride-sharing services under GPri and ND are also lower than SPRM, where the average profits are only about 27% and 45% of SPRM and the ratio of served orders is only about 44% and 57% of SPRM.

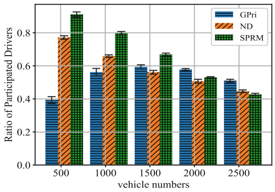

Fig. 3. Ratio of participated drivers

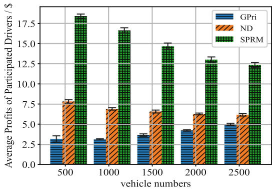

Fig. 4. Average profits of participated drivers

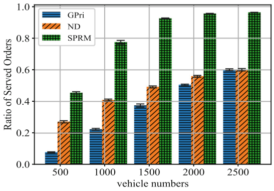

Fig. 5. Ratio of served orders

From Fig. 6, we can see that compared with SPRM, the average order payments of GPri and ND has been reduced by about 43% and 29% respectively. SPRM can make drivers obtain more orders, which can improve the average profits of drivers. From Fig. 7, we can find that as the number of vehicles increases, the average random travel distance of SPRM and ND vehicles gradually increases, and GPri does not increase or decrease monotonously. We believe this is caused by the drivers' participation. Comparing Fig. 3 with Fig. 7, we can see that as the number of vehicles increases, the drivers' participation ratio and the average random travel distance is opposite. The decreased participation ratio of drivers leads more drivers to travel randomly on the road, resulting in increased average random travel distance.

From Fig. 8, we can find that as the number of vehicles increases, the social welfare of three mechanisms increases. SPRM can make about 70% and 65% higher welfare than GPri and ND mechanisms. Furthermore, the social welfare of SPRM increases significantly until the number of drivers is saturated. This is because when the number of vehicles increases until 1500, more orders are matched, and thus the social welfare grows faster.

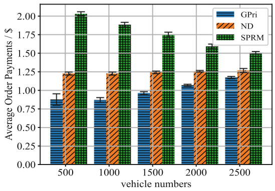

Fig. 6. Average order payments

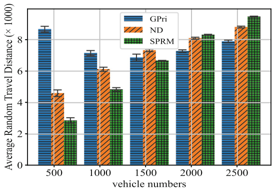

Fig. 7. Average random travel distance

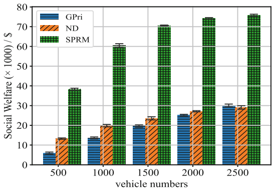

Fig. 8. Social welfare

6 Conclusion

In this paper, we designed an order matching and pricing mechanism which is incentive compatible, to maximize the social welfare of the platform and drivers. Specifically, we designed a second pricing based ride-sharing mechanism (SPRM), where we implemented an efficient order matching algorithm including branch and bound based route planning algorithm to accelerate the matching process and compute the payments to drivers based on second pricing rule. We proved that SPRM can satisfy incentive compatibility, individual rationality, budget balance and computational efficiency. Furthermore, we run extensive experiments to evaluate our mechanism based on the real Manhattan taxi order data. We found that SPRM can guarantee the driver's profits, achieve a better ratio of participated drivers and the ratio of served orders, and eventually achieve greater social welfare than two typical benchmark approaches, GPri and ND.

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. 18JHQ060) and the Fundamental Research Funds for the Central Universities (WUT:2022IVB004).

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: The 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 1–10 (2016)
2. Babaioff, M., Dughmi, S., Kleinberg, R., Slivkins, A.: Dynamic pricing with limited supply. ACM Trans. Econ. Comput. **3**(1), 1–26 (2015)
3. Bei, X., Zhang, S.: Algorithms for trip-vehicle assignment in ride-sharing. In: The 32nd AAAI Conference on Artificial Intelligence, pp. 3–9 (2018)
4. Cao, B., Alarabi, L., Mokbel, M.F., Basalamah, A.: SHAREK: a scalable dynamic ride sharing system. In: The 16th IEEE International Conference on Mobile Data Management. vol. 1, pp. 4–13 (2015)
5. Jeanes, P.: Can total car emissions be reduced enough by 2020? Traffic Eng. Control **51**(1), 6–8 (2010)
6. Kamar, E., Horvitz, E.: Collaboration and shared plans in the open world: studies of ridesharing. In: The 21st International Joint Conference on Artificial Intelligence, p. 187 (2009)
7. Kleiner, A., Nebel, B., Ziparo, V.: A mechanism for dynamic ride sharing based on parallel auctions. In: The 22nd International Joint Conference on Artificial Intelligence, pp. 266–272 (2011)
8. Ma, S., Zheng, Y., Wolfson, O.: T-share: a large-scale dynamic taxi ridesharing service. In: The 29th International Conference on Data Engineering, pp. 410–421 (2013)
9. Myerson, R.B.: Optimal auction design. Math. Oper. Res. **6**(1), 58–73 (1981)

10. Shen, W., Lopes, C.V., Crandall, J.W.: An online mechanism for ridesharing in autonomous mobility-on-demand systems. In: The 25th International Joint Conference on Artificial Intelligence, pp. 475–481 (2016)
11. Tong, Y., Zeng, Y., Zhou, Z., Chen, L., Ye, J., Xu, K.: A unified approach to route planning for shared mobility. Proc. VLDB Endow. **11**(11), 1633 (2018)
12. Xu, Y., Tong, Y., Shi, Y., Tao, Q., Xu, K., Li, W.: An efficient insertion operator in dynamic ridesharing services. In: The 35th International Conference on Data Engineering, pp. 1022–1033 (2019)
13. Yuen, C.F., Singh, A.P., Goyal, S., Ranu, S., Bagchi, A.: Beyond shortest paths: Route recommendations for ride-sharing. In: The World Wide Web Conference, pp. 2258–2269 (2019)
14. Zhang, C., Wu, F., Gao, X., Chen, G.: Online auctions with dynamic costs for ridesharing. In: The 23rd International Conference on Parallel and Distributed Systems, pp. 127–134 (2017)
15. Zhang, L., et al.: A taxi order dispatch model based on combinatorial optimization. In: The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2151–2159 (2017)
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: The 35th International Conference on Data Engineering, pp. 1034–1045 (2019)