Coalitional Double Auction for Ride-Sharing with Buyout Ask Price

Xingyi Yuan, Long Chen*, Jiale Huang, Jigang Wu
School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China, 510006
*Email: lonchen@mail.ustc.edu.cn

Abstract—Most of existing ride-sharing studies either consider to minimize the total travel distance of drivers or to maximize the profit of the service platform. Little attention has been paid to utility optimization with passengers’ budgets. Moreover, the fact that drivers have their preferable routes and the influence of buyout ask prices have been overlooked. We formulate the ride-sharing with driver buyout ask price and user-budget constraints as an integer linear programming. The objective is to maximize the total utility of passengers without hurting the benefits of drivers. We prove that the problem is NP-hard and design a coalitional double auction algorithm to solve the problem. We prove that the coalition formation approach can converge to a stable coalition partition within limited iterations and the secondary pricing based strategy satisfies essential economic properties including truthfulness, individual rationality and budget balance. Extensive experiments with real-world taxi data-set demonstrate that our algorithm outperforms the designed baseline algorithm in terms of four metrics: matching ratio of passengers, matching ratio of drivers, passenger saving rate and running time.

Index Terms—Ride-Sharing, Coalition Formation, Auction, Buyout Price

I. INTRODUCTION

With the advances of wireless communication technologies and the proliferation of smart devices, ride-sharing [1] has emerged as a promising transportation paradigm, which can reduce pollution and traffic. With ride-sharing, passengers can get fare reductions by tolerating slight extra delays. At the same time, drivers can receive compensation by taking multiple passengers in a shared trip. Service providers like Uber and DiDi can make full use of empty seats in private vehicles as public transportation resources to make profits.

In daily life, there exist drivers who prefer to travel along their daily routine paths. For those drivers with their travel plans, they are willing to offer ride-sharing services to compensate for fuel consumptions. There are some works for ride-sharing. For example, Hasan et al. [2] proposed the commuting trip sharing problem and combined hierarchical clustering and optimization to minimize daily car usage. Ta et al. [3] assumed that each driver had his/her own itinerary and shared the vehicle with one rider if the shared travel distance was large enough. They formulated the problem as max-weighted bigraph matching problem, and proposed an efficient method to accelerate calculation. However, the above studies [2], [3] assume both drivers and passengers are willing to share and cooperate, without addressing their individual preferences or characteristics, e.g., user budgets or revenue payoffs. For revenue optimization, extensive works of ride-sharing aim either to minimize the total travel distance of drivers [4] or to maximize profit of the platform [5], without guaranteeing the utility of passengers. Moreover, some works [6], [7] assume that payoffs of drivers are determined by the platform. In that case, drivers’ individual preferences, e.g., preferable routes and expected compensations, have been overlooked. For the benefits of both passengers and drivers, our goal is to develop a joint matching and pricing method that maximizes the total utility of passengers while ensuring the interests of both drivers and passengers.

Observing the fact that drivers are relatively familiar with their commuting routes or have daily routine preferences, we suppose that each driver tends to offer ride-sharing services with a buyout ask price, which represents the expected payoff. Buyout ask price can be determined by drivers based on evaluations for their historical trajectories. Since drivers also have their own travel demands, it is reasonable for them to bear part of travel costs. The buyout ask price is advantageous to both drivers and passengers. Because drivers can get their expected payoffs, while passengers do not have to pay for detour costs during shared journey. Although drivers have to pay for detour costs in ride-sharing, we can guarantee the interests of drivers by allowing them to set their maximum tolerable detour ratios. Note that if the maximum tolerable detour ratio of a driver is set to be greater than 1, his/her benefit can be negative.

Fig. 1 is a simple example to demonstrate the benefit of a driver with a buyout ask price. For ease of understanding and explanation, we assume the buyout ask price and budgets of passengers are related to travel distance. We assume that fuel cost per kilometer is $2/\text{km}$ and driver’s ask price for passengers per kilometer is $2.5/\text{km}$. There is a driver $m_1$ who drives from home to office and expects to earn compensation for fuel consumption. The red solid line represents the personalized route of driver $m_1$ with a length of 15 kilometers. The buyout ask price for all available seats of the driver thus is $15 \times 2 = 30$. There is his/her maximum tolerable detour ratio is 0.5, which means the maximum tolerable detour length is $15 \times 0.5 = 7.5$ km. Meanwhile, there are two passengers $n_1$ and $n_2$ who intend to save fares in ride-sharing. The green and blue solid lines indicate the unshared routes of those two passengers with total budgets of 25 and 20, respectively. We assume that driver $m_1$ matches with passengers $n_1$ and $n_2$. The red dashed line indicates the shared route of them with a length of 21 kilometers. We calculate that
The actual detour ratio of driver $m_1$ is 0.4, which satisfies the detour constraint and the benefit or utility of driver is 18 ¥. According to the algorithm proposed in this paper, the benefits of passengers are 10 ¥ and 5 ¥, accordingly.

In ride-sharing, there are two modes to handle requests: instant response mode [8] and round response mode [3]. The main difference is that the former mode processes requests one by one while the latter one handles a batch of requests at a time. Considering that one passenger may not afford the buyout ask price on himself/herself, we adopt the round response mode to process requests, where the platform collects a batch of requests in a short period of time and then allocates them at the end of the time interval. In this way, passengers with limited budgets have an opportunity to group together and share costs with others. As shown in Fig. 1, we can see that neither passenger $n_1$ nor passenger $n_2$ can afford the buyout ask price alone. If the round response mode is adopted, the two passengers with similar itineraries can be packed together to match with driver $m_1$ and share the costs. However, it is a challenge to partition passengers into several groups and find the best driver for each group due to the combinatorial complexity of passengers with different quality of experience (QoE) requirements. Moreover, passengers can be selfish and report their bids untruthfully, causing utility loss of drivers. Inspired by the coalition formation game which can iteratively change the partition to maximize the system performance [9], we use this game to derive the best partition of passengers. Besides, to guarantee the utility of drivers and the fairness of passengers, we incorporate the coalition formation approach with auction theory and model the matching process as a coalitional double auction game.

The main contributions are summarized as follows:

- We formulate the ride-sharing with driver buyout ask price and user budgets as a linear programming to maximize the total utility of passengers, while guaranteeing the interests of drivers. We also prove the NP-hardness of the problem.
- A coalitional double auction based matching and pricing algorithm is designed to solve the proposed problem. We prove that the coalition formation process can converge to a stable coalition partition structure and the proposed strategy satisfies the economic properties including truthfulness, individual rationality and budget balance.
- Extensive experiments with real-world taxi data-set demonstrate that our algorithm outperforms the designed baseline algorithm in terms of four metrics: matching ratio of passengers, matching ratio of drivers, passenger saving rate and running time.

The remainder of this paper is organized as follows. Section II presents system model. Section III describes the detailed solutions. The performance evaluation is presented in Section IV. Section V concludes this paper with future remarks.

II. SYSTEM MODEL

In this section, we first overview the system architecture which includes three main phases, they are information collection phase, coalition formation phase, allocation and pricing phase. Then we model the travel demands and valuation for both drivers and passengers. After that, we propose the matching model, which includes some basic ride-sharing QoE constraints. Finally, we propose the utility model for both drivers and passengers with problem formulation.

A. Overview

We consider a general scenario where $M$ drivers offer ride-sharing services with a buyout ask price and $N$ passengers want to save money by participating in ride-sharing. Let $M = \{m_1, m_2, ..., m_M\}$ and $N = \{n_1, n_2, ..., n_N\}$ denote the set of drivers and passengers, respectively. Then we have $|M| = M$ and $|N| = N$. We model the matching process with coalitional double auction, where drivers are sellers, passengers are buyers and the platform acts as the auctioneer, as shown in Fig. 2. The details of coalitional double auction are listed as follows:
1) **Information Collection**: Auctioneer collects some essential information (travel demands, bids and ask prices) from drivers and passengers.

2) **Coalition Formation**: We term the passengers who match to the same driver as a coalition. The auctioneer divides passengers into suitable coalitions based on the travel demands to maximize the utility of passengers until a stable coalition partition is obtained.

3) **Allocation and Pricing**: Based on the coalition partition, the auctioneer determines winning drivers and passengers, and calculates their payoffs and payments.

**B. Travel Demands & Valuation**

Each driver $m_i \in \mathcal{M}$ has a trip $T_i = (o_i^s, e_i^s, a_i^s, q_i^s)$ consisting of an origin $o_i^s$, a destination $e_i^s$, a maximum tolerable detour ratio $a_i^s$ and the number of available seats $q_i^s$. Similarly, each passenger $n_j \in \mathcal{N}$ has a trip $T_j = (o_j^s, e_j^s, \Delta_j^s, a_j^b, q_j^b)$ where $o_j^s, e_j^s, \Delta_j^s, a_j^b$ and $q_j^b$ represent the origin, destination, maximum tolerable waiting time, maximum tolerable detour ratio, and the number of required seats, respectively.

Each driver has an expected fuel cost of a trip based on his/her driving experiences and wants to obtain monetary refund for fuel consumption by offering ride-sharing services. Let $v_i^s$ denote the valuation of driver $m_i$ for the travel plan, which is related to the travel length $l_i^s$ and travel time $t_i^s$ of historical trajectory from origin $o_i^s$ to destination $e_i^s$. For each driver $m_i \in \mathcal{M}$, the valuation $v_i^s$ is calculated by

$$v_i^s = \rho_i \cdot l_i^s + \sigma_i \cdot t_i^s, \forall m_i \in \mathcal{M},$$

where $\rho_i$ and $\sigma_i$ are the fuel cost per kilometer and the labor cost per minute of driver $m_i$. Based on the valuation, the ask price of driver $m_i$ is defined as a buyout ask price $A_i^s$, which is exactly the price that driver $m_i$ wants to sell the service for.

Similarly, each passenger has a valuation $v_j^b$ on the trip related to unshared travel length $l_j^b$ and the unshared travel time $t_j^b$. We define the valuation of passenger $n_j$ as

$$v_j^b = \phi_j \cdot l_j^b + \varphi_j \cdot t_j^b, \forall n_j \in \mathcal{N},$$

where $\phi_j$ and $\varphi_j$ are the price per kilometer and per minute of passenger $n_j$. Based on the valuation, we denote the bid of passenger $n_j$ as $b_j^b$, which represents the maximum price that passenger $n_j$ is willing to pay.

**C. Matching Model**

There are certain constraints to be satisfied between drivers and passengers listed as follows.

1) **Waiting time constraint.** It takes a period of time for a driver to pick up passengers from his/her current position. For each passenger $n_j \in \mathcal{N}$, the actual waiting time $\mu_j$ for passenger $n_j$ to be picked up should not exceed the maximum tolerable waiting time $\Delta_j^b$:

$$\mu_j \leq \Delta_j^b, \forall n_j \in \mathcal{N}. \quad (3)$$

2) **Radius constraint.** There is a searching radius around the origin of each driver to determine his/her candidate passengers. Let $R$ denote the searching radius and $s(o_i^s, o_j^s) \leq R$, the passenger $n_j$ can be served by driver $m_i$ if the distance between them is smaller than the radius $R$. Thus, we have:

$$s(o_i^s, o_j^s) \leq R, \forall m_i \in \mathcal{M}, n_j \in \mathcal{N}. \quad (4)$$

3) **Detour constraint.** Let $\delta_i$ denote the actual detour distance of driver $m_i$. Since drivers need to bear part of detour costs, they are reluctant to detour too far. The detour distance of each driver should be no more than the maximum tolerable detour distance:

$$\delta_i \leq a_i^b \cdot l_i^s, \forall m_i \in \mathcal{M}. \quad (5)$$

Moreover, we denote the actual travel distance of passenger $n_j$ from pick-up point to drop-off point as $L_j$, then $L_j$ should satisfy with the following constraint:

$$L_j \leq (1 + a_j^b) \cdot l_j^b, \forall n_j \in \mathcal{N}, \quad (6)$$

which means the actual travel distance of passenger $n_j$ should not be higher than the maximum allowable detour plus the travel distance when traveling alone.

4) **Capacity constraint.** The number of available seats in one vehicle must be no less than the number of assigned passengers. The following constraint should be satisfied:

$$\sum_{n_j \in \mathcal{N}} x_{ij} \cdot q_j^b \leq q_i^s, \forall m_i \in \mathcal{M}. \quad (7)$$

After the matching process, the auctioneer will announce the allocation results. Let $x_{ij}$ denote the allocation result between driver $m_i$ and passenger $n_j$, and it is defined as follows:

$$x_{ij} = \begin{cases} 1, & \text{passenger } n_j \text{ is allocated to driver } m_i \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Moreover, each passenger can be served by at most one driver. That is, the following constraint must be satisfied:

$$\sum_{m_i \in \mathcal{M}} x_{ij} \leq 1, \forall n_j \in \mathcal{N}. \quad (9)$$
D. Utility Model

Based on the allocation results, the auctioneer computes the payoff $P_i^s$ for each winning driver and the payment $p_j^b$ for each winning passenger, usually, we let $P_i^s = A_i^s$, which means the total payoff of the driver is equal to the ask price. Let $M_w$ denote the set of winning drivers and $N_w$ represent the set of winning passengers. For each driver $m_i \in M$, the utility of driver $m_i$ is the difference between the payoff and detour cost incurred in ride-sharing, i.e.,

$$u_i^s = \begin{cases} P_i^s - c_i, & m_i \in M_w \\ 0, & \text{otherwise} \end{cases},$$

(10)

where $c_i$ represents the detour cost of driver $m_i$ in the shared trip. For each passenger $n_j \in N$, the utility of passenger $n_j$ is the difference between valuation of the trip and the payment for ride-sharing, i.e.,

$$u_j^b = \begin{cases} v_j^b - p_j^b, & n_j \in N_w \\ 0, & \text{otherwise} \end{cases}.$$  

(11)

E. Problem Formulation

Given a set of drivers and passengers with different travel demands, this paper aims to maximize the utility of passengers while guaranteeing the benefit of drivers, with the QoE requirements of both drivers and passengers. The objective function is given by

$$U = \sum_{n_j \in N} u_j^b,$$

(12)

Consequently, the optimization problem for ride-sharing is formulated as follows.

$$(\mathcal{P}): \max_U \quad \text{s.t.} \quad (3)-(7), \ (9) \quad \text{and,} \quad \begin{align*}
 u_i^s & \geq 0, \forall m_i \in M, \\
 u_j^b & \geq 0, \forall n_j \in N, \\
 x_{ij} & \in \{0,1\}, \forall m_i \in M, \ n_j \in N,
\end{align*}$$

(13)

where the QoE of both drivers and passengers are guaranteed by constraints (3)-(7). Constraint (9) indicates that each passenger can be allocated to at most one driver. Constraints (13a) and (13b) ensure that the benefits of drivers and passengers are non-negative. Constraint (13c) represents the binary decision variable.

Theorem 1. The problem $\mathcal{P}$ is NP-hard.

Proof. We prove the theorem with a reduction from the weighted set packing problem (WSP) [10]. The WSP problem can be described as follows: given a universe $U$ with elements $v \in U$ and a finite collection of subsets $S = \{S_1, S_2, \ldots, S_l\}$, where each subset $S_i \subseteq U$ is associated with a weight $w(S_i)$. The goal is to select some disjoint subsets $S^* \subseteq S$ such that the total weight $\sum_{S_i \in S^*} w(S_i)$ is maximized.

Given a WSP instance, we can transform it to a $\mathcal{P}$ instance within polynomial time. Let each subset $S_i \subseteq U$ contains one driver and multiple passengers satisfying the QoE constraints (3)-(7), where $U = M \cup N$. The weight of each subset $S_i$ represents the utility of passengers in $S_i$. Then, for this $\mathcal{P}$ instance, we want to select some disjoint subsets such that the overall utility of passengers is maximized. Since the WSP problem is known to be NP-hard, the problem $\mathcal{P}$ is also NP-hard.

III. ALGORITHMS

Since problem $\mathcal{P}$ defined in the previous section is NP-hard, it is not wise to derive the optimal solution when there are large number of both drivers and passengers in the network system. Because it will require a lot of running time to obtain one possible solution with almost exponentially time complexity growth with the network scale. Besides, the utility of the driver should be guaranteed while meeting the budgets requirement of passengers. Therefore, we try to derive a lightweight solution in the aspect of network economy. To meet the budget constraints of passengers and at the same time without hurting the utilities of drivers, we use coalitional auction game to solve the ride-sharing problem.

A. Coalition Formation Approach

In this subsection, we first list some basic definitions used in the coalitional formation, then we prove that the proposed algorithm can converge to a stable coalition partition within limited iterations.

Definition 1 (Coalition). A coalition $C_i$ is termed as a group of passengers assigned to the same driver $m_i$.

Definition 2 (Coalition Structure). Coalition structure $C = \{C_0, C_1, \ldots, C_M\}$ is defined as the partition of passengers among all drivers in $M$, where $C_0$ denotes a virtual coalition formed by the passengers who fail to join shared journeys.

The coalition structure must satisfy the following constraints:

$$C_m \cap C_n = \emptyset, \ m \neq n, \ \forall m, n \in [0, \ldots, M],$$

(14a)

$$\bigcup_{i=0}^{M} C_i = N,$$

(14b)

where constraint (14a) means that the intersection of any two coalitions is empty and constraint (14b) means the union of all coalitions is the set of passengers.

Given the above definitions, we propose a coalition formation approach (CFA), which is conducted by the auctioneer. The auctioneer iteratively adjusts the coalition partition $C$ via performing possible coalition operations on passengers, aiming to maximize the utility of them. The possible coalition operations are summarized as follows:

- Joining operation: Passenger $n_j$ in the virtual coalition $C_0$ selects a coalition $C_i$ ($i \neq 0$) from its candidate coalitions to join in.
- Switching operation: Passenger $n_j$ move from the coalition $C_{i_1}$ to another coalition $C_{i_2}$ ($i_1, i_2 \neq 0, i_1 \neq i_2$). It means that passenger $n_j$ leaves the current coalition $C_{i_1}$ to join coalition $C_{i_2}$.
### Algorithm 1: CFA /* Coalition Formation Approach */

**Input:** \( \mathcal{M}, \mathcal{N}, \mathcal{A}, B \);

**Output:** \( \mathcal{C} = \{C_0, C_1, \ldots, C_M\} \);

1. Initialize an initial coalition partition \( \mathcal{C} \);
2. \( U \leftarrow \text{SPS}(\mathcal{C}, \mathcal{M}, \mathcal{A}, B) \);
3. /* \( \tau \) is the maximum round number */
4. for \( \xi \leftarrow 1 \) to \( \tau \) do
5.   for \( j \leftarrow 1 \) to \( N \) do
6.     if \( n_j \in C_0 \) then
7.       Perform a joining operation for passenger \( n_j \) to obtain a new coalition partition \( \mathcal{C}' \).
8.       \( U' \leftarrow \text{SPS}(\mathcal{C}', \mathcal{M}, \mathcal{A}, B) \);
9.       if the joining operation is feasible according to feasible rule (See Definition 3) then
10.      \( \mathcal{C} \leftarrow \mathcal{C}', U \leftarrow U' \);
11.   end if
12.   else
13.     Perform a switching operation for passenger \( n_j \) to obtain a new coalition partition \( \mathcal{C}' \).
14.     \( U' \leftarrow \text{SPS}(\mathcal{C}', \mathcal{M}, \mathcal{A}, B) \);
15.     if the switching operation is feasible then
16.       \( \mathcal{C} \leftarrow \mathcal{C}', U \leftarrow U' \);
17.     else
18.       Perform a leaving operation for passenger \( n_j \) to obtain a new coalition partition \( \mathcal{C}' \).
19.       \( U' \leftarrow \text{SPS}(\mathcal{C}', \mathcal{M}, \mathcal{A}, B) \);
20.       if the leaving operation is feasible then
21.         \( \mathcal{C} \leftarrow \mathcal{C}', U \leftarrow U' \);
22.     end if
23.   end if
24. end if
25. end for
26. end for
27. return \( \mathcal{C} \).

- Leaving operation: Passenger \( n_j \) move from the coalition \( C_i \) (\( i \neq 0 \)) to the virtual coalition \( C_0 \).

Similar to the transfer rule defined in [11], we define the feasible rule of coalition operations as follows.

**Definition 3 (Feasible rule).** In the proposed coalition formation approach with current partition \( \mathcal{C} \), a coalition operation for passenger \( n_j \) is feasible if a higher total utility of passengers in \( \mathcal{C}' \) is achieved.

The details of CFA algorithm are described in Algorithm 1. The auctioneer first initializes a coalition partition, which satisfies the QoE of both drivers and passengers. Then an iterative process is executed for passengers to adjust the coalition partition. If a passenger is in the virtual coalition \( C_0 \), the auctioneer tries to perform a joining operation for the passenger. Otherwise, if a passenger is not in the virtual coalition \( C_0 \), the auctioneer tries to perform a switching operation or leaving operation for the passenger. If a coalition operation is feasible, the current coalition partition \( \mathcal{C} \) will be changed to a new coalition partition \( \mathcal{C}' \) with a higher utility of passengers. Finally, the iterative process terminates when no increment in total passenger utility can be achieved.

**Theorem 2.** Algorithm CFA can converge to a stable coalition partition within limited iterations.

**Proof.** Starting from an initial partition \( C_0 \), the coalition formation process can be seen as a sequence of coalition operations to adjust the coalition structure, e.g.,

\[ C = C_0 \rightarrow C_1 \rightarrow \cdots \rightarrow C_k \rightarrow C_{k+1} \rightarrow \cdots \]

where \( C_k \) is a partition including \( M \) coalitions after \( k \) feasible coalition operations, as shown in Lines 6 to 21 of Algorithm 1. Based on the feasible rule, we know that the total utility of passengers will not decrease. That is, for every two sequential partitions \( C_k \rightarrow C_{k+1} \) in the sequence, we have

\[ \sum_{C_i \in C_{k+1}} \sum_{n_j \in C_i} u^b_j \geq \sum_{C_i \in C_k} \sum_{n_j \in C_i} u^b_j \quad (15) \]

Thus, each possible partition occurs at most once in the sequence. Given the number of passengers and drivers, the total number of possible partitions is finite. This guarantees that the proposed algorithm can converge to a final coalition partition \( \mathcal{C}^* \).

In the following, we prove that the final coalition partition \( \mathcal{C}^* \) is a stable partition. Assume that \( \mathcal{C}^* \) is not stable. This indicates that there exists another coalition partition with a higher total utility of passengers. Then, the current partition will change to a new coalition partition via performing feasible coalition operations. However, it contradicts the fact that \( \mathcal{C}^* \) is the final coalition partition. Thus, \( \mathcal{C}^* \) must be stable. ■

### B. Secondary Pricing based Strategy

Although VCG-based auctions can guarantee the truthfulness of bidders, it is generally intractable to find the optimal allocation in polynomial time if the problem is NP-hard. Furthermore, the auctioneer (the ride-sharing service platform operator) may get a negative utility based on the variants of VCG auction [12], violating the property of budget balance. Therefore, we design a secondary pricing based strategy (SPS), which is proved to be truthful, budget balance, individual rational and computational efficient. This strategy is used to determine the winning drivers and passengers and to calculate payoffs and payments for them.

The set of asks submitted by drivers is denoted as \( \mathcal{A} = \{A_1, A_2, \ldots, A_M\} \) and the set of bids placed by coalitions is denoted as \( \mathcal{B} = \{B_1^b, B_2^b, \ldots, B_M^b\} \). The bid of coalition \( C_i \) is defined as:

\[ B_i^b = |C_i| \cdot \eta_i^b, \quad \forall i \in [1, \ldots, M], \]

where \( \eta_i^b \) is the minimum bid placed by the passengers in coalition \( C_i \), defined as follows:

\[ \eta_i^b = \min\{b_j^b | n_j \in C_i\}, \quad \forall j \in [1, \ldots, N]. \]

The main of the secondary pricing based strategy is shown in Algorithm 2. Initially, the auctioneer sorts the coalitions in \( \mathcal{C} \) to obtain an ordered list \( \mathcal{C} = \{C_1, C_2, \ldots, C_M\} \) such that \( \eta_{i_1}^b \geq \eta_{i_2}^b \geq \cdots \geq \eta_{i_M}^b \).
Next, the auctioneer retrieves the coalitions in $\mathcal{C}$ one by one. For each coalition $C_{i_k}\in\mathcal{C}$, the auctioneer judges that if the bid of the coalition $B_{i_k}^b$ can satisfy the price asked by the corresponding driver $m_{i_k}$, as shown in Line 5 of Algorithm 2, where $\kappa$ is the profit ratio set by the auctioneer. Otherwise, the driver $m_{i_k}$ and the coalition $C_{i_k}$ are designated as losers. Then, as shown in Line 6 of Algorithm 2, the auctioneer tries to find the minimum payment for each passenger in $C_{i_k}$ to afford the cost of shared journey, i.e.,

$$P_{i_k}^b = \arg \min_{\left\{ \eta_k^b : k+1 \leq g \leq M \right\}} \{ C_{i_k} \cdot \eta_k^b \cdot (1 - \kappa) \geq A_{i_k}^s \} \quad (18)$$

When the minimum payment $P_{i_k}^b$ for each passenger in coalition $C_{i_k}$ exists, the coalition $C_{i_k}$ and the driver $m_{i_k}$ are selected as winners. The payment paid by each passenger in the winning coalition $C_{i_k}$ is given by

$$p_{i_j}^b = P_{i_k}^b, \quad \forall n_{j} \in C_{i_k}. \quad (19)$$

The payoff obtained by the winning driver $m_{i_k}$ is given by

$$p_{i_k}^s = A_{i_k}^s, \quad i_k \in [1, \ldots, M]. \quad (20)$$

Theorem 3. SPS is truthful.

Proof. For each passenger $n_{j}$, there are two cases as follows.

1) When passenger $n_{j}$ wins with truthful bidding, the payment is $p_{i_j}^b$. If passenger $n_{j}$ bids untruthfully with $b_{i_j}^b < \bar{b}_j$, the payment of passenger $n_{j}$ is not changed, thus the utility is not changed; if passenger $n_{j}$ bids untruthfully with $b_{i_j}^b > \bar{b}_j$, the passenger $n_{j}$ may either win with no change in utility or lose in the auction.

2) When passenger $n_{j}$ loses with truthful bidding, the utility is zero. If passenger $n_{j}$ bids untruthfully with $b_{i_j}^b < \bar{b}_j$, the passenger $n_{j}$ still loses in the auction; if passenger $n_{j}$ bids untruthfully with $b_{i_j}^b > \bar{b}_j$, the passenger $n_{j}$ may win with a non-positive utility or lose in the auction.

Therefore, each passenger $n_{j}$ cannot improve his/her utility by submitting an untruthful bidding.

For drivers, it seems possible for them to improve their utilities by increasing the ask price when the ask price is not greater than the total bid of the corresponding coalition. Since drivers know nothing about which passengers they will be matched with before submitting the asks, the opportunity for them to improve their utilities is very limited. So it is sensible to consider it is truthful for drivers. 

Theorem 4. SPS is individual rational and budget balance.

Proof. For each passenger $n_{j}$ in the winning coalition $C_i$, we know that

$$p_{i_j}^b = P_{i_k}^b \leq \eta_i^b = \min\{\eta_i^b : n_{j} \in C_i \} \leq b_{i_j}^b = v_{i_j}^b \quad (21)$$

Thus, $u_{i_j}^b = v_{i_j}^b - p_{i_j}^b \geq 0$. To prove the rationality of drivers, it only needs to guarantee that the maximum tolerable detour ratio of drivers is not greater than 1.

Next, to prove the property of budget balance is equivalent to guarantee $P_{i_k}^b \cdot |C_i| \geq P_i^s, \forall m_i \in \mathcal{M}$. For each winning driver $m_i$ and coalition $C_i$, we have

$$P_{i_k}^b \cdot |C_i| \geq A_i^s = P_i^s \quad (22)$$

Hence, this theorem holds.

Theorem 5. SPS is computational efficient.

Proof. In Algorithm 2, we check at most $M$ pairs of drivers and coalitions. And it takes time $O(M)$ to find the lowest minimum payment for each coalition. Therefore, the time complexity of Algorithm 2 is $O(M^2)$.

IV. PERFORMANCE EVALUATION

In this section, we first detail the simulation setup, which includes the data-set used in the experiment, the baseline algorithm and the evaluation metrics. Then we describe the experimental results with full explanations.

A. Simulation Setup

Dataset. We conduct extensive simulations to evaluate the performance of the proposed algorithm CFA. It is devised based on the real data from taxi trajectory dataset in Beijing city [13], which contains 9 million kilometers taxi GPS trajectories within the 5th Ring Road of Beijing city from February 2 to February 8, 2008. However, the dataset cannot be directly applied to our experiment due to the travel demands of drivers in our work. By using the approach proposed in [14], we mine the taxi trajectories to generate a simulated dataset containing travel demands of both drivers and passengers for simulation. The experiment configurations and parameters are
The matching ratio of passengers (MRP) is defined as the ratio between the number of winning passengers and the total number of passengers.

2) The matching ratio of drivers (MRD) is defined as the ratio between the number of winning drivers and the total number of drivers.

3) The passenger saving rate (PSR) is defined as the ratio between the total utility of passengers and the total valuation of winning passengers.

B. Experimental Results

Fig. 3 shows the relationships between matching ratio of passengers and variable maximum detour ratios of drivers, between matching ratio of drivers and variable maximum detour ratios of drivers as well as between passenger saving rate and variable maximum detour ratios of drivers. We can see that algorithm CFA outperforms algorithm TSG in terms of MRP, MRD and PSR. As shown in Fig. 3(a) and Fig. 3(b), both MRP and MRD increase with the growth of the maximum detour ratio of drivers, indicating that the larger maximum detour ratio is, the more passengers and drivers will participate in the ride-sharing. In Fig. 3(a) algorithm CFA outperforms algorithm TSG by 16.4% ~ 18.7% on matching ratio of drivers. In Fig. 3(b) algorithm CFA outperforms algorithm TSG by 13.1% ~ 14.6% on matching ratio of drivers. Moreover, Fig. 3(c) depicts passenger saving rate of the two algorithms on variable maximum detour ratios of drivers. We can see that PSR increases with the increase of the maximum detour ratio of drivers because detour costs incurred in ride-sharing are borne by drivers and the buyout ask prices set by drivers are fixed. With different driver maximum detour ratios, the passenger saving rates of CFA and TSG are 55.0% and 38.7% on average. The reason why CFA outperforms TSG is that algorithm CFA can provide more opportunities for passengers to match with drivers via possible coalition operations, thus leading to more fare reductions.

The performance of running time for the two algorithms with different passenger’s maximum waiting time and driver’s maximum detour ratios are shown in Fig. 4. In Fig. 4(a), from a holistic perspective, the running time of CFA is relatively stable, while TSG increases linearly. The running time of algorithm CFA is higher than that of TSG at the beginning, and with the increase of passenger’s maximum waiting time, the running time of algorithm TSG exceeds the running time of algorithm CFA when maximum waiting time is larger than 4 minutes. In Fig. 4(b), the running time of both algorithm CFA and algorithm TSG increase as the maximum detour ratio increases, and TSG increases faster than CFA. The main reason is that the larger maximum detour ratio is, the more

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>EXPERIMENT CONFIGURATIONS AND PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameters</td>
<td>values</td>
</tr>
<tr>
<td>Number of drivers $M$ and passengers $N$</td>
<td>300, 600</td>
</tr>
<tr>
<td>The maximum quantity of available seats in a vehicle $q^b_i$</td>
<td>5</td>
</tr>
<tr>
<td>The quantity of required seats $q^p_j$</td>
<td>$[1, 3]$</td>
</tr>
<tr>
<td>The velocity of vehicle $v$</td>
<td>48 km/h</td>
</tr>
<tr>
<td>Price per minute $\phi^b_j$</td>
<td>0.4 $/\text{min}$</td>
</tr>
<tr>
<td>Price per kilometer $\phi^p_j$</td>
<td>2.5 $/\text{km}$</td>
</tr>
<tr>
<td>Labor cost per minute $\nu_i$</td>
<td>0.3 $/\text{min}$</td>
</tr>
<tr>
<td>Fuel cost per kilometer $\rho_i$</td>
<td>2 $/\text{km}$</td>
</tr>
<tr>
<td>The searching radius of drivers $R$</td>
<td>5 km</td>
</tr>
<tr>
<td>Maximum tolerable waiting time $\Delta^b_i$</td>
<td>2, 3, 4, 5, 6 minutes</td>
</tr>
<tr>
<td>Maximum tolerable detour ratio $a^b_i, a^p_j$</td>
<td>0.2, 0.3, 0.4, 0.5, 0.6</td>
</tr>
<tr>
<td>The profit ratio set by the auctioneer $\kappa$</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Algorithm TSG uses the McAfee based auction in [16] to calculate the payoff for each driver and the payment for each passenger.

Evaluation Metrics. We evaluate the performance between algorithms CFA and TSG in terms of the following metrics: matching ratio of the passenger (driver), passenger saving rate and algorithm’s running time.

1) The matching ratio of passengers (MRP) is defined as the ratio between the number of winning passengers and the total number of passengers.

2) The matching ratio of drivers (MRD) is defined as the ratio between the number of winning drivers and the total number of drivers.

3) The passenger saving rate (PSR) is defined as the ratio between the total utility of passengers and the total valuation of winning passengers.
packings are generated, and TSG needs to spend more time in the first stage to find the packing with the highest score for drivers.

V. CONCLUSION

Ride-sharing has been investigated in this paper, where the matching process is formulated as an optimization problem to maximize the total utility of passengers, while guarantee the interests of both passengers and drivers. The proposed problem has been proved to be NP-hard, and an efficient algorithm based on coalitional double auction has been developed. The coalition formation approach has been proved that it can converge to a stable coalition partition and the proposed strategy also has been proved to be truthful, budget balance, individual rational and computational efficient. Moreover, simulation results have demonstrated that the proposed algorithm outperforms the designed baseline algorithm in terms of various metrics. In the future, we will use machine learning to help derive optimal system parameters with real-world experiments to deep into the algorithms’ performance.

ACKNOWLEDGEMENT

This work was supported in part by Key-Area Research and Development Program of Guangdong Province under Grant 2019B0101212001. It was also supported in part by the National Natural Science Foundation of China under Grant No. 62072118 and No. 61702115. The corresponding author is Long Chen.

REFERENCES


