Dynamic Matching for Ride-sharing with Deadlines

Shuqin Gao[†], Costas Courcoubetis[†], and Lingjie Duan[*]
[†]School of Data Science, The Chinese University of Hong Kong, Shenzhen
[*]Engineering Systems and Design Pillar, Singapore University of Technology and Design
Email: gaoshuqin@cuhk.edu.cn, costas@cuhk.edu.cn, lingjie_duan@sutd.edu.sg

***Abstract*—Ride-sharing systems, which enable private car drivers to offer rides to passengers heading in the same direction, have gained popularity as a cost-effective urban transportation solution. We consider the market design problem, where drivers and riders dynamically enter the system over time, requiring transportation to specific destinations within certain deadlines. This study focuses on a simple but representative case, wherein all agents, i.e., drivers and riders, share a common origin but possess different destinations that define their types. Agents exhibit flexibility in their willingness to wait for a match, and matching agents of the same type generates higher utility than matching different types. Initially, we address the full information scenarios where agents reveal their true types to the platform and must accept the matching proposed by the platform. Given that the optimal matching for this problem requires solving an MDP with exponential complexity, we propose a computationally simple mechanism that matches a different-type pair only when one agent is about to leave immediately, and the other cannot wait beyond a threshold number of time periods. We theoretically prove that this mechanism has near-optimal performance, particularly under balanced demand and supply. Subsequently, we shift focus to the incentive properties of the mechanism in more practical scenarios where selfish agents can be non-cooperative with the platform, i.e., reject low-value different-type matchings hoping to get better ones in the future, and non-truth-telling, i.e., declare a different destination. We study the incentive compatibility of two interesting payment schemes and prove the Price-of-Anarchy.**

***Index Terms*—Ride-sharing, dynamic matching, online mechanism, price of anarchy**

I. Introduction

The swift evolution of mobile technologies and applications has brought about a surge in the popularity of ride-sharing systems such as GrabHitch [1] as an economical mode of urban transportation. Ride-sharing enables private car drivers to offer rides to passengers sharing similar itineraries and time schedules, yielding significant reductions in travel expenses while holding promise for alleviating urban traffic congestion. Within the realm of ride-sharing systems, considerable attention has been directed towards enhancing the efficacy of matching riders with appropriate drivers to maximize the benefits of sharing [2]–[5], as well as devising pricing strategies for these services [6], [7]. This matching process is inherently dynamic, as customer requests and driver availability fluctuate over time, thus necessitating a dynamic framework for optimal matching [8]–[11]. While efforts have been made to study an optimal online matching mechanism, notably by the authors in [10], their assumption that participants can indefinitely wait in the system without a deadline for matching does not align with the operational reality of ride-hailing systems. Indeed, a survey revealed that each passenger using Uber Pool is bounded by a waiting-time limit of 10 minutes [2]. This incongruity highlights the need for further investigation of dynamic matching mechanisms that appropriately consider real-world operational constraints in ride-sharing systems.

In this work, we explore a ride-sharing platform that operates in the long run and model a realistic scenario in which both riders and drivers will exit the system if they cannot be matched within their specified deadlines upon arrival. This practical consideration significantly complicates the development of the matching mechanism, as it requires the comprehensive assessment of all potential remaining availabilities of both existing riders and drivers at the time of pairing. Consequently, a driver may not be matched with a new or future rider heading to the same destination, but rather with another rider who is nearing their departure from the system.

We are the first to study this fundamental tradeoff between matching a pair with the same type and matching a different-type pair on the verge of departure for the system's decision. We begin by modeling the dynamic matching process as a Markov Decision Process (MDP). In each period, the 'state' of the market is characterized by the types of unmatched drivers and riders who have not met their deadlines. Then, we derive the optimal matching mechanism by solving this MDP using linear programming techniques. However, despite our efforts to reduce the action space for the MDP by demonstrating that the platform will only match a driver and rider when at least one of them is about to depart immediately, this process still presents formidable complexities.

To further reduce the mechanism's complexity, we propose a simple threshold-based heuristic mechanism that matches a same-type pair once it appears and matches a different-type pair only when one agent is imminently departing and his partner can only stay for a threshold number of time periods. Thus, the primary challenge of the heuristic mechanism lies in establishing the optimal thresholds, which can be done by exhaustive search. We prove that the optimal thresholds returned by the heuristic mechanism decrease as the extra cost incurred by different-type matching increases. Our numerical experiments confirm this trend and further show how these optimal thresholds vary with the deadline duration. Theoretically, we prove that this mechanism has near-optimal performance, especially when the flows of drivers and riders heading to each destination are balanced.

There is another challenge that agents capable of staying

ISBN 978-3-903176-65-2 © 2024 IFIP

more periods may selfishly decline the system's recommendation to match a different-type partner. Additionally, a strategic driver could misreport his private destination type to obtain some potential extra compensation by the platform for serving a different-type rider, although in reality they actually share the same destination. To handle these challenges, we first propose some simple pricing schemes and study their incentive compatibility properties regarding type declarations, assuming that non-cooperation is absent. We then extend our analysis to scenarios involving non-cooperation and demonstrate that these pricing schemes can still incentivize truthful type reporting when destinations are symmetric for both drivers and riders. Finally, we analyze the Price-of-Anarchy stemming solely from non-cooperation in the equilibrium.

The paper is organized as follows. Section II presents our ride-sharing platform model that operates over time. In Section III, we formulate the dynamic matching process and study the optimal matching mechanism. Section IV proposes a simple heuristic mechanism and theoretically analyzes its performance. Section V studies the incentive design for the heuristic mechanism and Section VI concludes the paper.

II. System Model

Our goal is to formulate and analyze several critical issues within a ride-sharing market, where a driver (the supply side) offers transportation service to riders (the demand side). To make the problem more tractable, we concentrate on a subset of the market in which both drivers and riders originate from the same location and are randomly heading towards two distinct locations, denoted as A and B, as depicted in Figure 1 for a one-shot example.[1] This scenario reflects instances when the origin location is a residential area, and locations A and B represent two key points in the Central Business District (CBD). We specifically consider morning traffic, during which all trips are directed to the CBD[2].

To simplify the cost structure, we denote that when a driver shares his vehicle with a rider who has the same destination, the driving cost is zero as the driver would need to make this trip regardless. When the driver shares his vehicle with a rider who has a different destination, the driver is committed to first drive the rider to his destination and then drive himself to his own destination. The driving cost associated with the additional travel between the two destinations is denoted as $c > 0$. Furthermore, we assume for simplicity that a driver can share his vehicle with a single rider. The value derived from the service is normalized as 1 when a rider reaches his destination. To avoid trivial matching scenarios, we set $0 < c < 1$ to ensure a strictly positive utility for matching a driver and a rider, even when they have different destinations.

In our model of the matching driver-rider market, we assume an infinite-horizon dynamic model where drivers and riders arrive in discrete time, consisting of equal-length periods. We

[1]Our analysis can be extended to encompass multiple destinations; however, for the sake of clarity, we will confine our discussion to this simpler scenario.

[2]We can conduct independent analysis for different outlying residential areas due to the potentially high travel costs between them.

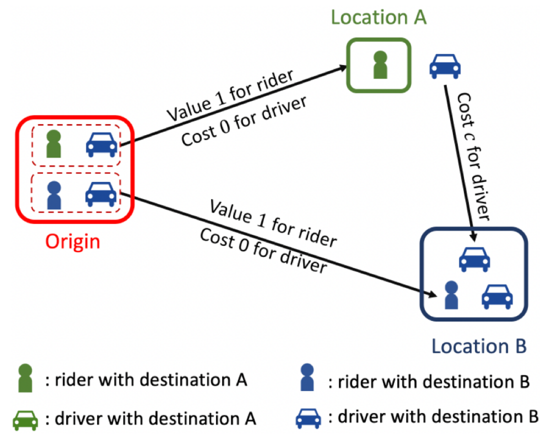

Fig. 1: A simple instance of the ride-sharing model. There are two matched pairs of drivers and riders: one pair has the same destination (both go to B), and the other pair has different destinations (rider with A and driver with B). The driver in the later pair needs to travel to B after dropping off the matched rider at location A, incurring extra driving cost c.

assume (to keep the model simple) that in each period a new pair of a driver and a rider enter the market[3] and declare their destinations. Each of them is assumed to have a waiting time window of the same length $T \geq 0$ periods, i.e., drivers/riders who arrive at period t wait in the market to be matched up to period $t + T$, and leave the market if not matched by that deadline. We introduce randomness by assigning randomly their destinations (types). Each arriving driver is of type A with probability p and B with probability $1 - p$. Each rider is of type A with probability q and B with probability $1 - q$. In the full information case, agents declare their true type to the platform. We will later analyze the scenario where agents might not be truthful.

We present an illustrative instance of the matching process in our ride-sharing model, covering periods 0 to 3 as depicted in Fig. 2. In each period, we match all available driver-rider pairs of the same type and consider matching pairs of different types only when at least one of them is imminently departing[4]. Initially, a blue type-B rider and a green type-A driver arrive (with probability $(1 - q)p$) at time 0, and both are capable of staying for $T = 2$ periods. As they belong to different destination types, we do not match them. But in period 1, a new green rider and a new green driver arrive, and the new green rider is matched with the existing same-type green driver who arrived at time 0 and has less remaining available time. Then in period 2, with the new arrivals, we have two blue riders and two green drivers in the waiting pool. At that point, the blue rider who arrived at time 0 is about to leave immediately, thus we match this rider with the green driver possessing the smallest remaining available time.

[3]Our analysis can be extended to accommodate an arbitrary number of arrivals in each period, although it becomes more complex.

[4]This policy is one of the possible implementations of the threshold-based heuristic matching mechanism, which will be discussed in detail in Section IV. Here, we employ it for illustrative purposes.

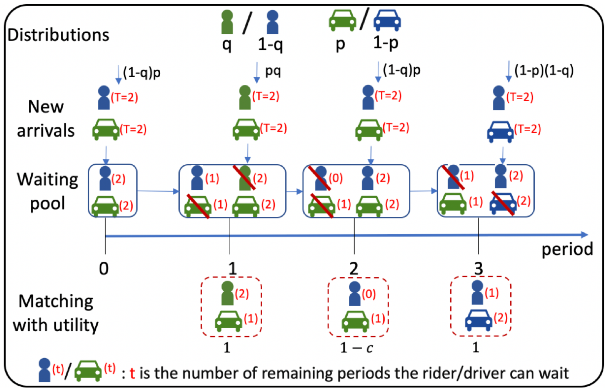

Fig. 2: An illustrative instance of matching in our ride-sharing market from period 0 to period 3. In each period, one driver and one rider enter the market, and they can be either type A or type B. Each of them is assumed to stay in the market for $T = 2$ time periods.

The matching utility for the same-type pair of a driver and a rider is 1 and the matching utility for the different-type pair is $1 - c$. Note that, to maximize the total matching utility, the platform needs to match as many pairs of drivers and riders as possible while minimizing the mismatching cost. However, such a matching mechanism is myopic and does not take into account the matching opportunities for future arrivals. We need to carefully model the dynamic matching process and analyze the optimal mechanism as demonstrated in Section III.

III. Formulation and Analysis of Optimal Matching Mechanism

A. Matching Process Formulation

Our matching process depends on the 'state' of the market. We denote by $\boldsymbol{s}_t \in S$ the market state, i.e., types of drivers and riders available for matching at the end of period t just before we make our matching decision (think that the driver/rider pair arrives during the period and the decision is taken at the end of the period). We define the market state at period t as $\boldsymbol{s}_t = \{s_t^0, s_t^1, \ldots, s_t^T\}$ where each component s_t^i is given by a 4-tuple $(\alpha_t^i, \bar{\alpha}_t^i, \beta_t^i, \bar{\beta}_t^i) \in \{0,1\}^4$ to describe the numbers of (type A drivers, type A riders, type B drivers, type B riders) that are currently available in the market and can wait for exactly i more periods counting from the current time t. Note that the driver/rider who can wait for i more periods must have arrived at time $t + i - T$. Thus, $s_t^0 = (\alpha_t^0, \bar{\alpha}_t^0, \beta_t^0, \bar{\beta}_t^0)$ tells us the types of a driver and/or a rider that arrived at time $t - T$, remain in the system, and will leave the system immediately if not matched in the current period. s_t^T refers to the types of the driver and the rider who newly arrived in the current period.

Since in every period, only one pair of driver/rider enters the market, we must have $\alpha_t^i + \beta_t^i \leq 1, \forall t, i \leq T$ and $\bar{\alpha}_t^i + \bar{\beta}_t^i \leq 1, \forall t, i \leq T$. Therefore, for any given t and $i < T$, one can easily check that there are totally 9 possible values[5] for s_t^i. For $i = T$, s_t^T can take four values, since there is always a driver and a rider that arrived. This implies that the number $|S|$ of possible values for a state $\boldsymbol{s}_t$ is $9^T \times 4$.

In our online market model, the platform should decide the matching at time t based on $\boldsymbol{s}_t$. Let $\pi(\boldsymbol{s}_t)$ denote the online matching decision as a function of $\boldsymbol{s}_t$. For any two given states $\boldsymbol{s}_t$ and $\boldsymbol{s}_{t+1}$, the transition probability from $\boldsymbol{s}_t$ to $\boldsymbol{s}_{t+1}$ only depends on the system parameters (i.e., the probabilities p, q of destination types and the length T of waiting time window) and the matching mechanism $\pi(\cdot)$. More specifically, for $i < T$, s_{t+1}^i is uniquely determined by the sub-state s_t^{i+1} in the previous period and whether a driver/rider available in s_t^{i+1} is matched under the mechanism at period t and hence removed from s_t^{i+1}. For $i = T$, s_{t+1}^T does not depend on the mechanism and is determined by the types of driver/rider that arrive in period $t + 1$, e.g., $s_{t+1}^T = (1, 1, 0, 0)$ implies a type A driver and a type A rider arrived, which happens with probability pq. Recall that the decision in period $t + 1$ will be taken after the driver and the rider of that period have arrived in the market.

This structure defines an MDP where the states are all possible market states given in set S, and actions in each state are all the possible driver/rider matchings consistent with the state. There are no restrictions on the number of simultaneous matchings in each period. Moreover, from each state, with any given action, 4 possible transitions are depending on the possible type combinations of the newly arrived rider and driver, and the corresponding probabilities. Each action incurs immediate utility 1 for each matched same-type pair and utility $1 - c$ for each matched different-type pair. A deterministic mechanism $\pi(\cdot)$ corresponds to choosing in each state a unique matching action and our objective is to find the optimal one that maximizes the total expected utility per period.

B. Structure of Optimal Mechanism

The number of theoretically possible matching actions in each state is excessively large, involving all possible matching options (including remaining unmatched) for up to T+1 drivers and T+1 riders. Next, we will show that the space of optimal actions in each state can be reduced to 7 regardless of T.

Lemma 1. *There are three essential properties in the optimal mechanism: (i) match a driver and a rider only when one of them is about to leave immediately; (ii) match a departing agent with a complementary agent*[6] *only when there exists no other same-type complementary agent with shorter residual available time; (iii) when both a driver and a rider are about to leave in the current period, then a) always match them if they are of the same type, or b) if they are of different types, then either b1) create a* single matching *by either matching them together or one of them departs unmatched and the other participates in a same-type matching, or b2) create* two matchings *where each of them is matched with a same-type complementary agent who is not ready to depart.*

[5] Which are $(0,0,0,0)$, $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$, $(1,1,0,0)$, $(1,0,0,1)$, $(0,1,1,0)$, and $(0,0,1,1)$.

[6] From here on, an agent refers to a rider or a driver. If an agent is a driver, its complementary agent refers to a rider.

The first property is because we can wait and match them later, giving us more options. The second property is because such a matching will produce the same immediate value but offer more flexibility to the future, and this can only increase the expected utility. The third property part a) is trivial, since this driver/rider pair is the best choice for matching, and the reward will be lost if we don't take the action without affecting the other actions that are possible in the future. Part b) avoids 'rushed' decisions, where the same or more reward could be obtained eventually in the system than if at the present time we decided to match only the two departing agents together.

Based on this lemma, we can further restrict the number of possible actions by the optimal mechanism in each state of the market. There is a single available action if no driver/rider departs immediately, or if there is an immediately departing pair of same-type driver and rider. Otherwise, a mechanism action consists of a leaf in the following decision tree, which can be defined independently of the state to which it applies.

1) Level 1: Decide to create a matching or not (Yes/No). (2 decisions)
2) Level 2: If Yes, choose to match only a departing driver, or only a departing rider, or both. (3 decisions).
3) Level 3: If the choice was for matching only a single departing agent, choose to match with the same or different types (2 decisions). If the choice was for matching both departing agents, then choose i) match them together, or ii) match them with same-type complementary agents (2 decisions).

One can check that this decision tree has 7 leaves, which define the largest set of available mechanism actions that can be considered at any state. Each state might reduce the set of available actions. For example, if there is a single departing driver, only the actions corresponding to: 'don't match anybody', 'do matchings and match the driver with a same-type rider', or 'do matchings and match the driver with a different-type rider', are available. Observe also that in certain states, some actions are dominated by others. For example, when there are a driver and a rider departing immediately, the action of not creating any matching is dominated by matching them together. We are not concerned with further optimizing the choice of actions in the different states, since our goal is to show that a small number of actions independent of T is sufficient to specify the optimal mechanism.

Since the number of possible states is $9^T \times 4$ and the number of actions is 7, standard MDP complexity results suggest the following result.

Lemma 2. *For any system state $\boldsymbol{s}_t$, there are at most $|A| = 7$ possible policies to consider. Assuming a fixed number of bits to describe p, q, the complexity of solving this MDP using linear programming techniques is polynomial in $|S| \times |A| = O(9^T)$, i.e., exponential in T.*

To further reduce the mechanism's complexity so that it can be used as an online algorithm, we propose a heuristic matching mechanism that has high efficiency and low complexity.

IV. A Heuristic Matching Mechanism

We propose a simple heuristic mechanism for choosing the best matching that uses a threshold-based rule on the residual time of agents in the market, reducing the number of feasible matching pairs. We prove that the heuristic has near-optimal performance, especially in the symmetric case of balanced demand and supply, where the arriving drivers and riders have equal probabilities to be of the given types (i.e., $p = q$).

A. Design of the Mechanism

Let $(x, y), x \in \{d, r\}, y \in \{A, B\}$ describe the characteristics of an agent (i.e., its full type: driver or rider, and its destination), and $\bar{x}$ ($\bar{y}$) denote the complement of x (y). In this heuristic mechanism, we follow the following rules, where the first applies with priority.

- Rule 1: If an agent (x, y) arrives and there is an agent $(\bar{x}, y)$ already in the system, then these two agents are matched immediately. If multiple $(\bar{x}, y)$ exist, we pick the one with the least residual time.
- Rule 2: If an agent (x, y) is about to depart and there is an agent of type $(\bar{x}, \bar{y})$ with the residual time less than (or equal to) the threshold $0 \leq \delta_{\bar{x},\bar{y}} \leq T$, then these two agents are matched. If multiple such $(\bar{x}, \bar{y})$ exist, we pick the one with the least residual time.

Our mechanism constantly applies these rules at each period, generating a unique outcome. Rule 1 gives priority to high-value (generate the maximum revenue of 1) matchings between drivers and riders that go to the same destination. Rule 2 tries to avoid low-value matchings for drivers and riders that go to different destinations if these are not necessary, and never matches a (say) departing driver with a rider that could stay in the system for much longer, since it could eventually contribute to a high-value matching. There is the following fundamental tradeoff between getting $1-c$ for sure and hoping to get 1 in the future with some probability (that depends on the mechanism). A higher value for each threshold value $\delta_{x,y}$ implies taking a lower risk, since we do a low value matching only if the non-departing agent has a low probability of being involved in a high-value matching.

We define now an optimization problem over the parameters $\Delta = \{\delta_{x,y}\}$ of the heuristic. We like to find the threshold values that maximize the average matching value generated by the mechanism. Note that each threshold $\delta_{x,y}$ can take values in $\{0, 1, \ldots, T\}$. This can be obtained by exhaustively comparing the average value obtained by all $(T+1)^4$ possible values for Δ. Notice that the search space reduces to $(T+1)^2$ when both drivers and riders have the same type distribution (i.e., $p = q$) and further reduces to $T + 1$ when $p = q = 0.5$. The problem we next address is how to search for the optimum $\delta_{x,y}$ values more efficiently, instead of using an exhaustive search. We prove the interesting monotonicity property that the optimal threshold values returned by the heuristic mechanism are decreasing in the cost c.

Lemma 3. *The optimal thresholds $\delta_{x,y}$ are decreasing in c.*

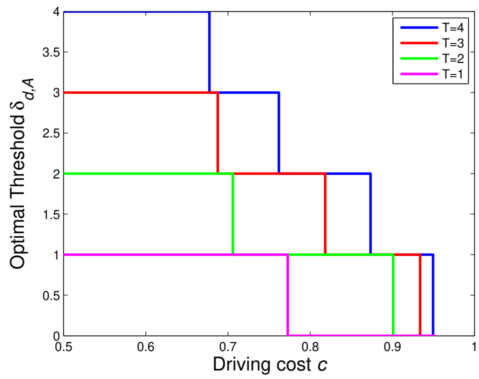

Fig. 3: The optimal threshold $\delta_{d,A}$ versus the extra driving cost c for different length T of waiting time window when $p = q = 0.5$.

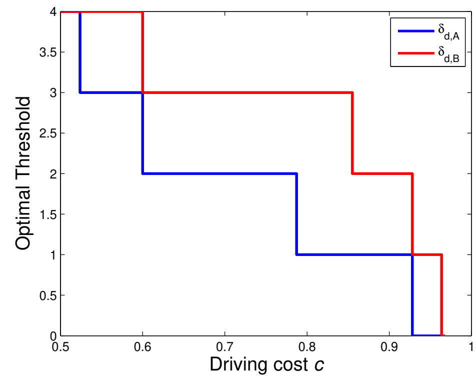

Fig. 4: The optimal thresholds $\delta_{d,A}$ and $\delta_{d,B}$ versus the extra driving cost c when $p = q = 3/4$ and $T = 4$.

Intuitively, a higher c makes the platform prefer matching more frequently same-type pairs. This happens when different-type pairs are matched more infrequently.

We next show numerically how the optimal thresholds change with the system parameters, i.e., the extra driving cost c and the probability p (q) of being type A for drivers (riders). First, we should note that when c is small (e.g., $c \to 0$), the heuristic mechanism with $\delta_{d,A} = \delta_{r,A} = \delta_{d,B} = \delta_{r,B} = T$ is close to the optimal mechanism. It is optimal to match all the available pairs, independently of their types, since these make almost no difference. When the extra driving cost c is large (e.g., $c \to 1$), the same holds with $\delta_{d,A} = \delta_{r,A} = \delta_{d,B} = \delta_{r,B} = 0$, since both of them match a different-type pair only when both the driver and the rider are about to leave immediately.

Therefore, our focus should be on the case of moderate values of c. In Fig. 3, we depict the optimal threshold $\delta_{d,A}$ against the extra driving cost c for different lengths T of waiting time window. Here, we consider the strongest symmetric case of $p = q = 0.5$ where both drivers and riders have equal probabilities to be type A or B. Therefore, the optimal thresholds returned by the heuristic mechanism must have $\delta_{d,A} = \delta_{r,A} = \delta_{d,B} = \delta_{r,B}$ and we illustrate $\delta_{d,A}$ only for simplicity. We observe that the optimal $\delta_{d,A}$ decreases as c increases. This is consistent with Lemma 3.

In Fig. 4, we further depict the optimal thresholds $\delta_{d,A}$ and $\delta_{d,B}$ against the extra driving cost c when $T = 4$. Here, we consider the asymmetric case of $p = q = 3/4$ where both drivers and riders have a higher probability to be type A than type B. Therefore we must have $\delta_{d,A} = \delta_{r,A}$, $\delta_{d,B} = \delta_{r,B}$, but $\delta_{d,A}$ might be different from $\delta_{d,B}$. We observe both $\delta_{d,A}$ and $\delta_{d,B}$ decreases as c increases and $\delta_{d,A}$ decreases faster than $\delta_{d,B}$. This is because when $p = 3/4$, the drivers/riders with type A have a higher probability of being matched with the same type in next periods than the drivers/riders with type B.

B. Theoretical Performance Analysis of the Mechanism

Next, we will study the performance of the heuristic matching mechanism as compared with the optimal matching mechanism presented in Section III. To start with, we define the ratio PR between the expected matching utility obtained by the heuristic mechanism and the expected utility obtained by the optimal mechanism. In the next proposition, we derive a lower bound for the performance ratio PR of our heuristic mechanism in the symmetric case with $p = q$, i.e., demand and supply heading to each destination are balanced since the arriving drivers and riders have equal probabilities to be of the same destination types.

Proposition 1. *Under balanced demand and supply, i.e., $p = q$, the performance ratio PR of our heuristic mechanism compared to the optimal mechanism is always above*

$$PR \geq \frac{(2T+1)p^2}{1+2Tp} + \frac{(2T+1)(1-p)^2}{1+2T(1-p)}, \tag{1}$$

where the RHS achieves its minimum value $\frac{2T+1}{2T+2}$ when p=0.5.

This bound is given by the average number of matched same-type pairs (each with a utility of 1) per-period under the fixed thresholds $\delta_{d,A} = \delta_{r,A} = \delta_{d,B} = \delta_{r,B} = 0$, and thus independent of the extra driving cost c. The minimum value $\frac{2T+1}{2T+2}$ increases as the length $T \geq 0$ of the waiting time window increases, and attains the minimum value 0.5 when $T = 0$. But when $T = 0$, the heuristic mechanism is the same as the optimal mechanism since in each period both of them will match the new arriving driver and rider who stay for only the current period with each other. Therefore, we can conclude that the efficiency of our heuristic mechanism is at least $\frac{2\times1+1}{2\times1+2}$=75% for any T.

Fig. 5 shows the performance ratio PR of our heuristic mechanism as a function of the probability p when supply and demand heading to each destination are balanced $p = q$, for different values of T. Our mechanism performs better when the two destinations are not symmetric (i.e., when $p, q \neq 0.5$) and when there is more flexibility in defining matching pairs (i.e., T is larger). We observe for all values of T that always the lower bound first decreases and then increases as p increases, and achieves its minimum value when $p = 0.5$.

Next, we extend our analysis to the general case with arbitrary p and q. In the next proposition we analytically obtain the closed-form upper bound for PR based on the idea that

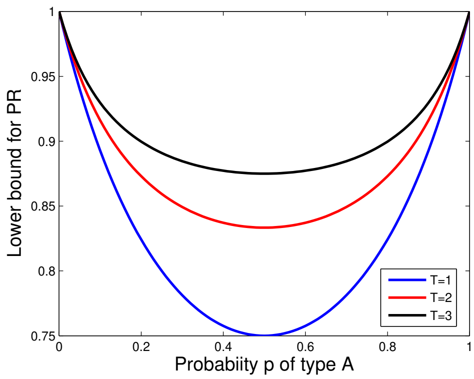

Fig. 5: The lower bound of the performance ratio PR of our heuristic mechanism as a function of the probability p when supply and demand heading to each destination are balanced (i.e., $p{=}q$), for different T. Our mechanism performs better when two destinations are not symmetric (i.e., when $p, q{\neq}0.5$) and when there is more flexibility in defining matching pairs.

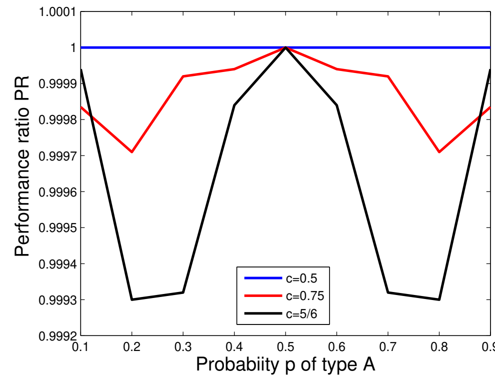

Fig. 6: The exact value of the performance ratio PR of our heuristic mechanism as a function of the probability p when supply and demand heading to each destination are balanced (i.e., $p = q$) and $T = 2$, for different values of c. By comparing this result with the previously displayed lower bound for PR shown in Fig. fig:pr, the bound appears very conservative when the two destinations are symmetric (i.e., $p = q = 0.5$).

our heuristic mechanism must outperform a mechanism with fixed thresholds $\delta_{d,A} = \delta_{r,A} = \delta_{d,B} = \delta_{r,B} = T$, ensuring that a rider/driver never departs unmatched.

Proposition 2. *For any probabilities p and q of destination type, our heuristic mechanism's performance ratio PR as compared to the optimal mechanism is lower bounded by*

$$PR \geq \frac{1-(p+q-2pq)c}{1-|p-q|c} \geq 1-\frac{c}{2}, \tag{2}$$

where the second inequality becomes equality when $p{=}q{=}0.5$.

Recall that when the extra driving cost c is small or large, the heuristic mechanism is always the same as the optimal mechanism. Therefore, we only need to worry about the case of moderate c and the obtained lower bound $1-c/2$ would not be bad in such a case.

Next, we will show the exact value of the performance ratio PR of our heuristic matching mechanism in the case of small waiting time window T. Notice that, due to the exponential computational complexity of the optimal mechanism, we can obtain the exact value of the optimal expected matching utility only for small $T \leq 2$. For such small window lengths, from our simulation results, we observe that the performance ratio of our heuristic mechanism is nearly 100%. Actually, only when $T{=}2$, $p{=}q{\neq}0.5$ and c is large, there exists a performance loss between our heuristic and the optimal mechanisms. In Fig. 6, we depict the exact value of our heuristic mechanism's performance ratio PR versus the probability p of type A for different c when $T = 2$. We observe that the performance ratio achieves its maximum value of 100% when $p{=}q{=}0.5$ for different c, and when c is not large (e.g., $c = 0.5$) the performance loss is zero for all p. This is overall consistent with our intuition that when T is small, the future arrivals play a smaller role and simple heuristic decisions work very well.

Further, we extend our experiments for the more general asymmetric case of $p \neq q$. In this case, we also study the performance ratio PR of our heuristic mechanism compared to the optimum for different value combinations of p and q. We observe that the $PR = 100\%$ is still always achieved by our heuristic mechanism when $T = 2$ and $c = 0.5$.

V. Incentive Design for the Heuristic Mechanism

We consider the practical case where agents act selfishly. This can result in a) non-cooperation, and b) misreporting. Non-cooperation is when an agent refuses a proposed matching by the platform because he might get a better one in the future. In any online matching mechanism, there is a challenge that selfish agents capable of staying more periods may be better off declining a suboptimal matching (unless adequately compensated) [10]. Misreporting arises when a strategic driver might prefer to misreport his private destination type to obtain some potential compensation by the platform for serving a different-type rider, although in reality they actually share the same destination. Both non-cooperation and misreporting can lead to market inefficiency.

It is true that the non-cooperation problem disappears if the ride-sharing platform can force the participants to accept any partner assigned by the matching mechanism as the precondition for not paying a large fine or being blocked from further participation. In this section, we consider two possible 'blocking' strategies by the platform, i.e., rules that must be followed by the participants in order to be accepted in the market. The first is very strict, and the second seems not to affect agents that report truthfully.

Definition 1. *Blocking Strategies:*

1) *Full blocking strategy: participants are not allowed to decline any matching partner assigned by the platform.*
2) *Partial blocking strategy: participants are not allowed to decline any assigned matching partner of the same type as their declaration.*

Assuming the platform uses any of the above platform policies, the participating agents can be strategic about their type declaration. The incentive problem of the platform is to induce truthful reporting. To achieve that, the mechanism must be extended to include payments between the agents. In this section, we will propose some simple pricing schemes and study their incentive compatibility properties in scenarios with the full blocking strategy, where non-cooperation is not an issue. Then, we extend our analysis to scenarios with only the partial blocking strategy and study the Price-of-Anarchy.

A. Incentive Compatibility under Full Blocking

We assume that the platform uses a full blocking strategy that forbids any non-cooperation behavior, and focus on incentive compatibility regarding type declarations. First notice that riders have no incentive to misreport their true type since, if they are dropped at a different destination, it is too costly for them to walk to their actual destination. Thus, we focus on the incentive design to induce drivers' truthful reporting of their destination types. Our incentive mechanism is very simple and involves direct payments from riders (who get the service) to drivers (who offer the service). The value of the payment by a rider is referred to as the 'price' of the service. We consider the case of prices r_0 for each same-type matching and r_1 for each different-type matching (paid by riders to drivers). This is the simplest way to price, by not differentiating among the residual waiting times of the matched drivers and riders.

A fair way to define such prices is $r_0 = \frac{1}{2}$ and $r_1 = \frac{1+c}{2}$, which induce equal splitting of the total utility (i.e., the value of rider minus the cost of driver) obtained by the rider-driver coalition. We refer to this as 'mid-pricing': using the mean of the matched driver's cost and rider's value. If the driver incurs no extra cost, the rider offers him half his obtained service value. If the driver incurs extra cost, the rider splits with the driver both his service value and the driver's cost. Note that although $r_1 > r_0$, a driver prefers a same-type matching because his net benefit (net utility) is higher. From different-type matching, he gets an extra $c/2$ but he pays c from his pocket, obtaining a net benefit of $0.5 - c/2$ that gets smaller for larger c. If $c = 1$ he has no incentive to participate in a different-type matching.

Lemma 4. *Mid-pricing is not incentive compatible regarding type declarations.*

This standard mid-pricing scheme is not incentive compatible for drivers. For example, when most of the riders are of type A (i.e., $q \to 1$), drivers with type A would benefit from declaring type B since the system will have to very frequently match them with type A riders and obtain an additional $c/2$, independently of what they actually declared as their type.

Next, in a market of balanced supply and demand $p = q$, we prove that drivers are truthful under mid-pricing when the probability p of type A does not take extreme values (e.g., not close to 0 or 1) or the extra driving cost c is large. We can rigorously prove this for the simple case of $T = 1$, but we expect similar conditions to hold for larger values of T.

Proposition 3. *If demand and supply for each destination location are balanced (i.e., $p = q$) and $T = 1$, drivers are truthful under the mid-pricing scheme if i) $p \in (0, 0.17)$ and $c \geq \frac{-p^4+p^3+p+1}{-5p^4+6p^3-p+2}$, ii) $p \in (0.17, 0.83)$ or iii) $p \in (0.83, 1)$ and $c \geq \frac{p^4-3p^3+3p^2-2}{5p^4-14p^3+12p^2-3p-2}$.*

Intuitively, this is because the algorithm tends to produce more same-type matchings than different-type matchings as much as possible. When p is far from 1 or 0, this implies a higher risk for untruthful drivers being matched with the 'same-type' partner (which is actually the different-type given untruthful reporting), which reduces any net benefit from not being truthful, even for low values of c. But, if the system is highly unbalanced, e.g., $p, q \to 1$, most matchings will be predictably of the popular destination/type A, and drivers of that type can safely misreport and get higher payments without extra driving costs. Moreover, when the extra driving cost c is large, the drivers are willing to take less risk of being matched with a rider of different type since their net benefit from such a matching is $0.5 - c/2$ instead of 0.5.

To design an incentive compatible pricing that works for all parameter p, q settings, we need to give up fairness in value sharing and consider an indiscriminate pricing $r_0 = r_1$.

Lemma 5. *The indiscriminate pricing $r_0 = r_1 = c$ is incentive compatible regarding type declarations.*

However, we should note that although drivers are truthful under the simplest pricing $r_0 = r_1 = c$, they are left with exactly zero net benefit when being matched with a different-type partner. Thus, if there exists an error in the estimation of the extra driving cost c, the performance of the ride-sharing system may be greatly reduced due to the potential negative net utility from matching to drivers.

B. Inefficiency of Non-cooperation under Partial Blocking

Recall that the partial blocking strategy allows participants to reject different-type matchings (that they claim to be undesired), but not same-type matchings. This prevents participants from jointly misreporting their types and rejecting their assigned partners.

Lemma 6. *Under the partial blocking mechanism, a selfish driver or rider will never choose to both untruthfully report its destination type and reject the assigned matching.*

Our goal is to analyze the efficiency of the (stationary) equilibrium in such a system. Agents choose to be truthful or not when they join the system, and then, depending on their residual times, to be cooperative or not. We make the assumption that agents do not randomize, and agents of the same full type (driver or rider, destination) choose the same policy, e.g., all drivers with destination A reject the different-type matching unless they are about to leave immediately.

To make this equilibrium analysis more tractable, we consider the important symmetric case of $p = q = 0.5$ and $T = 1$, which introduces in the most simple way time to the decisions of the agents. It gives agents the choice to refuse to be matched during their first period in the system, while they will always accept any matching in the second period since they are imminently departing. Notice that in the symmetric case, all drivers and riders that have the same residual time are of the same type and hence choose the same policy in the equilibrium. By utilizing this, we can first prove that the mid-pricing and indiscriminate pricing schemes are still incentive compatible regarding type declarations under partial blocking. Then, we study the inefficiency caused by non-cooperation in the equilibrium. This is the only interesting case, since Lemma 6 excludes the case of being both non-truthful and non-cooperative, and the case of being cooperative but not truthful has been analyzed in the previous subsection, where we assumed full blocking to deny non-cooperation.

To examine the inefficiency caused by non-cooperation, we define the Price-of-Anarchy (PoA) as the ratio of the optimal social welfare (i.e., maximum matching utility achieved when the platform has full control over the decisions of the agents) and the rate of value generated by the worst market equilibrium. We first analyze the PoA for the mid-pricing scheme (i.e., $r_0 = \frac{1}{2}, r_1 = \frac{1+c}{2}$). It is easy to see that the worst equilibrium under the mid-pricing corresponds always to the case when all drivers and riders do not accept the different-type matching unless they are about to leave immediately.

Proposition 4. *Given $p = q = 0.5$, $T = 1$ and using the mid-pricing scheme, the PoA is given by*

$$PoA = \begin{cases} \frac{60-18c}{50-5c}, & \textit{if } 0 \leq c < \frac{10}{13}, \\ 1, & \textit{if } \frac{10}{13} \leq c < 1\,, \end{cases} \tag{3}$$

which achieves its maximum value 1.2 when $c = 0$.

The proof is lengthy and given in our online technical report [12]. We observe for small to intermediate values of c, the socially optimal decision of being potentially matched early with a different-type agent is different than the action in the worst equilibrium, where agents always refuse low-value matchings if they can still wait. If c takes a high enough value, it becomes socially optimal not to match different-type agents unless they are departing, and the PoA is 1. Hence, selfish behavior can create more inefficiency when c gets smaller.

Next, we analyze the PoA given using the indiscriminate pricing scheme (i.e., $r_0 = r_1 = c$). Note that, in this case, drivers don't gain anything by accepting a different-type partner since the different-type matching induces zero utility for drivers, the same as departing unmatched. Meanwhile, riders have no difference between being matched with a same-type partner or being matched with a different-type partner, since the same utility $1-c$ is derived for both cases. Therefore, in the equilibrium, drivers will choose to accept a different-type partner only in the last round while riders will always choose to accept, for all values of c.

Proposition 5. *Given $p = q = 0.5$, $T = 1$ and using the indiscriminate pricing scheme, the PoA is given by*

$$PoA = \begin{cases} \frac{70-21c}{65-15c}, & \textit{if } 0 < c < \frac{10}{13}, \\ 1, & \textit{if } \frac{10}{13} \leq c < 1. \end{cases} \tag{4}$$

which achieves its maximum value 1.0769 when $c \to 0$.

The proof is given in our online technical report [12]. We observe that the indiscriminate pricing achieves a better PoA than the mid-pricing, while the mid-price is more robust to error in the estimation of c. This is because, for the indiscriminate pricing to work as expected, we need the exact value for c. This is not the case for mid-pricing.

VI. Conclusions

In this paper, we address the practical complexities of matching riders and drivers within specified deadlines as they dynamically enter the ride-sharing system. We begin by modeling the dynamic matching process as an MDP, successfully reducing the action space to a limited number. Despite this, the exponential number of states maintains formidable complexities in achieving the optimal matching. Then, we propose a simple threshold-based mechanism designed to match different-type agents only based on their residual time. Theoretical proofs establish the mechanism's near-optimal performance, especially when the demand and supply are balanced. Furthermore, we investigate the incentive properties of the mechanism considering potential non-cooperative and non-truth-telling behaviors by agents. Our analysis demonstrates that the two proposed simple pricing schemes effectively incentivize truthful type reporting, and exhibit a commendably limited PoA in the balanced case.

References

[1] GrabHitch. https://help.grab.com/driver/en-sg/900006356563

[2] Q. Lin, W. Xu, M. Chen, and X. Lin, "A Probabilistic Approach for Demand-Aware Ride-Sharing Optimization," In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), 2019, pp. 141–150.

[3] Q. Lin, L. Deng, J. Sun and M. Chen, "Optimal Demand-Aware Ride-Sharing Routing," in IEEE Conference on Computer Communications (INFOCOM), 2018, pp. 2699-2707.

[4] X. Bei, and S. Zhang, "Algorithms for trip-vehicle assignment in ride-sharing," In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[5] C. Wang, Y. Song, Y. Wei, G. Fan, H. Jin and F. Zhang, "Towards Minimum Fleet for Ridesharing-Aware Mobility-on-Demand Systems," in IEEE Conference on Computer Communications (INFOCOM), 2021, pp. 1-10.

[6] H. Yu, E. Wei, and R. A. Berry, "Learning to price vehicle service with un-known demand," In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), 2020, pp. 161–170.

[7] C. Courcoubetis and A. Dimakis, "Throughput and Pricing of Ridesharing Systems," in IEEE Conference on Computer Communications (INFOCOM), 2019, pp. 640-648.

[8] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," Proceedings of the National Academy of Sciences, 2017, 114(3), pp.462-467.

[9] N. Agatz, E. Alan, S. Martin, and W. Xing, "Optimization for dynamic ride-sharing: A review," European Journal of Operational Research, 2012, 223(2), pp.295-303.

[10] M. Baccara, S.M. Lee, and L. Yariv, "Optimal dynamic matching," Theoretical Economics, 2020, 15(3), pp.1221-1278.

[11] M. Akbarpour, S. Li, and S.O. Gharan, "Dynamic matching market design," 2014, [Online]. Available: https://arxiv.org/abs/cs/1402.3643.

[12] S. Gao, C. Courcoubetis, L. Duan, "Dynamic matching for ride-sharing with deadlines," Online technical report: https://www.dropbox.com/scl/fi/ndjojo4w2bm9vuhsn2t3b/Ride-sharing.pdf?rlkey=06xruuocsy258dopiebrb1ii6&dl=0