# Optimal Demand-Aware Ride-Sharing Routing

Qiulin Lin, Lei Deng, Jingzhou Sun, Minghua Chen

***Abstract*—We consider the problem of exploring travel demand statistics to optimize ride-sharing routing, in which the driver of a vehicle determines a route to transport multiple customers with similar itineraries and schedules in a cost-effective and timely manner. This problem is important for unleashing economical and societal benefits of ride-sharing. Meanwhile, it is challenging due to the need of (i) meeting travel delay requirements of customers, and (ii) making online decisions without knowing the exact travel demands beforehand. We present a general framework for exploring the new design space enabled by the demand-aware approach. We show that the demand-aware ride-sharing routing is fundamentally a two-stage stochastic optimization problem. We show that the problem is NP-Complete in the weak sense. We exploit the two-stage structure to design an optimal solution with pseudo-polynomial time complexity, which makes it amenable for practical implementation. We carry out extensive simulations based on real-world travel demand traces of Manhattan. The results show that using our demand-aware solution instead of the conventional greedy-routing scheme increases the driver's revenue by 10%. The results further show that as compared to the case without ride-sharing, our ride-sharing solution reduces the customers' payment by 9% and the total vehicle travel time (indicator of greenhouse gas emission) by 17%. The driver can also get 26% extra revenues per slot by participating in ride-sharing.**

## I. Introduction

Thanks to the fast growth of mobile Internet technologies, ride-sharing systems such as UberPool [1] and LyftLine [2] become popular in today's urban mobility. Ride-sharing allows multiple customers with similar itineraries and time schedules to share a vehicle. It can significantly increase vehicles' occupancy rate, alleviate traffic congestion, and reduce the energy consumption of urban commuting. Take Manhattan as an example. The study in [3] shows that allowing two customers to share a taxi can reduce taxis' cumulative trip length by 40%, and the authors in [4] show that 3000 vehicles each shared by at most four customers can meet 98% of the demands that are currently served by 13000+ taxis.

Ideally, ride-sharing represents a win-win-win transportation paradigm: customers pay less if they are willing to tolerate minor extra delay, drivers achieve higher income rates by transporting multiple customers in a single trip, and the service provider like Uber boosts its service capacity. Two key modules in the ride-sharing system are customer-vehicle matching and ride-sharing routing.

The customer-vehicle matching module concerns with two issues: (i) how to group customers to create proper ride-sharing opportunities, and (ii) how to match the formed groups to vehicles. Recent studies have been focusing on designing and optimizing the matching module; see e.g., [3]–[13]. Meanwhile, ride-sharing routing is to determine a path for the driver to pick up and deliver customer(s) in the formed group. Different choices of paths can lead to different cost or profit to the driver and/or service providers. If the customer travel demands are given, the driver can simply follow the path with minimum cost or maximum profit. In [3], [4], a shortest-path-like routing algorithm is applied for each formed group. In the design in [3], [4], the matching and routing are coupled with each other when doing ride-sharing decisions. In particular, the weights used in matching are closely related to the routing choices [3], [4].

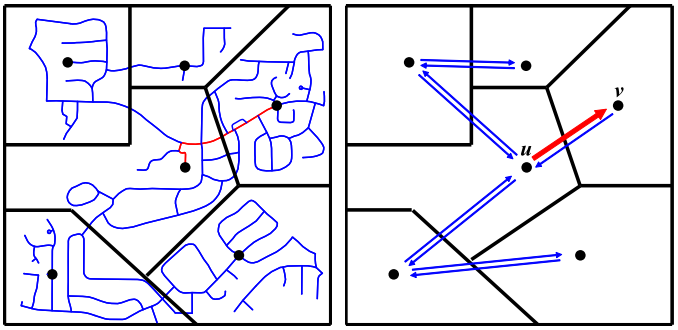
One main difficulty of performing the matching and routing is that future demands are unknown when making decisions. Existing works outline two approaches to addressing this difficulty. The first one is to assume that all future travel demands are known at each matching decision epoch, i.e., in an offline fashion; see e.g., [3]. Such assumption may not hold in practice. The second one is to assume zero knowledge of future demands, i.e., in a complete demand-oblivious fashion; see e.g., [4], [6], [10], [11].

In our study, we observe that even though we cannot know the future demands exactly, customer travel demands usually show weekly and daily pattern [14] and thus their statistics can be learned to improve the matching and routing decisions. In particular, we advocate a demand-aware approach of leveraging travel demand statistics to improve ride-sharing routing performance. This introduces a new design space into the important ride-sharing routing problem. We also discuss how to use our demand-aware routing algorithm as a building block to improve matching decisions with demand statistics in Appendix IX-A. We make the following contributions.

▷ In Sec. II, to explore the new design space in our demand-aware approach, we propose a probabilistic model for customer travel demands. The model is general and can be used to incorporate demand statistics into ride-sharing routing optimization.

▷ In Sec. III, we formulate the demand-aware ride-sharing routing problem of revenue maximization subject to customers' travel delay constraints. It is fundamentally a two-stage stochastic optimization problem. We prove that the problem is NP-Complete, which implies that it is impossible to obtain the exact solution in polynomial time unless P = NP.

▷ In Sec. IV, we propose a pseudo-polynomial-time algorithm to solve the demand-aware ride-sharing routing problem *optimally* with time complexity $O(\Delta_{s,d}^{\max}|\mathcal{V}|^2 + |\mathcal{V}|^3)$, where $|\mathcal{V}|$ is the number of nodes of the transportation network and $\Delta_{s,d}^{\max}$ is the travel delay requirement of the original customer (defined later as Rider-I). The complexity is pseudo-polynomial in the sense that it is polynomial in the value of the problem input $\Delta_{s,d}^{\max}$, but is exponential in the bit length of the problem input, i.e., $\log(\Delta_{s,d}^{\max})$ [15]. We remark that the complexity is polynomial in the network size and thus the solution is actually amendable for practical implementation. As a byproduct, our results also show that the problem is NP-




(a) Road graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$. (b) Region-based (constructed) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Fig. 1. An example of the road graph and the constructed region-based graph. We divide the road graph into 6 regions as shown in Fig. 1(a). Each region has a representative node marked as a black dot. The constructed region-based graph in shown in Fig. 1(b). Each node in the region-based graph represents a region in the road graph. Each edge $(u, v)$ in the region-based graph represents a fastest path in the road graph from the representative node of region $u$ to that of region $v$. For example, the red edge in the region-based graph in Fig. 1(b) represents the red path in the road graph in Fig. 1(a).

Complete in the weak sense [16].

▷ In Sec. VI, we carry out extensive simulations based on real-world travel demand traces in Manhattan. We show that as compared to a conventional greedy-routing scheme without using demand statistics, our proposed demand-aware ride-sharing routing solution improves the driver's revenue by 10%. We further use our proposed solution as a building block to show that as compared to the case without ride-sharing, ride-sharing reduces the customers' payment by 9%, increases the driver's revenue per slot by 26%, and reduces the total vehicle travel time by 17%.

## II. System Model

### A. Network Model

We consider an urban road network modeled as a directed road graph $\mathcal{G}_0 \triangleq (\mathcal{V}_0, \mathcal{E}_0)$ with node set $\mathcal{V}_0$ and edge set $\mathcal{E}_0$, as shown in Fig. 1(a). Time is divided into slots of equal length. Each edge $(u_0, v_0) \in \mathcal{E}_0$ is a road segment from node $u_0 \in \mathcal{V}_0$ to node $v_0 \in \mathcal{V}_0$. We further assign a segment-dependent travel time $\xi_{u_0,v_0}$ (unit: slots) to edge $(u_0, v_0)$. To introduce our customer travel demand model later in this section, we construct a region-based graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$, as shown in Fig. 1(b). More specifically, we partition the road graph $\mathcal{G}_0$ into multiple regions. For each region, we assign a representative node from which all other nodes in this region can be reached in a given number of slots, e.g., 5 slots, represented by the black dots in Fig. 1(a)[1]. Then each node $v \in \mathcal{V}$ in the region-based graph $\mathcal{G}$ represents a region in the road graph $\mathcal{G}_0$. We add a directed edge $(u, v)$ from node $u$ to node $v$ in $\mathcal{G}$, as illustrated in 1(b), if there exists one or multiple paths from the representative node of the former corresponding region to that of the latter one in the road graph such that the major fraction of the path is within those two regions[2]. Let $\delta_{u,v}$ be the travel time of the fastest path. All our later modeling and analysis are based on the region-based graph $\mathcal{G}$ unless otherwise specified.

[1]For a general urban road network, constructing the regions is equivalent to solving a clustering problem to find a set of clusters. Location points within each cluster are close to each other. The problem can be solved by using celebrated algorithms like k-means [17]. For a dense and regular urban road network like Manhattan, one can simply partition the district into regions of equal area. We adopt this method in the simulation in Sec. VI.

TABLE I
Key Notations.

(Note that all nodes and edges below are on the region-based graph $\mathcal{G}$)

| Notation | Meaning |
|---|---|
| $\delta_{u,v}$ | Travel time of edge $(u, v)$ (unit: slots) |
| $\beta$ | The discounting rate (unit: dollars per slot) |
| $\Delta^{\mathsf{F}}_{i,j}$ | The minimal travel time from node $i$ to node $j$ (unit: slots) |
| $\Delta^{\mathsf{max}}_{i,j}$ | The maximum travel delay of the customer from node $i$ to node $j$ (unit: slots) |
| $U^{\mathsf{NRS}}_{i,j}$ | The non-ride-sharing payment of the customer from node $i$ to node $j$ (unit: dollars) |
| $U^{\mathsf{RS}}_{i,j}(\tau)$ | The ride-sharing payment of the customer from node $i$ to node $j$ if the customer's travel time is $\tau$ slots (unit: dollars) |
| $P_i(t)$ | The probability that there is a customer request at node $i$ at slot $t$ |
| $P_{i,j}(t)$ | The probability that the customer is from node $i$ to node $j$, conditioning on that there is a customer request at node $i$ at slot $t$ |
| $\hat{P}_i(t)$ | The probability that there is a *feasible* customer request at node $i$ at slot $t$ |
| $w_{i,j}(t)$ | The total revenue of the driver if picking up a new customer from node $i$ to $j$ at slot $t$ (unit: dollars) |
| $w_i(t)$ | The expected total revenue of the driver, conditioning on that the driver picks up a new customer at node $i$ at slot $t$ (unit: dollars) |

### B. Ride-sharing Scenario

Suppose that a driver has already picked up a customer (*Rider-I*) at slot 0 at the source node $s$ and needs to transport Rider-I to the destination node $d$, as exemplified in Fig. 2. Consider the ride-sharing scenario where the driver may pick up another customer (*Rider-II*) along its way[3]. Conventionally, without considering the likelihood of encountering Rider-II, the driver follows a fastest/shortest path from $s$ to $d$ to deliver Rider-I. Consequently, there may be little chance of picking up Rider-II to share the ride by following such a demand-oblivious route, resulting in sub-optimal ride-sharing revenue for the driver and failure to maximize the social benefit of ride-sharing. In this paper, we advocate demand-aware ride sharing routing. That is to take likelihood of encountering Rider-II into account when planning the route between $s$ and $d$ of Rider-I. Being demand-aware can improve the ride-sharing performance in terms of increasing a driver's revenue and reducing the total travel distance, as showed in our simulation.

[2]More specifically, for any two nodes $u, v \in \mathcal{V}$, we denote $T(u, v)$ as the minimal travel time from the representative node of region $u$ to that of region $v$ in the road graph. Then there is an edge from $u$ to $v$ in the region-based graph if $T(u, v) \leq \alpha \cdot (T(u, k) + T(k, v))$ for any region $k \in \mathcal{G}$. In our simulation in Sec. VI, we set $\alpha = 0.8$.

[3]We discuss how to extend our solution into the case of picking up two or more customers in Sec. V-C.

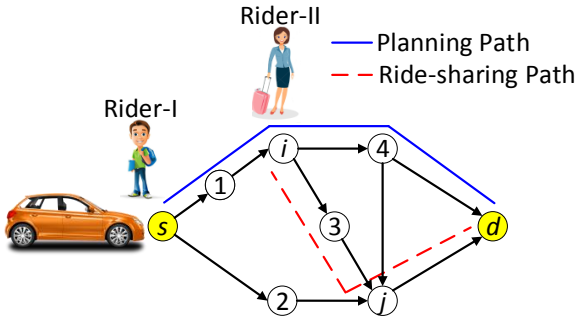



Fig. 2. The system model for ride-sharing routing on the region graph. The driver has already picked up a customer (Rider-I) at the source node $s$ and should deliver the customer to the destination node $d$. The driver then follows a planning path ($s \to 1 \to i \to 4 \to d$) to deliver Rider-I. Suppose that at node $i$, the driver meets the a customer (Rider-II) whose destination is node $j$. Then the driver picks up Rider-II and the service provider re-optimizes the routing choice. The output path is the ride-sharing path. Here at node $i$, the driver changes the path from the planning path ($s \to 1 \to i \to 4 \to d$) to the ride-sharing path ($i \to 3 \to j \to d$). Following the ride-sharing path, the driver delivers both Rider-I and Rider-II before their deadlines.

### C. Travel Demand Model

A travel demand is consisting of a customer's source, destination and pick-up time. We assume that travel demands follow time-dependent statistics patterns. Specifically, at slot $t$ and at node $i$, there is a travel demand with probability $P_i(t)$[4] and the customer goes to node $j$ with probability $P_{ij}(t)$. Note that $P_{ij}(t)$ is a conditional probability and we have $\sum_{j\in\mathcal{V}} P_{ij}(t) = 1$. In the rest of this paper, we assume that $P_s(0) = 0$ (i.e., no other customer is at node $s$ at slot 0). Otherwise, the driver can form a ride-sharing trip of two customers at slot 0 and there is no need to perform demand-aware routing. We further assume that customer arrivals are independent at different nodes and different time slots.

For any customer travel from node $i$ to node $j$, we associate a maximum travel delay $\Delta_{i,j}^{\max}$. That is, if the driver picks up this customer, she must transport the customer in $\Delta_{i,j}^{\max}$ slots. Note that UberPool has already provided such "Arrive By" service [18].

### D. Payment Model

For any customer from node $i$ to node $j$, we define the Non-Ride-Sharing payment as $U_{i,j}^{\mathsf{NRS}}$, which is the amount of money (unit: dollars) that the customer needs to pay to the service provider such as Uber and Lyft (or equivalently to the driver)[5] when he is served by the driver without participating in ride-sharing. Note that in practice, the service provider like Uber and Lyft will charge the customers according to the total travel time and total travel distance, which further depend on the current traffic condition. In this paper, we assume stable traffic condition and as such the edge travel time $\delta_{u,v}$ is constant. Without ride-sharing, the driver transports a customer from node $i$ to node $j$ along the fastest path and the travel delay is also constant. Thus, $U_{i,j}^{\mathsf{NRS}}$ is time-invariant.

We define the Ride-Sharing payment of a customer from node $i$ to node $j$ as

$$U_{i,j}^{\mathsf{RS}}(\tau) \triangleq U_{i,j}^{\mathsf{NRS}} - \beta \cdot (\tau - \Delta_{i,j}^{\mathsf{F}}), \tag{1}$$

where $\tau$ is the travel time of the customer, $\beta$ is the discounting rate (unit: dollars per slot), and $\Delta_{i,j}^{\mathsf{F}}$ is the minimal travel time from node $i$ to node $j$. Eq. (1) shows that the payment reduction of the customer is proportional to the extra travel time when participating in ride-sharing. That is, the rider gets more payment reduction if she is detoured more. We further show that it is a *fair* mechanism to the customers in the sense that the ratio of the cost (including payment and travel delay) to the amount of service (indicated by the minimum travel time) are the same among different customers (see more details Appendix V-A. ). In contrast, the current scheme used by UberPool or LyftLine is not a fair mechanism. As it charges a customer according to his total travel time and distance [20], Rider-I may not only suffer from longer travel delay, but also have to pay more.

In addition, to avoid the ride-sharing payment $U_{i,j}^{\mathsf{RS}}(\tau)$ from being negative, we assume that the maximum travel delay $\Delta_{i,j}^{\max}$ and the discounting rate $\beta$, set by the service provider, satisfy

$$0 \le \beta \le \frac{U_{i,j}^{\mathsf{NRS}}}{\Delta_{i,j}^{\max} - \Delta_{i,j}^{\mathsf{F}}}, \forall i, j \in \mathcal{V}. \tag{2}$$

We summarize key notations in Tab. I.

## III. Problem Formulation

In this work, we consider a Demand-Aware Ride-sharing rouTing problem, denoted as DART. Our objective is to maximize the expected total revenue of the service provider (or equivalently the revenue of the driver)[6]. Our constraint is to deliver Rider-I and Rider-II (if any) before their deadlines. Here we assume that when the driver picks up Rider-I, there is no available Rider-II such that they can form a ride-sharing trip right away. Thus, the service provider should plan a path against the uncertainty of future travel demands. If the driver indeed picks up Rider-II, the service provider optimizes the route again to deliver both riders and maximize its revenue. Such a practical consideration suggests two design spaces in our problem:

- A planning path: at slot 0, the service provider plans a path for the driver to transport Rider-I from $s$ to $d$, with future travel demand statistics taken into account.
- A ride-sharing path: if the driver picks up Rider-II at node $i$ traveling to node $j$ at slot $t$, the service provider determines a ride-sharing path such that the driver can deliver both Rider-I and Rider-II before their deadlines and maximize his total revenue.

[4]Note that we incorporate region-based graph $\mathcal{G}$ to consider the travel demand patterns since for each node on the road graph $\mathcal{G}_0$, it's too small to conclude its statistic patterns.

[5]Note that the driver normally can get certain portion of the payment from the served customer. For example, Uber drivers can get 80% of trip payment [19]. In this paper, ignoring the constant revenue-splitting coefficient and without loss of generality, we interchangeably use *service provider's revenue* and *driver's revenue*.

[6]For a service provider, its cost is slightly related to a single trip. Thus, we omit the cost here and aim to maximize its revenue. For a driver, we discuss how to take her cost into account and maximize her profit in Sec. V-C.

Note that both paths are on the region graph, where each node represents a region. In practice, the driver may need to pick up the customer in any location within the region. It may incur minor extra delay, an upperbound of which can be included when setting the travel delay requirement.

In this paper, we assume that a driver will pick up the first *feasible* customer appeared along the planning path, so as to minimize the waiting time of the customer. A customer at node-$i$, whose destination node is $j$, is *feasible* if there exists a path starting at node $i$ and passing through both node $d$ and node $j$ such that both customers can be delivered in time.

Overall, in our problem, the service provider needs to determine a planning path for the driver first. Then depending on when and where the driver can meet the first feasible customer, the service provider further determines a ride-sharing path to deliver both customers. This naturally suggests a two-stage structure for problem formulation [21]. That is, the service provider designs a planning path to deliver Rider-I at Stage-1 by considering future demand statistics and the expected revenue abstained from Stage-2, where a ride-sharing path is determined to deliver both riders.

### A. Stage 2

If the driver picks up Rider-II at node $i$ at slot $t$, whose destination is node $j$, then the Stage-2 problem is to select a path from node $i$ to both node $d$ and node $j$ so as to maximize the service provider's total revenue (or equivalently the driver's total revenue). We denote the problem as Stage-2$(i,j,t)$, which can be formulated as

$$\max_{\mathcal{P}^{\mathsf{RS}}\in\mathbb{P}_{i,j,d}} \quad U^{\mathsf{RS}}_{s,d}(t+\tau_{i,d}(\mathcal{P}^{\mathsf{RS}}))+U^{\mathsf{RS}}_{i,j}(\tau_{i,j}(\mathcal{P}^{\mathsf{RS}})) \tag{3a}$$

$$\text{s.t.} \quad \tau_{i,j}(\mathcal{P}^{\mathsf{RS}})\le \Delta^{\mathsf{max}}_{i,j}, \tag{3b}$$

$$t+\tau_{i,d}(\mathcal{P}^{\mathsf{RS}})\le \Delta^{\mathsf{max}}_{s,d}, \tag{3c}$$

where $\mathbb{P}_{i,j,d}$ is the set of the paths that start at node $i$ and pass through *both* node $d$ and node $j$[7], and $\tau_{i,j}(\mathcal{P}^{\mathsf{RS}})$ (resp. $\tau_{i,d}(\mathcal{P}^{\mathsf{RS}})$) is the total travel time from node $i$ to node $j$ (resp. node $d$) along the path $\mathcal{P}^{\mathsf{RS}}\in\mathbb{P}_{i,j,d}$.

Now we have the following two cases:

- If there exists a path $\mathcal{P}^{\mathsf{RS}}\in\mathbb{P}_{i,j,d}$ such that Stage-2$(i,j,t)$ is feasible (i.e., both customers can be delivered in time), then the customer from node $i$ to node $j$ at slot $t$ is feasible and the driver will pick up this customer as Rider-II.
- Otherwise, this customer is not feasible and the driver continues along the planning path.

We denote the optimal value of problem Stage-2$(i,j,t)$ as $w_{i,j}(t)$. It represents the total revenue of the driver upon picking up Rider-II from node $i$ to node $j$ at slot $t$.

Before we introduce our Stage-1 formulation, we denote $\hat{P}_i(t)$ as the probability that the driver can pick up a feasible customer at node $i$ at slot $t$, and it is given by

$$\hat{P}_i(t)\triangleq P_i(t)\cdot\sum_{j\in\mathcal{V}:j \text{ is feasible at slot } t} P_{i,j}(t), \tag{4}$$

[7]The path in $\mathbb{P}_{i,j,d}$ could first reach either node $j$ or node $d$.

where $P_i(t)$ is the probability that there is a customer request at node $i$ at slot $t$ and $\sum_{j\in\mathcal{V}:j \text{ is feasible at slot } t} P_{i,j}(t)$ is the probability that the customer is feasible.

Further, conditioning on that the driver picks up a *feasible* customer at node $i$ at time $t$, the expected total revenue is defined as

$$w_i(t)\triangleq \sum_{j\in\mathcal{V}:j \text{ is feasible at slot } t}\frac{P_i(t)\cdot P_{i,j}(t)}{\hat{P}_i(t)}\cdot w_{i,j}(t), \tag{5}$$

where $\frac{P_i(t)\cdot P_{i,j}(t)}{\hat{P}_i(t)}$ is the probability that there is a travel demand at node $i$ at slot $t$ traveling to node $j$, conditioning on that the driver picks up a *feasible* customer at node $i$ at slot $t$.

### B. Stage 1

In Stage 1, the service provider needs to determine a planning path for the driver to deliver Rider-I from node $s$ to node $d$ at slot 0. Suppose that the planning path is denoted as $\mathcal{P}^{\mathsf{planning}}=(v_0,v_1,v_2,\cdots,v_n,v_{n+1})$ where $v_0=s$ and $v_{n+1}=d$ and there are $n$ intermediate nodes. Then along this planning path, the driver will reach node $v_i$ ($i\in[1,n+1]$) at slot

$$t_{v_i}\triangleq \delta_{v_0,v_1}+\delta_{v_1,v_2}+\cdots+\delta_{v_{i-1},v_i}=\sum_{k=1}^{i}\delta_{v_{k-1},v_k}. \tag{6}$$

For any $i\in[1,n]$, the probability that the driver will pick up a feasible new customer at node $v_i$ at slot $t_{v_i}$ is

$$(1-\hat{P}_{v_1}(t_{v_1}))\cdot(1-\hat{P}_{v_2}(t_{v_2}))\cdots(1-\hat{P}_{v_{i-1}}(t_{v_{i-1}}))\cdot\hat{P}_{v_i}(t_{v_i}),$$

and the expected total revenue conditioning on that the driver picks up a feasible new customer at node $v_i$ at slot $t_{v_i}$ is $w_{v_i}(t_{v_i})$ as defined in (5). The probability that the driver does not pick up any feasible new customer along the planning path is

$$(1-\hat{P}_{v_1}(t_{v_1}))\cdot(1-\hat{P}_{v_2}(t_{v_2}))\cdot\cdots\cdot(1-\hat{P}_{v_n}(t_{v_n})).$$

The (expected) total revenue conditioning on that the driver does not pick up any feasible new customer is $U^{\mathsf{NRS}}_{s,d}-\beta\cdot(t_{v_{n+1}}-\Delta^{\mathsf{F}}_{s,d})$ as defined in (1). Namely, even though the driver does not pick up any new customer, Rider-I with a solo ride still pays the ride-sharing price [22].

Based on the total expectation theorem, the expected total revenue of the driver with planning path $\mathcal{P}^{\mathsf{planning}}=(v_0,v_1,v_2,\cdots,v_n,v_{n+1})$ is

$$\begin{aligned}&\textsf{Expected-total-revenue}(\mathcal{P}^{\mathsf{planning}})\\&=\sum_{i=1}^{n}\left(\prod_{j=1}^{i-1}(1-\hat{P}_{v_j}(t_{v_j}))\right)\hat{P}_{v_i}(t_{v_i})w_{v_i}(t_{v_i})\\&\quad+\prod_{j=1}^{n}(1-\hat{P}_{v_j}(t_{v_j}))\left[U^{\mathsf{NRS}}_{s,d}-\beta\cdot(t_{v_{n+1}}-\Delta^{\mathsf{F}}_{s,d})\right].\end{aligned} \tag{7}$$

Then our State-1 problem (or the whole DART problem) is formulated as

$$\max_{\mathcal{P}^{\mathsf{planning}}\in\mathbb{P}_{s,d}} \quad \textsf{Expected-total-revenue}(\mathcal{P}^{\mathsf{planning}}) \tag{8a}$$

$$\text{s.t.} \quad t_{v_{n+1}}\le\Delta^{\mathsf{max}}_{s,d}, \tag{8b}$$

where $\mathbb{P}_{s,d}$ is the set of all $s-d$ paths and constraint (8b) restricts that the driver can deliver Rider-I before his deadline along the planning path. Note that both the total number of intermediate nodes $n$ and the node-$v_i$ arriving time $t_{v_i}$ as calculated in (6) depend on the selected planning path $\mathcal{P}^{\mathsf{planning}}$.

We now present the hardness.

***Theorem** 1:* Problem DART is NP-complete.

*Proof:* We show that the well-known restricted shortest path problem [16], [23] is a special case of our problem. Since restricted shortest path problem is NP-complete, our problem DART is thus also NP-complete. The detailed proof is shown in Appendix IX-A. ∎

Theorem 1 shows that it is impossible to solve DART optimally in polynomial time unless P = NP.

## IV. Optimal Demand-Aware Ride-Sharing Routing

In this section, suggested by the two-stage structure of our problem, we propose a dynamic-programming-based algorithm to solve our problem DART *optimally* in pseudo-polynomial time. We first present an efficient shortest-path-like solution to solve the Stage-2 problem in (3) optimally and then present our dynamic-programming-based algorithm to optimally solve our Stage-1 problem in (8), i.e., DART. Note that as a byproduct, our results show that problem DART is actually NP-complete in the weak sense [16].

### A. Solution to the Stage-2 Problem

In problem Stage-2$(i,j,t)$ formulated in (3), we need to find a path $\mathcal{P}^{\mathsf{RS}} \in \mathbb{P}_{i,j,d}$ from node $i$ to both node $d$ (the destination of Rider-I) and node $j$ (the destination of Rider-II). If we cannot find a path $\mathcal{P}^{\mathsf{RS}}$ such that both (3b) and (3c) are satisfied, i.e., both Rider-I and Rider-II can be delivered before their deadlines, then problem Stage-2$(i,j,t)$ is infeasible. Otherwise, problem Stage-2$(i,j,t)$ is feasible.

According to our objective in (3a) and the ride-sharing payment in (1), we have

$$\begin{aligned}
&U^{\mathsf{RS}}_{s,d}(t+\tau_{i,d}(\mathcal{P}^{\mathsf{RS}})) + U^{\mathsf{RS}}_{i,j}(\tau_{i,j}(\mathcal{P}^{\mathsf{RS}}))\\
&= [U^{\mathsf{NRS}}_{s,d} - \beta\cdot(t+\tau_{i,d}(\mathcal{P}^{\mathsf{RS}}) - \Delta^{\mathsf{F}}_{s,d})] + [U^{\mathsf{NRS}}_{i,j} - \beta\cdot(\tau_{i,j}(\mathcal{P}^{\mathsf{RS}}) - \Delta^{\mathsf{F}}_{i,j})]\\
&= \underbrace{[U^{\mathsf{NRS}}_{s,d} + U^{\mathsf{NRS}}_{i,j} + \beta(\Delta^{\mathsf{F}}_{s,d} + \Delta^{\mathsf{F}}_{i,j} - t)]}_{\text{Constant term}} - \beta\underbrace{[\tau_{i,d}(\mathcal{P}^{\mathsf{RS}}) + \tau_{i,j}(\mathcal{P}^{\mathsf{RS}})]}_{\text{Path-dependent term}}.
\end{aligned}$$

Thus, problem Stage-2$(i,j,t)$ is equivalent to finding a ride-sharing path to minimize the summation of the travel time of Rider-I from node $i$ to node $d$ and the travel time of Rider-II from node $i$ to node $j$ while satisfying their delay constraints in (3b) and (3c). Thus, both the objective and the constraints in problem Stage-2$(i,j,t)$ suggest delivering both riders as soon as possible. We then define the following two paths

$$\mathcal{P}_1 \triangleq (\mathcal{P}^{\mathsf{F}}_{i,j}, \mathcal{P}^{\mathsf{F}}_{j,d}), \text{ and } \mathcal{P}_2 \triangleq (\mathcal{P}^{\mathsf{F}}_{i,d}, \mathcal{P}^{\mathsf{F}}_{d,j}). \tag{9}$$

Here in (9), we denote $\mathcal{P}^{\mathsf{F}}_{u,v}$ as the fastest path from node $u$ to node $v$ and denote $(\mathcal{P},\mathcal{P}')$ as a path that follows sub-path $\mathcal{P}$ first and then sub-path $\mathcal{P}'$. Thus, $\mathcal{P}_1$ is the path that reaches node $j$ first by following the fastest path from $i$ to $j$ and then reaches node $d$ by following the fastest path from $j$ to $d$; while $\mathcal{P}_2$ is the path that reaches node $d$ first by following the fastest path from $i$ to $d$ and then reaches node $j$ by following the fastest path from $d$ to $j$. By the following theorem, we only need to consider $\mathcal{P}_1$ and $\mathcal{P}_2$ to check the feasibility and find the optimal solution of problem Stage-2$(i,j,t)$.

**Algorithm 1** Algorithm to Stage-2$(i,j,t)$

1: Get the fastest paths $\mathcal{P}^{\mathsf{F}}_{i,j}, \mathcal{P}^{\mathsf{F}}_{j,d}, \mathcal{P}^{\mathsf{F}}_{i,d}$, and $\mathcal{P}^{\mathsf{F}}_{d,j}$ (using the shortest path algorithm)
2: Construct path $\mathcal{P}_1 = (\mathcal{P}^{\mathsf{F}}_{i,j}, \mathcal{P}^{\mathsf{F}}_{j,d})$ and path $\mathcal{P}_2 = (\mathcal{P}^{\mathsf{F}}_{i,d}, \mathcal{P}^{\mathsf{F}}_{d,j})$
3: **if** $\mathcal{P}_1$ satisfies both (3b) and (3c) **then**
4:  **if** $\mathcal{P}_2$ satisfies both (3b) and (3c) **then**
5:  **if** $\tau_{i,d}(\mathcal{P}_1) + \tau_{i,j}(\mathcal{P}_1) < \tau_{i,d}(\mathcal{P}_2) + \tau_{i,j}(\mathcal{P}_2)$ **then**
6:  return $\mathcal{P}_1$
7:  **else**
8:  return $\mathcal{P}_2$
9:  **end if**
10:  **else**
11:  return $\mathcal{P}_1$
12:  **end if**
13: **else**
14:  **if** $\mathcal{P}_2$ satisfies both (3b) and (3c) **then**
15:  return $\mathcal{P}_2$
16:  **else**
17:  return Infeasible
18:  **end if**
19: **end if**

***Theorem** 2:* Stage-2$(i,j,t)$ is feasible if and only if either $\mathcal{P}_1$ or $\mathcal{P}_2$ satisfies both (3b) and (3c). If Stage-2$(i,j,t)$ is feasible, then either $\mathcal{P}_1$ or $\mathcal{P}_2$ is an optimal solution.

*Proof:* See Appendix IX-B. ∎

We summarize the above understanding into Algorithm 1 to solve Stage-2$(i,j,t)$ optimally. The time complexity mainly comes from line 1 where we need to obtain four fastest paths. By adapting the best fastest-path algorithm based on Dijkstra's algorithm with Fibonacci heap [24], our Algorithm 1 has a complexity of $O(|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$. In practice, we can pre-compute and store the fastest paths of all pairs with complexity $O(|\mathcal{V}|^3)$ using Floyd-Warshall algorithm [24]. Consequently Stage-2$(i,j,t)$ can be solved with a complexity of $O(1)$.

### B. Solution to the Stage-1 Problem

For Stage-1, we propose a dynamic-programming-based algorithm to solve the problem optimally in pseudo-polynomial time. Define $f_i(t)$ as the the maximum expected total revenue that the driver can obtain if she reaches at node $i$ at slot $t$ and no new customer has appeared yet.

Then at node $i$ at slot $t$, the driver possibly faces two situations:

- A feasible customer at node $i$ at slot $t$, which happens with probability $\hat{P}_i(t)$ (see (4)). In this situation, the driver will pick up this feasible customer as Rider-II and her expected total revenue is $w_i(t)$ (see (5)).
- No feasible customer at node $i$ at slot $t$, which happens with probability $(1-\hat{P}_i(t))$. In this situation, the driver

**Algorithm 2** Algorithm to Stage-1/DART

Get the fastest paths of all $(i,j)$ pairs where $i,j \in \mathcal{V}$
Solve Stage-2$(i,j,t)$ by Algorithm 1, $\forall i,j \in \mathcal{V}, t \in [0, \Delta_{s,d}]$
Get $\hat{P}_i(t)$ according to (4), $\forall i \in \mathcal{V}, t \in [0, \Delta_{s,d}^{\max}]$
Get $w_i(t)$ according to (5), $\forall i \in \mathcal{V}, t \in [0, \Delta_{s,d}^{\max}]$
Initialize $f_i(t) = -\infty, \forall i \in \mathcal{V}, t \in [0, \Delta_{s,d}^{\max}]$
Set $f_d(t) = U_{s,d}^{\mathsf{NRS}} - \beta \cdot (t - \Delta_{u,v}^{\mathsf{F}}), \quad \forall t \in [0, \Delta_{s,d}^{\max}]$
**for** $t = \Delta_{s,d}^{\max}, \Delta_{s,d}^{\max} - 1, \Delta_{s,d}^{\max} - 2, \cdots, 0$ **do**
  **for** $i \in \mathcal{V} \setminus \{d\}$ **do**
    **for** $j \in \mathsf{out}(i)$ **do**
      **if** $t + \delta_{i,j} \le \Delta_{s,d}^{\max}$ **then**
        $f_i(t) = \max\{f_i(t), \hat{P}_i(t) w_i(t) + (1 - \hat{P}_i(t))(f_j(t + \delta_{i,j}))\}$
      **end if**
    **end for**
  **end for**
**end for**
return $f_s(0)$ /* the corresponding planning path can also be returned, but we omit the procedures here for simplicity */

cannot pick up any feasible customer at node $i$. She then continues along the planning the path and goes to one of $i$'s outgoing nodes, say node $j$. The driver spends $\delta_{i,j}$ slots to travel across edge $(i,j)$ and reaches node $j$ at slot $(t + \delta_{i,j})$. The driver's expected total revenue is $f_j(t + \delta_{i,j})$, according to our definition of $f_i(t)$.

Based on these two situations, we obtain the following dynamic-programming structure for $f_i(t)$. Let $\mathsf{out}(i)$ be the set of all outgoing nodes of $i$,

$$f_i(t) = \max_{j: j \in \mathsf{out}(i), t + \delta_{i,j} \le \Delta_{s,d}} \Big\{ \hat{P}_i(t) w_i(t) + (1 - \hat{P}_i(t))(f_j(t + \delta_{i,j})) \Big\}, \forall t, \forall i \in \mathcal{V} \setminus \{d\}, \quad (10)$$

with initial condition

$$f_d(t) = U_{s,d}^{\mathsf{RS}}(t) = U_{s,d}^{\mathsf{NRS}} - \beta \cdot (t - \Delta_{u,v}^{\mathsf{F}}), \quad t = 0, 1, ..., \Delta_{s,d}.$$

Clearly, our objective is to calculate $f_s(0)$, which is the maximum revenue the driver can get. Based on the recursive structure in (20), we design a dynamic-programming algorithm (Algorithm 2) to obtain $f_s(0)$. We show the correctness and the time complexity of Algorithm 2 in the following theorem.

***Theorem** 3:* Algorithm 2 solves problem DART optimally and the time complexity is $O(\Delta_{s,d}^{\max} |\mathcal{V}|^2 + |\mathcal{V}|^3)$.

*Proof:* See Appendix IX-C. ∎

Our proposed algorithm (Algorithm 2) has a pseudo-polynomial time complexity in the sense that the complexity is polynomial in the value of the problem input $\Delta_{s,d}^{\max}$, but is exponential in the length of the problem input, i.e., $\log(\Delta_{s,d}^{\max})$ [15]. However, we should note that the complexity of Algorithm 2 is polynomial in the network size and thus the solution is amenable for practical implementation. Note that Theorem 3 together with Theorem 1 also shows that our problem DART is actually NP-complete in the weak sense [16].

## V. Discussions

In this section, we discuss how to ensure customers' fairness using our proposed ride-sharing mechanism in (1), how to extend our solution to model the drives' cost instead of only maximizing the revenue of the driver, how to extend our solution to pick up two or more riders instead of only picking up one rider, and how to use our solution as a building block to perform rider-sharing matching with future demand statistics taken into account.

### A. How to Ensure Customers' Fairness?

In this subsection, we discuss why our proposed ride-sharing mechanism in (1) can guarantee certain *fairness* for the customers. To measure customers' fairness, we define the following unit-cost metric for a customer from node $i$ to node $j$ who participates ride-sharing,

$$\mathsf{Unit\text{-}Cost}_{i,j} = \frac{\mathsf{Ride\text{-}Sharing\text{-}Payment}_{i,j} + \beta \tau_{i,j}}{\Delta_{i,j}^{\mathsf{F}}}, \quad (11)$$

where $\mathsf{Ride\text{-}Sharing\text{-}Payment}_{i,j}$ is the ride-sharing payment that the customer gives to the service provider, and $\tau_{i,j}$ is the total travel time of the customer and $\Delta_{i,j}^{\mathsf{F}}$ is the minimal travel time of the customer (if the driver follows the fastest path to deliver her). The denominator of (11), i.e., $\Delta_{i,j}^{\mathsf{F}}$, is the travel time that the customer *expects*, while the nominator of (11), i.e., $\mathsf{Ride\text{-}Sharing\text{-}Payment}_{i,j} + \beta \tau_{i,j}$, is the total cost of the customer during the ride-sharing trip, including the monetary cost $\mathsf{Ride\text{-}Sharing\text{-}Payment}_{i,j}$ and the time cost $\beta \tau_{i,j}$ if we interpret our discounting rate $\beta$ as the cost per unit of travel time of the customer. Thus, overall, $\mathsf{Unit\text{-}Cost}_{u,v}$ is the cost per unit of *effective* time for the customer.

Let us further make an assumption on the non-ride-sharing payment, i.e., $U_{i,j}^{\mathsf{NRS}}$. In current practice, the non-ride-sharing price normally depends on the total travel time/distance. As we discussed in Sec. II-D, we can assume that the driver follows the fastest path of the customer from node $i$ to node $j$ to deliver this customer in the non-ride-sharing case. Thus, we assume that,

$$U_{i,j}^{\mathsf{NRS}} = \alpha \cdot \Delta_{i,j}^{\mathsf{F}}, \quad (12)$$

where $\alpha$ is the price per unit of time in the non-ride-sharing case[8]. Then for our ride-sharing payment mechanism in (1), we can obtain the unit-cost as

$$\begin{aligned} \mathsf{Unit\text{-}Cost}_{i,j} &= \frac{U_{i,j}^{\mathsf{RS}}(\tau_{i,j}) + \beta \tau_{i,j}}{\Delta_{i,j}^{\mathsf{F}}} \\ &= \frac{U_{i,j}^{\mathsf{NRS}} - \beta(\tau_{i,j} - \Delta_{i,j}^{\mathsf{F}}) + \beta \tau_{i,j}}{\Delta_{i,j}^{\mathsf{F}}} \\ &= \frac{\alpha \cdot \Delta_{i,j}^{\mathsf{F}} + \beta \cdot \Delta_{i,j}^{\mathsf{F}}}{\Delta_{i,j}^{\mathsf{F}}} \\ &= \alpha + \beta. \end{aligned} \quad (13)$$

Therefore, both Rider-I and Rider-II in ride-sharing scenario have the same unit-cost. We thus say that our proposed payment mechanism in (1) ensures *fairness* among ride-sharing customers.

[8] in our simulation in Sec. VI, we use $\alpha = 0.4$ according to the real-world New York taxi rate [25].

### B. How to Model Driver's Cost?

Previously, we modeled customers' payments, which are the revenues of drivers or service providers as well. We make some simplification by ignoring the cost of drivers, like fuel cost and time cost. Service providers should take it into account in oder to benefit current drivers and appeal potential drivers. This can be done implicitly by carefully designing the payment parameters, like unit time payment, unit distance payment, and/or the discounting rate $\beta$ to ensure significant amount of profits against the cost of drivers. Also, we can expand our model to explicitly consider the cost of drivers in a trip. Particularly, we assume the unit time cost of a driver is fixed, denoted as $\gamma$, which accounts for all the cost of drivers mentioned above. A driver's profit is equal to the payment(s) she gets minus her travel cost. Taking the driver's time cost into account, both Stage-I and Stage-II problem should be revised by changing the calculation of driver's profit to address driver's cost. Both algorithms can be adapted to the new problem easily.

### C. How to Pick up Two or More Riders?

In Stage-II, we assume that after picking up Rider-II, we simply find a path to deliver both customers and maximize the profit without considering other potential customers. As a result, the path we design is similar with the shortest path. As showed in the simulation VI-C, both Rider-I and Rider-II's travel time is significantly less than the maximum travel delay allowed for ride-sharing. What's more, Rider-II's travel time is very close to the minimum travel delay. These indicate that at Stage-II, by taking picking up other customers along the $\mathcal{P}^{\mathsf{RS}}$ into account, we can further exploit the additional travel time allowed in the ride-sharing scenarios. To do so, we then reformulate Stage-II, rename it as Stage-II' for distinction and introduce Stage-III' problem. First, Stage-III' is that, assumed a driver has picked up another new $p-q$ customer along the $\mathcal{P}^{\mathsf{RS}}$, the driver need to delivery all three customer before their deadlines and maximize her revenues. As one can see, Stage-III' is similar with the original Stage-II problem and the solution to Stage-II can be easily adapted to solve Stage-III'. Second, Stage-II' becomes to find a path such that both customer can be delivered in time and the expected profit along the route, which is the probability-weighted sum of solutions to all scenarios in Stage-III', is maximized. This is similar to the Stage-I problem and the algorithm to Stage-I can be generalized to solve Stage-II'. In order to consider picking up totally three customers along the road, we have to form a three-stage problem, which is solvable but more complex. This can be generalized to cases with many riders using multi-stage formulation, but the complexity gets higher as the number of riders considered in a single trip increases.

### D. How to Consider Demand-aware Ride-sharing Matching?

Supposed at a time slot, a service provider needs to dispatch $n$ available drivers to server $m$ customers. We assume that each car can at most have two customers on aboard at every time slot (this can be extended as Sec. V-C) and that at each time slot, at most one new customer can be assigned to a driver (this can be extended like [4]). The profit of assigning a customer to an empty car is expected profit attained from Stage-I of our approach. The profit of assigning a customer to a car that is delivering a customer or on her way to a customer is calculated by Stage-II of our approach. Then a maximum weighted bipartite matching algorithm [26] can be applied to solve the problem.

## VI. Simulation

We carry out numerical experiments based on real-world data traces to evaluate the performance of our proposed solution and the benefit of ride-sharing. Our objectives are two-fold. First, we compare the performance of our algorithm with that of a demand-oblivious greedy-routing scheme [4], which serves as a baseline. Second, using our proposed solution as a building block, we gauge the benefit of ride-sharing to customers, drivers and the society. As compared to the evaluation in [4] [3], our simulation uniquely demonstrates the benefit of exploiting demand statistics to improve ride-sharing performance.

### A. Dataset and Methodology

**Dataset.** We use a public dataset of taxi trips in Manhattan, New York City [27]. We consider $10.6$ GB data of $58,271,050$ trips in 6 months (2016-01-01 to 2016-06-30). For each trip, we obtain the pick-up time and location as well as the drop-off location. We use such trip information to construct the travel demand model (Sec. II-C).

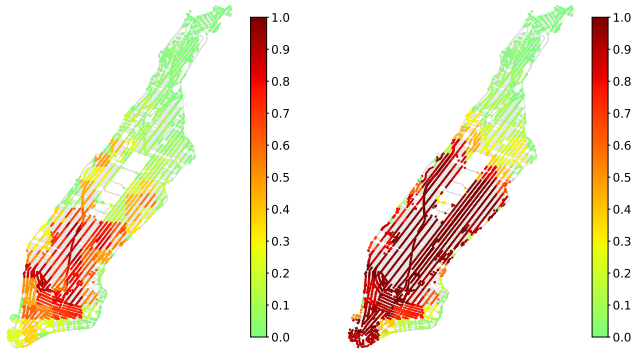
We get the Manhattan map from OpenStreeMap [28] and then construct the road graph $\mathcal{G}_0$ using the open-sourced NetworkX [29] and OSMnx [30] packages. For each edge $(u_0, v_0)$ in the node-based graph $\mathcal{G}_0$, we get its length $d_{u_0,v_0}$ (km). We further assume that the taxi speed is 15km/h according to a mobility report from NYC Department of Transportation [31]. The travel time of edge $(u_0, v_0)$ is then estimated as

$$\xi_{u_0,v_0} = \left\lceil \frac{60 \cdot d_{u_0,v_0}}{15} \right\rceil \text{ (minutes)}.$$

Next we divide the whole area of 59.5 km$^2$ into small rectangular regions of length 700 meters and width 600 meters. And we have in total 147 regions. We then construct the region-based graph $\mathcal{G}$ according to the procedure in Sec. II-A. Recall that the travel time $\delta_{u,v}$ (minutes) of each link $(u, v) \in \mathcal{E}$ in the region-based graph $\mathcal{G}$ is the travel time of the fastest path from the representative node of region $u$ to that of region $v$ in the road graph.

**Demand Statistics.** We set the slot length to be one minute and consider one whole day as the time span. Thus, we have in total $24 \times 60 = 1440$ slots. In our problem formulation, we need to input parameters $P_i(t)$ and $P_{i,j}(t)$[9]. Based on our dataset with 182-day trip information, for any node

[9]Note that in our system model, slot $t$ is relative to the start time of Rider-I at node $s$, which we denote as slot 0. However, here in our simulation, we use absolute time, i.e., slot $t$ is the $t$-th minute in a day ($t \in [1, 1440]$). Thus if the start time of Rider-I at node $s$ is slot $t_0$, then we should interpret $P_i(t)$ in our system model as $P_i(t + t_0)$ in our simulation here.



(a) 2:00 a.m. (b) 6:00 p.m.

Fig. 3. Probability heat maps for $P_i(t)$ in Manhattan.

$i \in [1, 147]$, any slot $t \in [1, 1440]$ and any day $k \in [1, 182]$, we define

$$A_{i,t,k} = \begin{cases} 1, & \text{If there is at least one customer} \\ & \text{at node } i \text{ at slot } t \text{ in day } k; \\ 0, & \text{Otherwise.} \end{cases}$$

Then we estimate $P_i(t)$ as $\frac{\sum_{k=1}^{182} A_{i,t,k}}{182}$, which is the frequency of days when at least a customer appears at node $i$ at slot $t$ (e.g., 2 pm).

We further estimate $P_{i,j}(t)$ as the average ratio of the number of customers at node $i$ at slot $t$ that go to node $j$ to the total number of customers at node $i$ at slot $t$ among all those days when there exists at least one customer at node $i$ at slot $t$.

We show $P_i(t)$ for the whole 147 regions at 2:00 a.m. (off-peak hour) and 6:00 p.m. (peak hour) in Fig. 3. As seen, at 6:00 p.m., there are more travel demands in the lower Manhattan and midtown Manhattan than the upper Manhattan. Even at 2:00 a.m., the demand in the lower Manhattan is quite high, as it is the business and cultural center of the City of New York.

**Parameters.** In our simulation, we set the maximum travel delay from node $i$ to node $j$, $\Delta_{i,j}^{\max}$ as

$$\Delta_{i,j}^{\max} = \alpha \cdot \Delta_{i,j}^{\mathsf{F}}, \quad \forall i, j \in \mathcal{V}, \tag{14}$$

where $\Delta_{i,j}^{\mathsf{F}}$ is the minimum travel delay from node $i$ to $j$ and $\alpha \geq 1$ is a parameter to tune the maximum travel delay. We set the non-ride-sharing payment $U_{i,j}^{\mathsf{NRS}}$ as

$$U_{i,j}^{\mathsf{NRS}} = 0.4 \cdot \Delta_{i,j}^{\mathsf{F}}, \quad \forall i, j \in \mathcal{V}, \tag{15}$$

where the coefficient 0.4 (customer's travel payment per minute) is in line with the real-world New York taxi rate [25].

We further use the tuple $(s, d, t_0)$ to denote a problem instance where $s$, $d$, and $t_0$ are respectively the source node, the destination node, and the pick-up time/slot of Rider-I. Then in following subsections, we randomly choose 1000 $(s, d)$ pairs and $t_0$ at each hour among a day, i.e., in total $1000 \times 24 = 24,000$ problem instances. We evaluate our algorithm and other algorithms on those $24,000$ instances and compute the average performance.

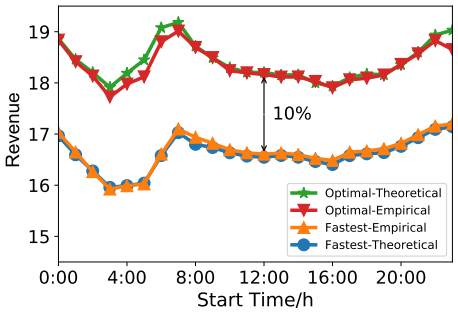
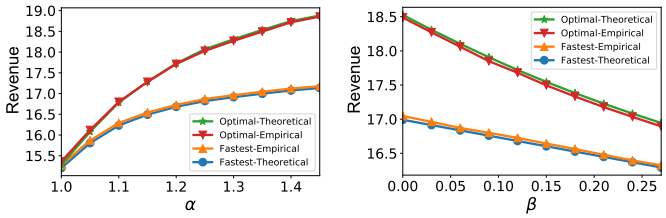



Fig. 4. Average revenues over time with $\alpha = 1.3$ and $\beta = 0.05$. Empirically, our proposed algorithm gains 10% more revenue than the fastest path algorithm.




(a) $\beta = 0.03$. (b) $\alpha = 1.3$.

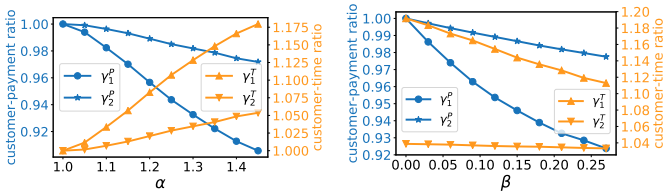
Fig. 5. Average revenues with different $\alpha$ and $\beta$.

### *B. Performance Evaluation of Our Algorithm as Compared to the Baseline*

In this paper, we propose a demand-aware ride-sharing routing scheme by exploiting the demand statistics. As we mentioned in Sec. II-B, in the common practice without demand statistics, the driver "greedily" follows the shortest/fastest path from $s$ to $d$ to deliver Rider-I and potentially picks up new customers along the way. To see the benefit of exploring demand statistics, we use this greedy-routing scheme as a demand-oblivious baseline and compare it with the demand-aware algorithm we propose.

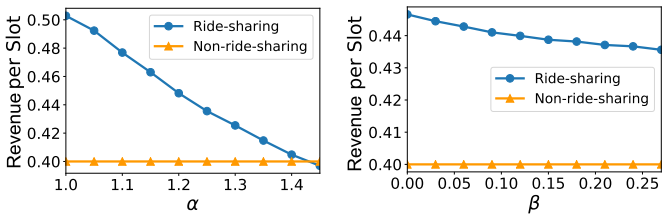
For both our algorithm and the baseline algorithm, in each problem instance $(s, d, t_0)$, we first get the *theoretical revenue* according to our problem DART. We then apply the two algorithms to the real-world trace and obtain the *empirical revenue*, as the average revenue of the driver over 182 days. (See Appendix IX-D).

Fig. 4 shows the revenues of our algorithm and the baseline algorithm over 24 hours of a day, averaged over 182 days. As we can see, on average our demand-aware algorithm achieves 10% more revenue for the service provider (or equivalently the driver) than the baseline demand-oblivious algorithm. This shows that exploiting demand statistics indeed brings substantial benefits in ride-sharing. While one may think 10% is not significant, the detailed simulation result shows that the driver tends to pick up a Rider-II with much shorter travel distance than that of Rider-I, because travel distances of most demands in the dataset are short. Hence, 10% extra revenue is quite substantial under this setting. One of our future work is to consider sequential ride-sharing for the cases that Rider-I travels a long distance. We also observe that the




(a) $\beta = 0.21$. (b) $\alpha = 1.3$.

Fig. 6. Customer-time ratios and customer-payment ratios with different $\alpha$ and $\beta$.




(a) $\beta = 0.21$. (b) $\alpha = 1.3$.

Fig. 7. Drivers' revenues per slot with different $\alpha$ and $\beta$.

theoretical revenue is close to the empirical one, which verifies the effectiveness of our modeling approach. In addition, we observe that in the morning rush hours (6 - 9 a.m.), the driver can obtain high revenue, due to the high travel demand.

We further evaluate the impacts of parameters $\alpha$ and $\beta$ to the revenue, shown in Fig. 5. Recall that we set the maximum travel delay of the customer from node $u$ to node $v$ to be $\alpha \cdot \Delta^{\mathsf{F}}_{u,v}$ (see (14)). If we increase $\alpha$, we allow more travel delay of the customers, which potentially increases the chance of ride-sharing. Thus, the driver may get more revenue, as shown in Fig. 5(a). Meanwhile, as $\alpha$ increases, the travel time may increase (as shown later in Fig. 6(a)) and the customers get more discounts (i.e., the driver gets less revenue). Overall, this leads to a diminishing return observation in Fig. 5(a). Also, recall that $\beta$ is the discounting rate for the customer (see (1)). Thus, larger $\beta$ leads to less revenue of the driver, as shown in Fig. 5(b). Meanwhile, when $\beta$ increases, the driver has incentive to travel along a faster path and thus the customer's travel time decreases (as shown later in Fig. 6(b)). Overall, the driver's revenue decreases almost linearly as $\beta$ increases.

### C. The Benefit of Ride-sharing to Customers

We use our solution as a tool to evaluate the benefit of ride-sharing. From a customer's perspective, we evaluate his travel time and payment in both ride-sharing and non-ride-sharing scenarios. In the non-ride-sharing scenario, we assume that each customer shall travel along the fastest path and thus the customer from node $i$ to node $j$ experiences a travel time of $T^{\mathsf{NRS}}_{i,j} = \Delta^{\mathsf{F}}_{i,j}$, and the payment is $U^{\mathsf{NRS}}_{i,j} = 0.4 \cdot \Delta^{\mathsf{F}}_{i,j}$ (see (15)).

In the ride-sharing scenario and each problem instance $(s, d, t_0)$, we run our algorithm using real-world data trace and then get the actual travel time of the customer(s). The travel time of a ride-sharing customer (which could be either Rider-I or Rider-II) from node $i$ to node $j$ is denoted as $T^{\mathsf{RS}}_{i,j}$. The ride-sharing payment of a customer from node $i$ to node $j$, i.e., $U^{\mathsf{RS}}_{i,j}(T^{\mathsf{RS}}_{i,j})$ is calculated according to (1). We then compute the ratios of the travel time (and payment) under the ride-sharing scenario to those under non-ride-sharing as

$$\gamma^T_{i,j} = \frac{T^{\mathsf{RS}}_{i,j}}{T^{\mathsf{NRS}}_{i,j}}, \text{ and } \gamma^P_{i,j} = \frac{U^{\mathsf{RS}}_{i,j}(T^{\mathsf{RS}}_{i,j})}{U^{\mathsf{NRS}}_{i,j}}.$$

For Rider-I, we obtain the average of $\gamma^T_{i,j}$ (resp. $\gamma^P_{i,j}$) among 24000 instances, denoted as $\gamma^T_1$ (resp. $\gamma^P_1$), which we call the *Rider-I customer-time ratio* (resp. the *Rider-I customer-payment ratio*). Similarly, for Rider-II, we obtain the *Rider-II customer-time ratio* $\gamma^T_2$ and the *Rider-II customer-payment ratio* $\gamma^P_2$. The results responding to different parameters $\alpha$ and $\beta$ are shown in Fig. 6.

Fig. 6 shows that, as compared to the non-ride-sharing case, customers participating in ride-sharing can save up to 9% in payment if they can tolerate 17% more travel time. In addition, as seen from Fig. 6(a), When $\alpha$ increases and thus the maximum travel delay is larger, the driver can deliver customers with more detours, resulting in longer travel time of the customers (i.e., the customer-time ratio increases). According to our payment mechanism defined in (1), the customer's payment is lower when the travel time increases for fixed discounting rate $\beta$, i.e., the customer-payment ratio decreases. Further, from Fig. 6(b), we can see that both the customer-time ratio and customer-payment ratio decrease as $\beta$ increases. This is because: (i) the customer's travel time decreases if the discounting rate $\beta$ increases as the driver tends to detour less, and (ii) the customer pays less when $\beta$ increases and saves less when the travel time decreases according to (1) in our payment model (Section II-D), but the overall effect is that the customer's payment will decrease as $\beta$ increases.

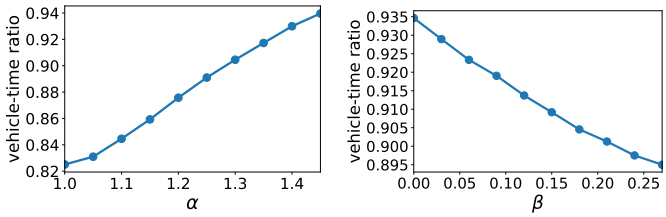
Also, we can see that Ride-I suffers more extra delay. Intuitively, Rider-II only gets detoured when the driver delivers Rider-I first, but for Rider-I, the driver may detour to pick up Rider-II and may also deliver Rider-II first. Besides, our demand-aware routing strategy generates a planning path which detours from the fastest path of Ride-I so as to explore more ride-sharing opportunities, leading to a extra delay of Rider-I. As a compensate, Rider-I gets more discount according to our payment scheme (Section II-D) and as shown in Fig. 6, which shows the fairness of our payment scheme.

### D. The Benefit of Ride-sharing to Drivers

In Sec. VI-B, we have evaluated the total revenue of drivers. However, besides the total revue, drivers also care about their revenue per unit time [32]. Thus, in this section, we evaluate the *revenue per slot* of drivers in both ride-sharing and non-ride-sharing scenarios.

In the non-ride-sharing scenario, as we set up in (15), the revenue per slot is fixed as 0.4. In the ride-sharing scenario and each problem instance, we run our algorithm over the real-world data trace and then get the actual total travel time and revenue of the driver. Then the revenue per slot of the driver equals to the driver's revenue divided by the driver's total travel time. We show the results in Fig. 7.

In Fig. 7, we can see that the ride-sharing revenue per slot decreases as $\alpha$ or $\beta$ increases. When $\alpha$ increases, i.e., the




(a) $\beta = 0.21$. (b) $\alpha = 1.3$.

Fig. 8. Vehicle-time ratio with different $\alpha$ and $\beta$.

allowable maximum travel delay increases, the driver may detour more to explore more ride-sharing opportunities to pursue higher revenue. However, the resulting travel time also increases and the overall effect is that the revenue per slot decreases as $\alpha$ increases. In addition, as the discounting rate $\beta$ increases, the customer gets more discounts due to additional travel delay and thus the driver gets less revenue per slot. Overall, in terms of revenue per slot, ride-sharing always outperforms non-ride-sharing and the driver can gain up to 26% extra revenue per slot when participating in ride-sharing.

### E. The Benefit of Ride-sharing to the Society

As discussed in Sec. I, ride-sharing has well-recognized societal benefits. In this subsection, we consider the total travel time of the drivers, which we called *vehicle time*, in both ride-sharing and non-ride-sharing scenarios. We argue that the reduction of vehicle time directly indicates the reduction of social travel distance which amounts to air pollution reduction and the alleviation of traffic congestion.

In each problem instance $(s, d, t_0)$, when using our demand-aware routing algorithm for ride-sharing, suppose that the driver picks up a new customer from node $i$ to node $j$ and the resulting ride-sharing path is $\mathcal{P}^{\mathsf{RS}}$. Then the vehicle time with ride-sharing to deliver both riders is $T^{\mathsf{RS}}_{\mathsf{society}}$ which is difference between the time when the driver picks up Rider-I and the time when she finishes delivering both Rider-I and Rider-II. On the other hand, without ride-sharing, if we need to deliver both riders (one from node $s$ to node $d$ and one from node $i$ to node $j$), we need two vehicles and the vehicle time is $T^{\mathsf{NRS}}_{\mathsf{society}} = \Delta^{\mathsf{F}}_{i,j} + \Delta^{\mathsf{F}}_{s,d}$. We now evaluate the *vehicle-time ratio* as $T^{\mathsf{RS}}_{\mathsf{society}}/T^{\mathsf{NRS}}_{\mathsf{society}}$. The average ratios among all problem instances with regard to different $\alpha$ and $\beta$ are shown in Fig. 8.

We can see that when we increase $\alpha$ and thus the allowable maximum travel delay, the vehicle time increases as the driver may detour more to exploit more ride-sharing opportunities. Similarly, when we increase the discounting rate $\beta$, the driver tends to detour less to ensure her revenue and thus the vehicle time decreases. Overall, ride-sharing can reduce the vehicle-time by up to 17% as compared to non-ride-sharing.

## VII. Related Work

### A. Taxi Recommender Systems

Some Taxi Recommender Systems have been developed for a taxi driver to reduce idle periods and increase her income. In [32], the authors suggest a sequence of road segments for a taxi driver considering the probability of pick-up and traveling cost. In [14], the authors recommend parking places and the corresponding routes with high potential of picking up a customer. However, existing works didn't consider the ride-sharing case under which a driver can increasing her income by picking up more customers along the route towards Rider-I's destination.

### B. Real time Dispatching Systems

The first dispatching system considering dynamic taxis ride-sharing in [6] provides a heuristic algorithm to match travel requests and taxis while minimizing additional travel distance. In [4], the authors further consider real time ride-sharing system with high capacity vehicles and provide a reactive anytime optimal algorithm. A carpooling recommendation system is designed in [6], by which a passenger is assigned to a vacant or carpoolable taxicab with a unified method. However, they focused on vehicle assignment problem or trip-sharable passengers group formation problem, and didn't exploit the potential of carefully routing to increasing the probability of ride-sharing or the profit of a taxi driver. We focus on route design for each taxi driver by utilizing demand statistic information

### C. Vehicle Routing for Ride-sharing Systems

Many kinds of vehicle routing problems have been studied for decades [33] [34]. The most related one is dynamic dial-a-ride Problem(DARP) [35]. However, we focus on modern real-time ride-sharing systems where the number of demand is significantly large and the travel demand pattern is more clear. To the best of our knowledge, no of them consider exploiting future demand statistics to enhance the routing performance in ride-sharing systems.

## VIII. Concluding Remarks

In this paper, we propose a general framework incorporating travel demand statistics to improve the performance of ride-sharing systems, with an emphasis on the routing module. We show that demand-aware ride-sharing routing is fundamentally a two-stage stochastic optimization problem. We then show that the problem is NP-complete and thus it is impossible to get the optimal solution in polynomial time unless P = NP. By exploiting the problem structure, we propose a dynamic-programming algorithm to optimally solve our problem with pseudo-polynomial time complexity. This also indicates that the problem is NP-complete in the weak sense [16]. We use real-world dataset to show that our proposed demand-aware routing algorithm achieves 10% more revenue for the driver than the conventional demand-oblivious greedy-routing algorithm. In our technical report [36], we further discuss how to extend our results to more practical settings, including how to take the driver's cost into consideration, how to consider the cases of picking up two or more riders, and how to perform demand-aware ride-sharing matching. In particular, we believe that optimizing the demand-aware ride-sharing matching module would be an interesting and important future direction.

## REFERENCES


[1] UberPool. https://www.uber.com/ride/uberpool/.
[2] LyftLine. https://www.lyft.com/line.
[3] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 37, pp. 13 290–13 294, 2014.
[4] 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*, vol. 114, no. 3, pp. 462–467, 2017.
[5] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transportation Research Part B: Methodological*, vol. 57, pp. 28–46, 2013.
[6] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE ICDE*, 2013, pp. 410–421.
[7] A. Braverman, J. Dai, X. Liu, and L. Ying, "Fluid-model-based car routing for modern ridesharing systems," in *Proc. ACM SIGMETRICS*, 2017, pp. 11–12.
[8] A. Biswas, R. Gopalakrishnan, T. Tulabandhula, K. Mukherjee, A. Metrewar, and R. S. Thangaraj, "Profit optimization in commercial ridesharing," in *Proc. ACM AAMAS*, 2017, pp. 1481–1483.
[9] D. Zhang, T. He, Y. Liu, S. Lin, and J. A. Stankovic, "A carpooling recommendation system for taxicab services," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 254–266, 2014.
[10] Y. Jia, W. Xu, and X. Liu, "An optimization framework for online ridesharing markets," in *Proc. IEEE ICDCS*, 2017, pp. 826–835.
[11] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He, "coRide: Carpool service with a win-win fare model for large-scale taxicab networks," in *Proc. ACM SenSys*, 2013.
[12] D. Zhang, T. He, F. Zhang, M. Lu, Y. Liu, H. Lee, and S. H. Son, "Carpooling service for large-scale taxicab networks," *ACM Transactions on Sensor Networks*, vol. 12, no. 3, 2016.
[13] X. Bei and S. Zhang, "Algorithms for trip-vehicle assignment in ridesharing," in *Proc. AAAI*, 2018.
[14] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in *Proc. ACM UbiComp*, 2011, pp. 109–118.
[15] M. R. Garey and D. S. Johnson, ""Strong" NP-completeness results: Motivation, examples, and implications," *Journal of the ACM*, vol. 25, no. 3, pp. 499–508, 1978.
[16] ——, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1979.
[17] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
[18] UberPOOL Just Got More Punctual. https://newsroom.uber.com/punctual-uberpool/.
[19] What percentage cut does Uber take from the total fare cost of a ride? https://www.quora.com/What-percentage-cut-does-Uber-take-from-the-total-fare-cost-of-a-ride/.
[20] How Does UberPool Pricing Really Work? http://therideshareguy.com/how-does-uberpool-pricing-work/.
[21] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2009.
[22] UberPOOL: Your Questions, Answered. https://www.uber.com/en-SG/blog/uberpool-your-questions-answered/.
[23] S. Biswas, A. Ganguly, and R. Shah, "Restricted shortest path in temporal graphs," in *Proc. Springer DEXA*, 2015, pp. 13–27.
[24] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2009.
[25] Taxicab Rate of Fare. http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml.
[26] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.
[27] TLC Trip Record Data. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[28] OpenStreetMap. https://www.openstreetmap.org.
[29] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. SciPy*, 2008, pp. 11–15.
[30] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.
[31] New York City Mobility Report. http://www.nyc.gov/html/dot/downloads/pdf/mobility-report-2016-screen-optimized.pdf.
[32] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, "A cost-effective recommender system for taxi drivers," in *Proc. ACM SIGKDD*, 2014, pp. 45–54.
[33] V. Pillac, M. Gendreau, C. Guret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1 – 11, 2013.
[34] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *European Journal of Operational Research*, vol. 202, no. 1, pp. 8 – 15, 2010.
[35] Y. Molenbruch, K. Braekers, and A. Caris, "Typology and literature review for dial-a-ride problems," *Annals of Operations Research*, 2017.
[36] Q. Lin, L. Deng, J. Sun, and M. Chen, "Optimal demand-aware ride-sharing routing," *CUHK Technical Report*, 2017. [Online]. Available: https://www.ie.cuhk.edu.hk/~mhchen/papers/DART.tr.pdf


## IX. APPENDIX

### A. Proof of Theorem 1

We reduce our problem to the classic restricted shortest path (RSP) problem, which is NP-Complete even for directed-acyclic graph [16], [23].

We construct a special case of our problem $\mathbf{P}$ as follows.

- We set $\beta = 0$.
- For each node $i$, we denote the fastest path between $i$ and the destination node $t$ is $\mathcal{F}_i$. We pick an arbitrary node $j_i \neq i$ in the path $\mathcal{F}_i$. Then we assume $p_{ij_i} > 0$ and $p_{ij} = 0, \forall j \neq j_i$. We set $\delta_{ij_i}$ as the total travel time between the node $i$ and node $j_i$ along the path $\mathcal{F}_i$. We let $p_{ij_i}$ to be arbitrarily set but let

$$U^{\mathsf{RS}}_{i,j_i} \tag{16}$$

Thus, for any feasible[10] $x$, we have $w_{ij_i}(x) = U^{\mathsf{RS}}_{s,d} + U^{\mathsf{RS}}_{i,j_i}$ be setting the path $\mathcal{P}$ to be path $\mathcal{F}_i$ in State-2 problem. Hence,

$$p_i(x) = p_{ij_i}, \tag{17}$$

and

$$w_i(x) = p_{ij_i} \cdot w_{ij_i}(x) = U^{\mathsf{RS}}_{s,d} + 1. \tag{18}$$

Then in our Stage-1 problem, the objective function is equivalent to

$$\begin{aligned}
&(U^{\mathsf{RS}}_{s,d}+1)\sum_{i=1}^{n_{\mathcal{P}}}\left(\prod_{j=1}^{i-1}(1-P_{v_j})\right)P_{v_i} + U^{\mathsf{RS}}_{s,d}\prod_{j=1}^{n_{\mathcal{P}}}(1-P_{v_j}) \\
&= (U^{\mathsf{RS}}_{s,d}+1)\left(1-\prod_{j=1}^{n_{\mathcal{P}}}(1-P_{v_j})\right) + U^{\mathsf{RS}}_{s,d}\prod_{j=1}^{n_{\mathcal{P}}}(1-P_{v_j}) \\
&= (U^{\mathsf{RS}}_{s,d}+1) - \prod_{j=1}^{n_{\mathcal{P}}}(1-P_{v_j}),
\end{aligned} \tag{19}$$

where $P_{v_i} = p_{v_i,j_{v_i}}$.

Thus, the Stage-1 problem is equivalent to,

$$\begin{aligned}
(\mathbf{P}') \min_{\mathcal{P}\in\mathbb{P}} \quad & \sum_{j=1}^{n_{\mathcal{P}}} \log((1-P_{v_j})) \\
\text{s.t.} \quad & \tau_{s,d}(\mathcal{P}) \leq \Delta_{s,d}
\end{aligned}$$

- Problem $\mathbf{P}'$ is similar to the RSP problem, but the path cost is on the node rather than on the edge in RSP. Now we do the following transformation. Consider an

[10]Here $x$ is feasible if $x \leq \Delta_{s,d} -$ (travel time of $\mathcal{F}_i$).

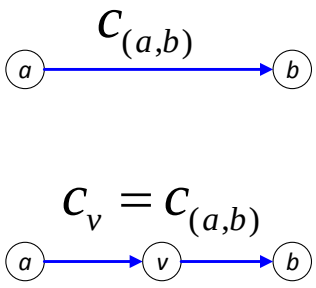



Fig. 9. Edge cost to node cost.

arbitrary RSP problem with graph $\mathcal{G}$. Each edge $e$ has a cost $c_e < 0$[11] and a travel time $t_e$. There is a hard deadline $T$ from the source node $s$ to destination node $d$. Then for each edge $e = (a, b)$, we insert a virtual node $v$ between $a$ and $b$. The new graph is denoted as $\mathcal{G}'$. We set the node-$v$ cost $c_v$ to be $c_e$, and set node-$a$ cost $c_a$ and node-$b$ cost $c_b$ to be 0. We set the travel time to be $\frac{t_e}{2}$ for edge $(a, v)$ and edge $(v, a)$. In the new graph, for each node $i$, we tune the parameter $p_{ij_i}$ to be $1 - \exp(c_i)$. Thus, we have $\log(1 - P_i) = \log(1 - P_{ij_i}) = c_i$. Under this transformation, we immediately see that problem $\mathbf{P}'$ is equivalent to the RSP problem.

Since RSP is NP-Complete, we prove that our problem $\mathbf{P}$ is NP-complete also.

### B. Proof of Theorem 2

Recall that $\mathbb{P}_{i,j,d}$ is the set of the paths that start at node $i$ and pass through *both* node $d$ and node $j$, and they either first reach node $j$ and end at node $d$, or first reach node $d$ and end at node $j$. $\mathbb{P}_{i,j,d}$ can be divided into two categories; one is the set of paths which reach $j$ first and end at $d$ (denoted as $\mathbb{P}^1$), and the other is the set of paths which reach $d$ first and end at $j$ (denoted as $\mathbb{P}^2$). Thus, every path $\mathcal{P}^{\mathsf{RS}} \in \mathbb{P}_{i,j,d}$ is either in $\mathbb{P}^1$ or $\mathbb{P}^2$. $\{\mathbb{P}^1, \mathbb{P}^2\}$forms a partition of $\mathbb{P}_{i,j,d}$

a) We first show that Stage-2$(i, j, t)$ is feasible if and only if either $\mathcal{P}_1$ or $\mathcal{P}_2$ satisfies both (3b) and (3c). (See (9) for the definition of $\mathcal{P}_1$ or $\mathcal{P}_2$). The 'if' part is trivial. For the 'only if' part, we instead show its contrapositive, i.e., if both $\mathcal{P}_1$ and $\mathcal{P}_2$ don't satisfy (3b) or (3c), Stage-2$(i, j, t)$ is not feasible.
To proceed, we first show that given $\mathcal{P}_1$ is infeasible, every path $\mathcal{P}$ in $\mathbb{P}^1$ is infeasible. Divide $\mathcal{P}$ into two segment: $(\mathcal{P}_{i,j}, \mathcal{P}_{j,d})$. The travel time of each segment is larger than that of corresponding segment of $\mathcal{P}_1$. Thus $\tau_{i,j}(\mathcal{P}) \geq \tau_{i,j}(\mathcal{P}_1)$ and $\tau_{i,d}(\mathcal{P}) \geq \tau_{i,d}(\mathcal{P}_1)$. Thus if $\mathcal{P}_1$ is infeasible, then all paths in $\mathbb{P}_1$ are infeasible. The same argument applies to $\mathbb{P}_2$ and $\mathcal{P}_2$ as well. Thus if $\mathcal{P}_2$ is infeasible, then all paths in $\mathbb{P}_2$ are infeasible. Finally, we conclude that if both $\mathcal{P}_1$ and $\mathcal{P}_2$ don't satisfy (3b) or (3c), Stage-2$(i, j, t)$ is not feasible.

[11]We assume the edge cost to be negative in the RSP problem here for simplicity. The reduction can be easily generalized to positive edge cost case because we consider DAG.

b) We then show that If Stage-2$(i, j, t)$ is feasible, then either $\mathcal{P}_1$ or $\mathcal{P}_2$ is an optimal solution.
Proof: From a), if Stage-2$(i, j, t)$ is feasible, then either $\mathcal{P}_1$ or $\mathcal{P}_2$ is feasible. Every path $\mathcal{P} \in \mathbb{P}_1$, $\tau_{i,j}(\mathcal{P}) \geq \tau_{i,j}(\mathcal{P}_1)$ and $\tau_{i,d}(\mathcal{P}) \geq \tau_{i,d}(\mathcal{P}_1)$; thus, it returns the largest objective value among $\mathbb{P}_1$ when we choose $\mathcal{P} = \mathcal{P}_1$. If $\mathcal{P}_1$ is not feasible, then all paths in $\mathbb{P}_1$ are also infeasible. Following the same argument, it returns the largest objective value among $\mathbb{P}_2$ when we choose $\mathcal{P} = \mathcal{P}_2$. If $\mathcal{P}_2$ is not feasible, then all paths in $\mathbb{P}_2$ are also infeasible. Finally, either $\mathcal{P}_1$ or $\mathcal{P}_2$ is the optimal solution.

### C. Proof of Theorem 3

a) Complexity: We first show that the complexity of our algorithm is $O(\Delta_{s,d}|\mathcal{V}|^2 + |\mathcal{V}|^3)$. In Algorithm 2, Step 1 is $O(|\mathcal{V}|^3)$. Knowing the travel time of the fastest paths of all pairs, each Stage-2$(i, j, t)$ instance can be sloved in $O(1)$ and with totally $\Delta_{s,d}|\mathcal{V}|^2$ instances in Step 2, the complexity of Step 2 is thus $O(\Delta_{s,d}|\mathcal{V}|^2)$. Step 3 computes $P_i(t)$, $\Delta_{s,d}|\mathcal{V}|$ times and each invokes $O(|\mathcal{V}|)$ operation. Thus Step 3 takes $O(\Delta_{s,d}|\mathcal{V}|^2)$. Step 4 also takes $O(\Delta_{s,d}|\mathcal{V}|^2)$, following the same analysis of Step 3. Step 7- 15 actually iterates all the edges $|\Delta_{s,d}|$ times and the complexity is $O(\Delta_{s,d}|\mathcal{E}|)$. Overall, the complexity of Algorithm 2 is $O(\Delta_{s,d}|\mathcal{V}|^2 + |\mathcal{V}|^3)$

b) Correctness: As defined previously, $f_i(t)$ is the maximum expected total revenue the driver can obtain if she arrives at node $i$ at slot $t$ without picking up any new customer before node $i$. We claim that $f_i(t)$ satisfies the following recurrence:

$$f_i(t) = \max_{j:j\in\mathsf{out}(i), t+\delta_{i,j}\leq\Delta_{s,d}} \Big\{\hat{P}_i(t)w_i(t) + (1-\hat{P}_i(t))(f_j(t+\delta_{i,j}))\Big\}, \forall t, \forall i \in \mathcal{V}\setminus\{d\},$$

and,

$$f_d(t) = U^{\mathsf{RS}}_{s,d}(t) = U^{\mathsf{NRS}}_{s,d} - \beta\cdot(t - \Delta^{\mathsf{F}}_{u,v}), \quad t = 0, 1, ..., \Delta_{s,d}.$$

If the driver arrives at $d$ at $t$ without picking up other customers, she can get the payment of Rider-I, $U^{\mathsf{NRS}}_{s,d} - \beta \cdot (t - \Delta^{\mathsf{F}}_{u,v})$. Otherwise, if the driver arrives at some node $i$ ($i \neq d$) at time $t$, as discussed in Section IV-B, she either picks up another customer at node $i$ or goes to one of its neighbors (named $j$) without picking up any new customers, and gains expectantly $\hat{P}_i(t)w_i(t) + (1 - \hat{P}_i(t))(f_j(t + \delta_{i,j}))\Big\}$. And the maximum expected revenue $f_i(t)$ is the maximum among all its neighbors, which corresponds to the recurrence.
Finally, Algorithm 2 evaluates the recurrence of $f_i(t)$, as $t$ decreases from $\Delta^{\mathsf{max}}_{s,d}$ to 0 and $i$ increases from 1 to $n$. When evaluating $f_i(t)$, all values for the recurrence will have already been computed. At the end, our algorithm returns $f_0(s)$, which corresponds to the maximum revenue a driver can obtain.

*D. Empirical Procedure*

- We let the taxi driver follows the planning path $\mathcal{P}^{\mathsf{planning}}$ produced by our algorithm or the baseline algorithm and observes the trip information of day-$k$;
- Along the planning path $\mathcal{P}^{\mathsf{planning}}$, suppose the taxi driver comes to node $i$ at slot $t$ without picking up any new customer before node $i$. If there does not exist any customer at node $i$ at slot $t$, then the driver comes to the next node along the planning path $\mathcal{P}^{\mathsf{planning}}$. If there exists at least one customer at node $i$, the drive uniformly picks up one. If the customer is not feasible, the driver comes to the next node along the planning path $\mathcal{P}^{\mathsf{planning}}$. If the customer is feasible, the driver solve the State-2 problem and get the real path $\mathcal{P}^{\mathsf{real}}$ and then follow $\mathcal{P}^{\mathsf{RS}}$ to deliver both customers;
- Empirical-Profit$(s, d, t_0; k)$ is calculated based on the resulting ride-sharing path $\mathcal{P}^{\mathsf{RS}}$ or the $\mathcal{P}^{\mathsf{planning}}$ if no new customer is picked up along the road.

We then get the *empirical revenue* for the problem instance $(s, d, t_0)$ by

$$\mathsf{Empirical\text{-}Revenue}(s, d, t_0) = \frac{\sum_{k=1}^{182} \mathsf{Empirical\text{-}Revenue}(s, d, t_0; k)}{182}.$$