A Probabilistic Approach for Demand-Aware Ride-Sharing Optimization

Qiulin Lin, Wenjie Xu, Minghua Chen
Information Engineering
The Chinese University of Hong Kong

Xiaojun Lin
Electrical and Computer Engineering
Purdue University

ABSTRACT

Ride-sharing is a modern urban-mobility paradigm with tremendous potential in reducing congestion and pollution. Demand-aware design is a promising avenue for addressing a critical challenge in ride-sharing systems, namely joint optimization of request-vehicle assignment and routing for a fleet of vehicles. In this paper, we develop a *probabilistic* demand-aware framework to tackle the challenge. We focus on maximizing the expected number of passenger pickups, given the probability distributions of future demands. The key idea of our approach is to assign requests to vehicles in a probabilistic manner. It differentiates our work from existing ones and allows us to explore a richer design space to tackle the request-vehicle assignment puzzle with a performance guarantee but still keeping the final solution practically implementable. The optimization problem is non-convex, combinatorial, and NP-hard in nature. As a key contribution, we explore the problem structure and propose an elegant approximation of the objective function to develop a dual-subgradient heuristic. We characterize a condition under which the heuristic generates a $(1 - 1/e)$ approximation solution. Our solution is simple and scalable, amendable for practical implementation. Results of numerical experiments based on real-world traces in Manhattan show that, as compared to a conventional demand-oblivious scheme, our demand-aware solution improves the passenger pickups by up to 46%. The results also show that joint optimization at the fleet level leads to 19% more pickups than that by separate optimizations at individual vehicles.

CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Mathematics of computing** → *Probabilistic algorithms*; • **Theory of computation** → *Probabilistic computation*; *Routing and network design problems*;

KEYWORDS

Ride-sharing, stochastic optimization, demand-aware routing, and request-vehicle assignment.

ACM Reference Format:
Qiulin Lin, Wenjie Xu, Minghua Chen and Xiaojun Lin. 2019. A Probabilistic Approach for Demand-Aware Ride-Sharing Optimization. In *The Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19), July 2–5, 2019, Catania, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3323679.3326512

Mobihoc '19, July 2–5, 2019, Catania, Italy

ACM ISBN 978-1-4503-6764-6/19/07…$15.00
https://doi.org/10.1145/3323679.3326512

1 INTRODUCTION

Dynamic ride-sharing or ride-sharing in short, is a modern paradigm for urban mobility, where passengers with similar itineraries and time schedules share riders on short-notice. Popular ride-sharing services, such as uberPOOL[1] and Lyftline[2], not only can provide convenient and cost-effective transportation to individuals, but also can create significant positive impacts on congestion and pollution. Take Manhattan as an example. The annual cost of congestion is more than $20 billion [28], which includes 24 million hours of time lost to sitting in traffic and an extra 500 million gallons of fuel burned. With ride-sharing, the authors in [6, 26] show that 98% of the Manhattan rides currently served by over 13,000 taxis could be served with just 3,000 vehicles of capacity four, with marginal increment in the trip delay. The aggregate trip distance, an indicator of commute time and gasoline consumption, can also be reduced by more than 30%. Overall, ride-sharing offers a clear opportunity for alleviating congestion, reducing pollution, and improving transportation efficiency.

A number of societal and economic issues need to be resolved in order to capitalize the maximum benefit of ride-sharing [4, 15]. On the technical front, the holy-grail problem is *how to jointly optimize the request-vehicle assignments and routing for a fleet of vehicles (considering future request-vehicle dynamics)*; see e.g., [7] for a discussion. It is a multi-slot vehicle pickup-and-delivery problem with ride-sharing in consideration, which is challenging to solve as even its single-slot version is already NP-hard [6, 29].

There are mainly three lines of studies in the literature [4, 16]. The first is to devise offline solutions, assuming full knowledge of future travel requests when making decisions. The offline problem is known to be NP-hard. The authors in [9] propose a 2.5-approximation algorithm under a constrained setting. Many studies consider efficient heuristics and metaheuristics [6, 10, 16, 20, 24, 30, 32]. These offline solutions may serve as performance benchmarks, but they are usually not practical. The second is to design demand-oblivious solutions, assuming zero knowledge of future requests when making decisions; see e.g., [6, 20, 24, 33]. While demand-oblivious solutions are more amenable for practical implementation, their performance can be very conservative as they do not adapt to future demand patterns. The third is to develop *demand-aware* solutions, assuming only distributional information on future travel requests when making decisions. Thanks to the advance in machine learning and data analytics, statistical knowledge of future

[1] UberPool. https://www.uber.com/ride/uberpool/
[2] LyftLine. https://www.lyft.com/line

travel requests can be efficiently learned by leveraging their regular hourly/daily/weekly patterns [31]. This approach opens up new design space for optimizing ride-sharing systems, and the initial success of developing demand-aware ride-sharing routing solution [22] is encouraging.

In this paper, we adopt the demand-aware mindset and develop a joint request-vehicle assignment and routing solution given the distributional information on future travel requests. A key idea in our design is to assign requests to vehicles in a *probabilistic* manner. This allows us to explore a richer design space to tackle the request-to-vehicle assignment puzzle with performance guarantee, but still keeping the final solution practically implementable.[3] Our particular study in this paper focuses on maximizing the expected number of new (ride-sharing) passengers picked up for a fleet of vehicles of capacity two, with passenger waiting time limit and passenger transportation deadline taken into account. The problem is important for enhancing the quality of service of the ride-sharing service platform such as uberPOOL and Lyftline. It is also equivalent to maximizing the revenue of VIA[4]. Our probabilistic approach is general and can be applied to optimize other system objectives. Our main contributions in this paper are as follows.

▷ In Sec. 2, we propose a general probabilistic framework for demand-aware ride-sharing optimization. The framework allows us to explore a bigger design space of joint request-vehicle assignment and routing with request statistics taken into account.

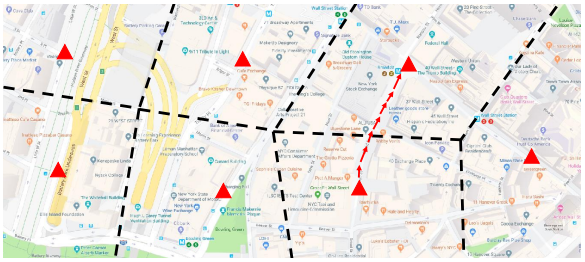
▷ In Sec. 3, applying the probabilistic framework, we formulate the important problem of joint request-vehicle assignment and routing for a fleet of vehicles given request statistics, in order to maximize the expected number of new (ride-sharing) passengers picked up. The problem is nonlinear and combinatorial, and we show that it is NP-hard. We then reformulate the problem into a linear-combinatorial one. We show that solving the reformulated problem gives an approximation solution to the original problem with an approximation ratio of $1 - 1/e$.

▷ In Sec. 4, the reformulated linear-combinatorial problem is still challenging, especially for large-scale instances; indeed, it is still NP-hard. To this end, by leveraging elegant insights from studying the dual of the re-formulated problem, we design a scalable heuristic solution. We further characterize a condition under which the heuristic generates an optimal solution to the reformulated problem, and hence a $(1 - 1/e)$ approximation solution to the original problem.

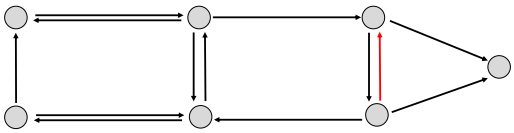
▷ In Sec. 5, we carry out numerical experiments based on real-world travel request traces in Manhattan. The results show that as compared to a conventional demand-oblivious scheme, our demand-aware solution improves the total number of passenger pickups by up to 46%. The results also show that joint optimization at the fleet level gives 19% more pickups than that obtained by individual vehicles carrying out optimization separately.

[3]We remark that our probabilistic approach is different from the randomization mechanism commonly used in algorithm design. Specifically, the joint assignment and routing solution derived by our approach is not a time-shared version of multiple optimal deterministic solutions for different sample-path realizations of the future travel requests.

[4]https://ridewithvia.com/. In VIA, the income of picking up a new passenger is a fixed amount regardless of the trip distance. The expected total revenue is thus proportional to the expected number of passengers picked-up.



(a) Transportation network $\mathcal{G}_0$.



(b) Region graph $\mathcal{G}$.

Figure 1: An example of the transportation network and the corresponding region graph. Each region has a representative node marked as a red triangle. The constructed region graph is shown in Fig. 1(b). Each node in the region graph represents a region. Each edge (u, v) in the region graph represents a fastest path in the transportation network from the representative node of region u to that of region v. For example, the edge in color red in the region graph in Fig. 1(b) represents the path in color red in the transportation network in Fig. 1(a).

Due to the space limitation, all proofs are included in our technical report [23], unless stated otherwise.

2 PROBLEM SETTINGS

Time is divided into slots of equal length and $\mathcal{T}$ is the set of the slots. We consider the scenario of a fleet of N vehicles of capacity two serving an urban area. We present the system modeling in this section and the problem formulation in the next section.

2.1 Transportation Network and Region Graph

We model the urban transportation network as a directed 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. Each edge $(u_0, v_0) \in \mathcal{E}_0$ is a road segment from node $u_0 \in \mathcal{V}_0$ to node $v_0 \in \mathcal{V}_0$, and the travel time of edge (u_0, v_0) is denoted as ξ_{u_0,v_0} (unit: slots). To introduce our travel request model later, we construct a region graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E}$, as shown in Fig. 1(a). In particular, we partition the transportation network $\mathcal{G}_0$ into multiple regions. Each node $v \in \mathcal{V}$ in the region graph $\mathcal{G}$ represents a region in $\mathcal{G}_0$. We assign a representative node in each region, from which all other nodes in the region can be reached in a small number of slots, e.g., 5 slots, illustrated as the black dots in Fig. 1(b)[5]. We add a directed edge (u, v) in $\mathcal{G}$, as illustrated in 1(a), if there exists a path from the representative node

[5]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 [25]. For a dense and regular urban road network like Manhattan, one can

of region u to that of region v in the road graph such that the major fraction of the path is within those two regions[6]. Let $\delta_{u,v}$ be the travel time of the fastest path.

We assume that when a travel request appears in a region, it only waits for a limited amount of time, e.g.,5 slots. As such, only the vehicles in the same region can pick up the passenger. This captures the observation that each passenger is associated with a waiting-time limit, e.g., 10 minutes for uberPOOL; vehicles outside the region cannot pick up the passenger on time. Hence, the "size" of regions can be determined by the waiting-time limit set by the ride-sharing system. Our road network and region graph models are similar to those used in the literature; see e.g., [13, 14, 22]. All our modeling and analysis are based on the region graph $\mathcal{G}$.

2.2 Travel Request

Each travel request consists of the time of request, a pickup region $v \in \mathcal{V}$ and a drop-off region $u \in \mathcal{V}$, a waiting time limit, and a trip deadline. We assume that the waiting-time limit for all travel requests to be the same, e.g., 5 slots, and it is used to properly gauge the size of regions as described in Sec. 2.1. The trip deadline for a travel request from v to u is denoted as $\Delta^{\max}_{v,u}$ slots, and we note that UberPOOL has already provided such "Arrive By" service[7]. In our study, we set the $\Delta^{\max}_{v,u} = \alpha \cdot \delta_{v,u}$, where $\alpha > 1$ represents the delay tolerance factor and $\delta_{v,u}$ denotes the travel time of the fastest path from the representative node in region v to that in region u.

Under the *demand-aware* setting, the time-dependent distributional information on (future) travel requests are available. Specifically, the probability that at time slot t, there are k passengers appearing at region v on $\mathcal{G}$ and going to region u is given as $p^t_{v,u,k}$. Without loss of practical relevance, we assume that there are at most $K > 0$ passengers going from v to u at any given time. Apparently, $\sum_{k=1}^{K} p^t_{v,u,k} = 1,\ \forall u \in \mathcal{V}$. We further assume that travel request arrivals across different nodes and in different time slots are independent to each other.

2.3 Vehicle State

At a given time t, each of the N vehicles is in one of the three states:

- (i) the vehicle is delivering two passengers on board and will not pick up any passenger,
- (ii) the vehicle is delivering one passenger on board and can pick up one more ride-sharing passenger,
- and (iii) the vehicle is empty and roaming towards a pre-selected hot spot (e.g., regions with good chances of picking up new passengers), with a self-selected and usually sufficiently large deadline.

Note that an empty vehicle in state (iii) can be regarded as a vehicle in state (ii) with one "virtual" passenger on board and is looking for picking up a new passenger along the way to the hot spot. Thus, from the modeling point of view, there is no difference between state (ii) and state (iii).

We also note that a vehicle may transit between states upon passenger pickup and delivery. For the empty vehicle in state (iii), its state will be updated to (ii) once it picks up a new passenger. Similarly, a vehicle's status will change from state (ii) to state (iii) if it picks up a ride-sharing passenger. Furthermore, a vehicle's state will change from state (i) or (ii) to state (iii) if the passenger(s) on board are delivered.

2.4 Request-Vehicle Assignment and Routing

Under the offline setting, the travel requests are assumed to be known in advanced. The requests are assigned to vehicles in the same regions by solving a combinatorial puzzle, taking into account the vehicle states and capacities, the waiting-time limit of the requests, and the system objective. This step is also called a trip-vehicle matching in the literature. Given the assignments, individual vehicles then compute the best pickup-delivery routes to transport the passengers. It is known that the joint assignment and routing problem is NP-hard and challenging to solve; see e.g., [9]. Furthermore, such an offline setting can be impractical because the exact information of future travel requests is usually not available.

Under the demand-aware setting, the distributional information of travel requests is available. In particular, at time t, with probability $p^t_{v,u,k}$, there are k travel requests appearing in region v and going to region u. We expect that such distributional information is much easier to obtain in practice. The key idea in our demand-aware design is to assign requests to vehicles in a probabilistic manner. Specifically, let $y^t_{i,v,u,k} \in \left[0, p^t_{v,u,k}\right]$ be the probability of the joint event that (i) k travel requests appear in region v going to region u at time t and (ii) one of the requests is assigned to vehicle i ($1 \le i \le N$). We remark that $y^t_{i,v,u,k}$'s are to be designed later, and they need to satisfy a set of conditions to be practically feasible, i.e., there exists a practical scheme that can realize the probabilistic assignment. We will present the feasibility conditions in the next section. Given $y^t_{i,v,u,k}$'s, upon k travel requests appearing in region u and going to region v at time t, we assign one of the k requests to vehicle i with probability $y^t_{i,v,u,k}/p^t_{v,u,k}$. The overall probability that vehicle i is assigned a travel request going from region v to u at time t is $\sum_{k=1}^{K} y^t_{i,v,u,k} \in [0,1]$. The probability that vehicle i is assigned a request in region v at time t (going to any region) is

$$y^t_{i,v} \triangleq 1 - \prod_{u \in \mathcal{V}} \left(1 - \sum_{k=1}^{K} y^t_{i,v,u,k}\right), \quad \forall 1 \le i \le N, v \in \mathcal{V}, t \in \mathcal{T}. \tag{1}$$

Given the probabilities of picking up (new or ride-sharing) passengers in individual regions in every time epoch, individual vehicles compute their own routes to maximize the expected reward or minimize the expected cost. As it will become clearer in the next section, the joint probabilistic assignment and routing problem admits bigger design space for designing efficient algorithms.

3 PROBLEM FORMULATION

In this section, we formulate the joint probabilistic request-vehicle assignment and vehicle routing problem for the fleet of N vehicles.

simply partition the district into regions of equal area. We adopt this method in the simulation in Sec. 5.

[6] 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 transportation network. Then there is an edge from u to v in the region graph if $T(u,v) \le \eta \cdot (T(u,k) + T(k,v))$ for any region $k \in \mathcal{G}$. In our simulation in Sec. 5, we set $\eta = 0.8$.

[7] uberPOOL Just Got More Punctual. https://newsroom.uber.com/punctual-uberpool/.

Without loss of generality, we assume that we start at time $t = 0$ and all vehicles are in state (ii), which also covers vehicles in state (iii) as there is no difference between the two from the modeling perspective. Our objective is to maximize the expected number of new or ride-sharing passenger pickups by the fleet from time $t = 0$ to the first time epoch involving a change in the vehicle states, i.e., a passenger arriving at the destination or an empty vehicle picking up a new passenger. To optimize the overall performance in a time horizon, e.g., a day, the optimization will be re-carried out at these state-changing time epochs.

3.1 Ride-Sharing Feasibility

Definition 1. *Suppose that vehicle i is transporting a passenger from region s_i to region d_i and passing through region v at time t. We say that a $s_i \to v \to \{u, d_i\}$ ride-sharing plan for vehicle i at time t is feasible if the vehicle can pick up another passenger from v to u at time t and deliver both passengers before their trip deadlines.*

Let $\mathcal{R}_{v,u,d_i}$ denote the set of two types of routes: (i) going from v to u and then to d_i, all by the fastest paths, and (ii) going from v to d_i and then to u, all by the fastest paths. Let $z^t_{i,v,u} \in \{0, 1\}$ be the indicator variable of whether a $s_i \to v \to \{u, d_i\}$ ride-sharing plan for the vehicle i at time t is feasible, i.e., whether the following problem has a feasible solution:

$$\begin{aligned} \max \quad & 1 \\ \text{s.t.} \quad & \gamma_{v,u}(r) \le \Delta^{\max}_{v,u}, && (2) \\ & \gamma_{s_i,v} + \gamma_{v,d_i}(r) \le \Delta^{\max}_{s_i,d_i}, && (3) \\ \text{var.} \quad & r \in \mathcal{R}_{v,u,d_i}, \end{aligned}$$

where $\Delta^{\max}_{v,u}$ and $\Delta^{\max}_{s_i,d_i}$ are trip deadlines, $\gamma_{s_i,v}$ is the amount of time the vehicle i already spent in traveling from s_i to v, and $\gamma_{v,u}(r)$ and $\gamma_{v,d_i}(r)$ are the travel times from region v to region u and region d_i along one of the two routes r in $\mathcal{R}_{v,u,d_i}$, respectively. The problem involves finding two fastest paths and some simple calculus and it is easy to solve. For each vehicle i, we need to solve the problem for every (s_i, d_i) pair, every (v, u) pair, and every $\gamma_{s_i,v} \in \left[0, \Delta^{\max}_{s_i,d_i}\right]$. The total complexity is polynomial in the size of the region graph and the maximum trip deadline. We note that these problems can be solved beforehand and the feasibility indicators $z^t_{i,v,u}$ and the corresponding feasible paths can be stored for lookup.

3.2 Feasibility of Request-Vehicle Assignments

Recall that $p^t_{v,u,k}$ is the probability of k travel requests appearing at time t in region v and going to region u. Variable $y^t_{i,v,u,k}$ is the probability of the joint event that (i) k travel requests appear in region u going to region v at time t and (ii) one of the requests is assigned to vehicle i ($1 \le i \le N$). For all $1 \le i \le N$, define

$$\mathbf{y_i} \triangleq \left[y^t_{i,v,u,k},\ \forall t \in \mathcal{T}, v, u \in \mathcal{V}, 1 \le k \le K\right].$$

The following proposition characterizes the feasible region of $\mathbf{y_i}$'s.

Proposition 2. *The request-vehicle assignment probability vectors $[\boldsymbol{y}_i, 1 \le i \le N]$ are feasible if and only if they satisfy that, for all $1 \le i \le N$, $1 \le k \le K$, $t \in \mathcal{T}$, and $v, u \in \mathcal{V}$,*

$$y^t_{i,v,u,k} \le p^t_{v,u,k}, \quad (4)$$

$$0 \le y^t_{i,v,u,k} \le z^t_{i,v,u}, \quad (5)$$

$$\sum_{i=1}^{N} y^t_{i,v,u,k} \le k \cdot p^t_{v,u,k}. \quad (6)$$

Let $\mathcal{Y}$ be the set of all feasible request-vehicle assignment vectors.

The inequalities in (4) state that the assignment probability $y^t_{i,v,u,k}$ should not be larger than $p^t_{v,u,k}$, the probability that the k requests appear. The inequalities in (5) state that the assignment probability can be positive only if the assignment is feasible (see Definition 1). The inequalities in (6) can be understood as follows. First, $y^t_{i,v,u,k}/P^t_{v,u,k}$ is simply the *conditional* probability that given k requests appearing in region v at time t and going to region u, one request is assigned to vehicle i. The inequalities in (6) say that the conditional expected total number of requests assigned to the fleet of N vehicles should be bounded by k.

Proposition 2 lays down an important foundation for our probabilistic demand-award approach. First, it characterizes the necessary and sufficient conditions for a request-vehicle assignment probability vector. Second, it says that there exists a scheme that can realize the probabilistic request-vehicle assignment, such that the conditional probability of vehicle i being assigned a passenger out of k appearing requests from v to u at time t is exactly the desired conditional probability $y^t_{i,v,u,k}/p^t_{v,u,k}$. Specifically, we present one such scheme as follows, which is very similar to the ones in [11, 19] for network caching system designs. The scheme is also applied for assigning requests to vehicles later in our simulations.

A probabilistic request-vehicle assignment scheme. Given k appearing requests from v to u at time t and $[\boldsymbol{y}_i, 1 \le i \le N]$ satisfying the conditions in (4)-(6), we define

$$q_i = \min\left\{y^t_{i,v,u,k}/p^t_{v,u,k}, z^t_{i,v,u}\right\}.$$

Let $a_0 = 0$ and $a_i = (a_{i-1} + q_i) - 1_{a_{i-1}+q_i>1}$, $1 \le i \le N$. We associate each vehicle i with an interval $S_i \subset [0, 1]$ as follows:

$$S_i = \begin{cases} [a_i, a_{i-1}), & a_i > a_{i-1}; \\ \emptyset, & a_i = a_{i-1} \text{ and } q_i = 0; \\ [0, 1], & a_i = a_{i-1} \text{ and } q_i = 1; \\ [a_{i-1}, 1] \cup [0, a_i), & a_i < a_{i-1}. \end{cases}$$

It's straightforward to check that the length of interval S_i is q_i. Then, we generate a number η in $[0, 1]$ uniformly at random. We assign one of the k requests to vehicle i if $\eta \in S_i$. Since $|S_i| = q_i$, the probability that vehicle i is assigned a request is exactly q_i. Also for any value of η, at most k vehicles are assigned requests. An illustrating example is shown in Fig. 2.

3.3 Problem Formulation

Suppose vehicle i transports passenger following the path

$$r_i = \left(s_i, v_{i,1}, v_{i,2}, \cdots\cdots, v_{i,n_i}, d_i\right) \in \mathcal{R}_i,$$

where $\mathcal{R}_i$ is the set of all the routes from s_i to d_i with travel time smaller than $\Delta^{\max}_{s_i,d_i}$ and n_i is the number of intermediate regions in route r_i. Given the probabilities of getting request assignment, i.e.,

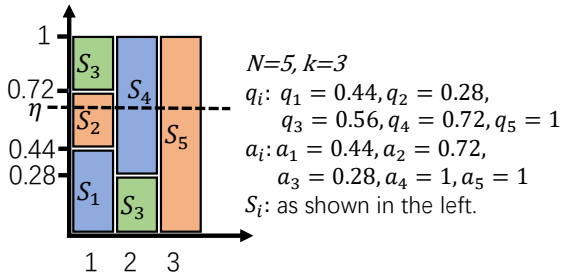


Figure 2: In this particular instance where 3 requests appear in a region of 5 vehicles, the 3 requests are assigned to vehicle 2, 4, and 5.

$\mathbf{y_i}$, the expected number of passengers that vehicle i will pick up (and starts a feasible ride-sharing trip) is given by

$$\begin{aligned} f_i(\mathbf{y_i}, r_i) \triangleq & 1 - \prod_{j=1}^{n_i} \left(1 - y_{i,v_{i,j}}^{t_{i,j}}\right) \\ = & 1 - \prod_{j=1}^{n_i} \prod_{u \in \mathcal{V}} \left(1 - \sum_{k=1}^{K} y_{i,v_{i,j},u,k}^{t_{i,j}}\right), \end{aligned} \tag{7}$$

where $\left(1 - y_{i,v_{i,j}}^{t_{i,j}}\right)$ is the probability that the vehicle i will not pick up a passenger when passing region $v_{i,j}$ at time $t_{i,j}$ and hence $f_i(\mathbf{y_i}, r_i)$ is the probability that vehicle i will pick up a passenger along the route r_i. Since all vehicles have already picked up a passenger when taking their routes r_i, each vehicle has only one seat and can at most pick up one more passenger. As such, $f_i(\mathbf{y_i}, r_i)$ is also the expected number of passengers that vehicle i will pick up along the route r_i. We note that $f_i(\mathbf{y_i}, r_i)$ is non-convex in $\mathbf{y_i}$ and it involves combinatorial routing decisions r_i's.

We formulate the joint (probabilistic) request-vehicle assignment and vehicle routing problem as follows:

$$\begin{aligned} \mathbf{MP}: \quad \max \quad & \sum_{i=1}^{N} f_i(\mathbf{y_i}, r_i) \\ \text{var.} \quad & [\boldsymbol{y}_i, 1 \le i \le N] \in \mathcal{Y},\ r_i \in \mathcal{R}_i,\ 1 \le i \le N, \end{aligned} \tag{8}$$

where $\mathcal{Y}$ is the feasible set of $\boldsymbol{y}_i$'s defined in Proposition 2. The following proposition shows that Problem **MP** is NP-hard.

Proposition 3. *The problem **MP** is NP-hard as it covers the NP-hard single-vehicle demand-aware routing problem in [22] as a special case.*

4 A DUAL-SUBGRADIENT ALGORITHM

In this section, we first introduce a time-expanded graph and then study a linear-combinatorial problem, by leveraging on an approximation for $f_i(\mathbf{y_i}, r_i)$ and an elegant reformulation. We will then explore the insights from studying the dual of the linear-combinatorial problem to derive a dual-subgradient algorithm for the original problem **MP**.

4.1 Reformulation over a Time-Expanded Graph

To facilitate the discussions and problem re-formulation for algorithm design, we first construct a time-expanded graph as follows.

Definition 4. *Given $\tau = \max_{1 \le i \le N} \{\Delta_{s_i,d_i}^{\max}\}$ and the regional graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the time-expanded graph $\mathcal{G}^{[\tau]} = \left(\mathcal{V}^{[\tau]}, \mathcal{E}^{[\tau]}\right)$ contains*

- *$(1+\tau) \times |\mathcal{V}|$ nodes, labeled as v^t where $v \in \mathcal{V}$ and $t \in [0, \tau]$.*
- *N virtual destination nodes, labeled as d_i^{-i}, when $1 \le i \le N$ and d_i is the destination of the current rider at vehicle i.*

The edge set $\mathcal{E}^{[\tau]}$ is constructed as follows:

- *For each edge $e = (u, v) \in \mathcal{E}$ with travel delay $\delta_{u,v}$, for each $t \in [1, \tau - \delta_{u,v}]$, create an edge $e^t \in \mathcal{E}^{[\tau]}$ from u^t to $v^{t+\delta_{u,v}}$. Note that there is no travel delay associated with e^t*
- *for each vehicle i, for each $t \in [1 : \Delta_{s_i,d_i}^{\max}]$, create an edge from d_i^t to d_i^{-i}.*

For any node $v^t \in \mathcal{V}^{[\tau]}$, it is associated with the probabilities $\left[p_{v,u,k}^t, \forall u \in \mathcal{V}, 1 \le k \le K\right]$. For ease of expression, in the following ,we use $\bar{v}$ to represent the nodes in $\mathcal{V}^{[\tau]}$; note that each $\bar{v}$ is one-to-one map to a v^t is associated with the time t implicitly. Similarly, we use $y_{i,\bar{v},u,k}$ to represent $y_{i,v,u,k}^t$ and $y_{i,\bar{v}}$ to represent $y_{i,v}^t$. We also use r_i and $f_i(\cdot,\cdot)$ to represent the route of vehicle i and its probability of picking up along route r_i, over the time-expanded graph.

Let $\mathcal{R}_i^{[\tau]}$ be the set of all the routes from s_i^0 to d_i^{-i} on the time expanded graph, we reformulate the problem **MP** over the time-expanded graph as follows:

$$\begin{aligned} \mathbf{MP\text{-}T}: \quad \max \quad & \sum_{i=1}^{N} f_i(\mathbf{y_i}, r_i) \\ \text{var.} \quad & [\boldsymbol{y}_i, 1 \le i \le N] \in \mathcal{Y},\ r_i \in \mathcal{R}_i^{[\tau]},\ 1 \le i \le N, \end{aligned} \tag{9}$$

By introducing the time-expanded graph, we omit the delay constraint on each route r_i as $R_i^{[\tau]}$ is all the simple path (i.e., involving no loop) from s^0 to d_i^{-i} in the time-expanded graph. However, we note that the network size of the time-expanded graph become τ times of the original graph. In the case that all the probabilities are time-invariant, this actually leads to exponential increment of the input size as it is polynomial in τ which is exponential in the bit length of the input τ. Consequently, any polynomial time algorithm on the time-expanded graph becomes a pseudo-polynomial time algorithm to the original problem **MP**.

4.2 A $(1 - 1/e)$ Approximation

We first note that $f_i(\boldsymbol{y}_i, r_i)$ in (9) can be upper-bounded and lower-bounded by two simple concave functions.

Proposition 5. *Define a concave function*

$$g_i(\mathbf{y_i}, r_i) \triangleq \min\left(1, \sum_{j=1}^{n_i} \sum_{u\in\mathcal{V}} \sum_{k=1}^{K} y_{i,\bar{v}_{i,j},u,k}\right).$$

Then

$$(1-1/e)\, g_i(\mathbf{y_i}, r_i) \le f_i(\boldsymbol{y}_i, r_i) \le g_i(\mathbf{y_i}, r_i).$$

The upper bound is obtained by the standard union-bound argument and the lower bound is according to [17].

With the above understanding, we formulate a problem **MP-A** as follows:

$$\textbf{MP-A}: \quad \max \quad \sum_{i=1}^{N} g_i(\mathbf{y_i}, r_i) \tag{10}$$
$$\text{var.} \quad [\boldsymbol{y}_i, 1 \le i \le N] \in \mathcal{Y},\ r_i \in \mathcal{R}_i^{[\tau]},\ 1 \le i \le N.$$

It has the same feasible region as **MP-T** but the objective function is replaced by a concave one. The following theorem says that interestingly, solving **MP-A** gives an approximation solution to **MP-T**.

Theorem 6. *Let $\left(\boldsymbol{y}_i^*, r_i^*\right)$, $1 \le i \le N$, be an optimal solution to **MP-T**, and let $(\bar{\boldsymbol{y}}_i, \bar{r}_i)$, $1 \le i \le N$, be that of **MP-A**. Then*

$$(1-1/e)\sum_{i=1}^{N} f_i\left(\boldsymbol{y}_i^*, r_i^*\right) \le \sum_{i=1}^{N} f_i(\bar{\boldsymbol{y}}_i, \bar{r}_i) \le \sum_{i=1}^{N} f_i\left(\boldsymbol{y}_i^*, r_i^*\right).$$

Proof. By the definition of $\left(\boldsymbol{y}_i^*, r_i^*\right)$, we have $\sum_{i=1}^N f_i(\bar{\boldsymbol{y}}_i, \bar{r}_i) \le \sum_{i=1}^N f_i\left(\boldsymbol{y}_i^*, r_i^*\right)$. We also have

$$\begin{aligned}(1-1/e)\sum_{i=1}^{N} f_i\left(\boldsymbol{y}_i^*, r_i^*\right) &\le (1-1/e)\sum_{i=1}^{N} g_i\left(\boldsymbol{y}_i^*, r_i^*\right)\\ &\le (1-1/e)\sum_{i=1}^{N} g_i(\bar{\boldsymbol{y}}_i, \bar{r}_i)\\ &\le \sum_{i=1}^{N} f_i(\bar{\boldsymbol{y}}_i, \bar{r}_i),\end{aligned}$$

where the first and third steps utilize Proposition 5, and the second step uses the definition of $(\bar{\boldsymbol{y}}_i, \bar{r}_i)$. □

An important insight from Theorem 6 is that the optimal solution of **MP-A** is a $(1-1/e)$ approximation solution of **MP-T**.

4.3 A Linear-Combinatorial Reformulation

We present an *equivalent* formulation of **MP-A** to facilitate the algorithm design discussion later. To proceed, we first introduce a set of routing variables $\mathbf{x}_i = [x_{i,\bar{v}}]_{\bar{v}\in\mathcal{V}^{[\tau]}}$ to indicate whether vehicle i pass node $\bar{v}$ on the time-expanded regional graph $\mathcal{R}_i^{[\tau]}$:

$$x_{i,\bar{v}} = \begin{cases} 1, & \text{if the route } r_i \text{ passes through } \bar{v};\\ 0, & \text{otherwise.}\end{cases} \tag{11}$$

Define $\mathcal{X}_i$ as the set of all valid $\mathbf{x}_i$ that corresponds to a $r_i \in \mathcal{R}_i^{[\tau]}$. It should be clear that the time-expanded graph is acyclic and directed, and any valid $\mathbf{x}_i \in \mathcal{X}_i$ maps to a valid $r_i \in \mathcal{R}_i^{[\tau]}$ and vice verse.

Next, we observe that for $[\mathbf{y}_i, 1 \le i \le N] \in \mathcal{Y}$, we can replace $\mathcal{Y}$ by $\bar{\mathcal{Y}}$, which is the set of $[\mathbf{y}_i, 1 \le i \le N]$ that satisfies

$$\sum_{i=1}^{N} x_{i,\bar{v}} \cdot y_{i,\bar{v},u,k} \le k \cdot p_{\bar{v},u,k},, \forall \bar{v}, u, k \tag{12}$$
$$0 \le y_{i,\bar{v},u,k} \le p_{\bar{v},u,k} \cdot z_{i,\bar{v},u}, \forall i, \bar{v}, u, k \tag{13}$$

where $z_{i,\bar{v},u}$ is defined in Sec. 3.1 and (13) is equivalent to (5) and (4). As compared to (6), we only count the probability allocation to vehicles that pass node $\bar{v}$ in (12). Although $\bar{\mathcal{Y}}$ is larger than $\tilde{\mathcal{Y}}$, we can easily see that any feasible solution in $\bar{\mathcal{Y}}$ can map to a feasible solution in $\mathcal{Y}$. Hence, we replace the constraints $[\mathbf{y}_i, 1 \le i \le N] \in \mathcal{Y}$ by $[\mathbf{y}_i, 1 \le i \le N] \in \bar{\mathcal{Y}}$.

Finally, we replace (13) with $0 \le x_{i,\bar{v}} \cdot y_{i,\bar{v},u,k} \le x_{i,\bar{v}} \cdot p_{\bar{v},u,k} \cdot z_{i,\bar{v},u}$, introduce new variables $y'_{i,\bar{v},u,k} = x_{i.\bar{v}} \cdot y_{i,\bar{v},u,k}$, and reformulate the problem **MP-A** as follows:

$$\textbf{MP-AN}: \max \quad \sum_{i=1}^{N} \sum_{\bar{v}\in\mathcal{V}^{[\tau]}} \left(\sum_{u\in\mathcal{V}} \sum_{k=1}^{K} y'_{i,\bar{v},u,k}\right) \tag{14}$$
$$s.t. \quad \sum_{\bar{v}\in\mathcal{V}^{[\tau]}} \left(\sum_{u\in\mathcal{V}} \sum_{k=1}^{K} y'_{i,\bar{v},u,k}\right) \le 1, \forall 1 \le i \le N, \tag{15}$$
$$\sum_{i=1}^{N} y'_{i,\bar{v},u,k} \le k \cdot p_{\bar{v},u,k}, \forall \bar{v}, u, k, \tag{16}$$
$$0 \le y'_{i,\bar{v},u,k} \le x_{i,\bar{v}} \cdot p_{\bar{v},u,k} \cdot z_{i,\bar{v},u}, \tag{17}$$
$$\text{var.} \quad y'_{i,\bar{v},u,k} \in [0,1], \mathbf{x}_i \in \mathcal{X}_i,$$
$$1 \le i \le N, 1 \le k \le K, v \in \mathcal{V}^{[\tau]}, u \in \mathcal{V}.$$

Problem **MP-AN** is a linear-combinatorial one. The following corollary says that solving **MP-AN** also gives an $(1-1/e)$ approximation solution to **MP-T**.

Corollary 7. *Let $\left(\boldsymbol{y}_i^*, r_i^*\right)$, $1 \le i \le N$, be an optimal solution to **MP-T**, and let $(\bar{\mathbf{y}}_i, \bar{\mathbf{x}}_i)$, $1 \le i \le N$, be that of **MP-AN**. Then*

$$(1-1/e)\sum_{i=1}^{N} f_i\left(\boldsymbol{y}_i^*, r_i^*\right) \le \sum_{i=1}^{N} f_i(\bar{\mathbf{y}}_i, \bar{r}_i) \le \sum_{i=1}^{N} f_i\left(\boldsymbol{y}_i^*, r_i^*\right),$$

where $\bar{r}_i$ is the route for vehicle i corresponding to $\bar{\mathbf{x}}_i$.

Remark: For problem **MP-AN**, one may think that we can use flow balance equations with unit flow from s_i^0 to d_i^{-i} on $G^{[\tau]}$ and require the flow to take integer value to represent $\mathbf{x}_i \in \mathcal{X}_i$; relax the binary variables to $[0,1]$ and solve it as an LP. However, such relaxation in general incurs optimality loss, as we construct a counterexample with a positive integrity gap in our technical report [23].

4.4 Dual Sub-Gradient Decent Algorithm

We now present our dual sub-gradient algorithm for solving problem **MP-AN**. Relaxing the the right-hand-side constrains in (17) by

introducing Lagrangian dual variables

$$\lambda \triangleq [\lambda_{i,\bar{v},u,k}]_{1\le i\le N,\bar{v}\in\mathcal{V}^{[\tau]},u\in\mathcal{V},1\le k\le K} \ge 0,$$

we obtain the following Lagrangian function,

$$L(\mathbf{x},\mathbf{y},\lambda) = \sum_{i=1}^{N}\sum_{\bar{v}\in\mathcal{V}^{[\tau]}}\sum_{u\in\mathcal{V}}\sum_{k=1}^{K}\left(1-\lambda_{i,\bar{v},u,k}\right)y'_{i,\bar{v},u,k} + \sum_{i=1}^{N}\sum_{\bar{v}\in\mathcal{V}^{[\tau]}} x_{i,\bar{v}}\sum_{u\in\mathcal{V}}\sum_{k=1}^{K}(\lambda_{i,\bar{v},u,k}\times p_{\bar{v},u,k}\times z_{i,\bar{v},u}).$$

The dual function is then

$$\begin{aligned} D(\lambda) = \quad & \max \quad L(x,y,\lambda) \\ & s.t. \quad (15),(16) \\ & \text{var.} \quad y'_{i,\bar{v},u,k}\in[0,1], \mathbf{x}_i\in\mathcal{X}_i, \\ & \qquad 1\le i\le N, 1\le k\le K, v\in\mathcal{V}^{[\tau]}, u\in\mathcal{V}. \end{aligned}$$

As the variables $\mathbf{x}$ and $\mathbf{y}$ are decoupled in $D(\lambda)$, we can decompose $D(\lambda)$ into two optimization problems as follows:

$$\begin{aligned} \textbf{D1}: \quad & \max \quad \sum_{i=1}^{N}\sum_{\bar{v}\in\mathcal{V}^{[\tau]}}\sum_{u\in\mathcal{V}}\sum_{k}^{K}\left(1-\lambda_{i,\bar{v},u,k}\right)y'_{i,\bar{v},u,k} \\ & s.t. \quad (15),(16) \\ & \text{var.} \quad y'_{i,\bar{v},u,k}\ge 0, 1\le i\le N, 1\le k\le K, v\in\mathcal{V}^{[\tau]}, u\in\mathcal{V}, \end{aligned}$$

and

$$\begin{aligned} \textbf{D2}: \quad & \max \quad \sum_{i=1}^{N}\sum_{\bar{v}\in\mathcal{V}^{[\tau]}} x_{i,\bar{v}}\sum_{u\in\mathcal{V}}\sum_{k}^{K}(\lambda_{i,\bar{v},u,k}\times p_{\bar{v},u,k}\times z_{i,\bar{v},u}) \\ & \text{var.} \quad \mathbf{x}_i\in\mathcal{X}_i, 1\le i\le N. \end{aligned}$$

Note that the problem **D1** is simply an LP and can be solved with a complexity polynomial in the size of the regional network and maximum travel time deadline τ. The problem **D2** is a longest path problem on an acyclic time-expanded regional graph, and it can be solved with a complexity polynomial in the size of the regional network and maximum travel time deadline τ [21].

To this end, we arrive at an iterative dual-subgradient algorithm as follows: in each iteration,

- given a set of $\lambda_{i,\bar{v},u,k}$, we solve the problems **D1** and **D2** with a complexity polynomial in the size of the regional network and maximum travel time deadline τ;
- we update the dual variables using a sub-gradient update: for $1\le i\le N, \bar{v}\in\mathcal{V}^{[\tau]}, u\in\mathcal{V}, 1\le k\le K$,

$$\lambda_{i,\bar{v},u,k} \leftarrow \lambda_{i,\bar{v},u,k} + \phi(\lambda)\left(y'_{i,\bar{v},u,k} - x_{i,\bar{v}}\times p_{\bar{v},u,k}\times z_{i,\bar{v},u}\right),$$

  where $\phi(\lambda)$ is a diminishing step size suggested for subgradient algorithms [8]. The basic idea is choosing large initial stepsize and gradually decreasing the stepsize as the gap between the dual value and current recovered primal value gets smaller. And when we detect that the gap is smaller than a predefined threshold, we decrease the stepsize much faster. We leave the details to our technical report [23] .

The dual-subgradient algorithm is known to converge at a rate of $O(\frac{1}{\sqrt{K}})$, where K is the number of iteration. In our implementation, We terminate the iterations when either gap between the dual value and current recovered primal value gets smaller than a preset threshold or the number of iteration exceeds a preset limit.

Upon convergence, the dual-subgradient algorithm is known to generate an optimal dual solution. However, it is not guaranteed to generate an optimal solution for the primal problem, since there could be duality gap for the linear-combinatorial problem studied in this section. In the following, we establish a condition under which our dual-subgradient algorithm also gives an optimal solution to the primal problem.

Theorem 8. *If upon termination of the dual-subgradient algorithm, the dual variables λ satisfy that*

$$[y'_{i,\bar{v},u,k} - x_{i,\bar{v}}\times p_{\bar{v},u,k}\times z_{i,\bar{v},u}]^{+}_{\lambda_{i,\bar{v},u,k}} = 0, \forall i,\bar{v},u,k, \quad (18)$$

where function $[f]^{+}_{g}$ is defined as

$$[f]^{+}_{g} = \begin{cases} f, & \text{if } g>0; \\ \max(f,0), & \text{otherwise.} \end{cases}$$

*Then each x^* and y^*, specifying routes for vehicles and request-vehicle assignments, is an optimal solution to **MP-AN** and hence a $(1-1/e)$ approximation solution to **MP-T**.*

The results are proved in our technical report [23] by utilizing an argument based on complementary slackness.

Remarks. We discuss how the dual-subgradient algorithm is implemented as follows. After attaining the solution $\mathbf{x}_i$ and $y'_{i,\bar{v},u,k}$, $\forall 1\le i\le N, 1\le k\le K, v\in\mathcal{V}^{[\tau]}, u\in\mathcal{V}$. For vehicle i, it travels along the route r_i to its destination. When vehicle i passes by region $v_{i,j}$ in the corresponding slot $t_{i,j}$, given the actual requests information u and k, we assign the request to the vehicles according to the request-vehicle assignment scheme in Sec. 3.2. If vehicle i is assigned a request, then it follows a feasible ride-sharing plan by solving the feasibility problem in Sec. 3.1 to deliver both passengers (and such plan exists since the vehicle is assigned a request); otherwise, it goes to next region along the route r_i.

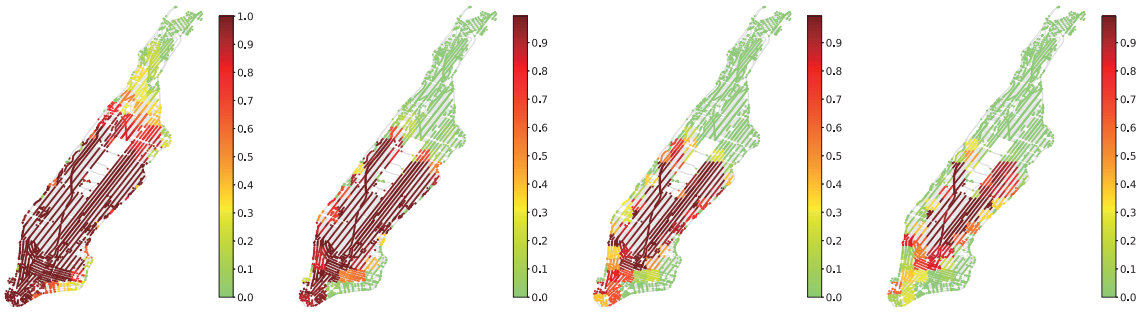
5 NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of our solution of joint request-vehicle assignment and vehicle routing for a fleet of vehicles, through extensive simulation using real-world traces. Our purposes are to evaluate the benefit of (i) demand-aware ride-sharing at the fleet level as compared to the demand-oblivious baseline and (ii) our new joint fleet-level optimization as compared to the previous separate optimization at individual vehicles. We use the number of fulfilled requests, terms as empirical pickups, as the performance metrics.

5.1 Evaluation Setup

5.1.1 Dataset and Region Graph. We use the taxi-trip dataset of Manhattan, New York City [1]. We extract around 60 million taxi-trip records in 6 months from the dataset (2016-01-01 to 2016-06-30), each including pickup time, pickup location, drop-off time, and drop-off location.

We obtain the Manhattan map from the OpenStreetMap [2]. We then use python package NetworkX [18] and OSMnx [12] to build



(a) $\mathbb{P}\left(\sum_{d\in\mathcal{V}} \Phi_{s,t,d} \geq 1\right)$. (b) $\mathbb{P}\left(\sum_{d\in\mathcal{V}} \Phi_{s,t,d} \geq 10\right)$. (c) $\mathbb{P}\left(\sum_{d\in\mathcal{V}} \Phi_{s,t,d} \geq 20\right)$. (d) $\mathbb{P}\left(\sum_{d\in\mathcal{V}} \Phi_{s,t,d} \geq 30\right)$.

Figure 3: Probability heatmaps calculated by using (19) for the time slot $t = 216$, corresponding to the time window $[18:00, 18:05]$ of the day. As seen, the distributions are highly heterogeneous across regions and many regions have limited demands. As such, it is important to perform a demand-aware, instead of demand-oblivious, design and to jointly assign requests to vehicles, in order to achieve proper demand-supply balancing and avoid passenger pickup conflict.

a road network $\mathcal{G}_0$. Then we divide the whole area of 59.5km^2 into 147 rectangular regions, each of length 700 meters and width 600 meters. We then construct the region-based graph $\mathcal{G}$ according to the procedure described in Sec. 2.1.

We compute the distance $d_{u,v}$ (km) for edge (u,v) in the road network $\mathcal{G}_0$. We assume that the taxis travel at the speed of 15km/h, according to a mobility report from NYC Department of Transportation [3]. The travel time of edge (u,v) is estimated as

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

5.1.2 Empirical distribution of travel requests. We use the trip data to obtain the empirical distribution of the travel requests. We first set the time slot length to be 5 minutes and divide the 24 hours in a day into 288 slots. The number of requests from node s to node d at time slot $t \in [1, 288]$ in a day is modeled by a random variable $\Phi_{s,t,d}$. We use the taxi-trip data in the 6-month period, 182 days in total, to compute the empirical distribution as follows: for all s, d in $\mathcal{G}$ and $t \in [1, 288]$,

$$\mathbb{P}\left(\Phi_{s,t,d} = k\right) = \frac{1}{182} \left(\#\text{ days with } k\ s-d \text{ requests in } t\right). \quad (19)$$

We then plot the demand heat map to visualize the request distribution, i.e., $\mathbb{P}\left(\sum_{d\in V} \Phi_{s,t,d} \geq k\right)$, in Fig. 3.

5.1.3 Simulation Environment. We use a server cluster with 34 i7-3770/3.40GHz CPUs and 7 E5-2623/v3/3.00GHz CPUs for simulation. Each machine has on average a memory size of 17GB and has installed Red Hat as its operating system. The computing resource allows us to carry out real-world trace driven simulations for several hundreds of vehicles in the demand-crowded lower Manhattan area, consisting of ten 1.25 km x 1.25 km regions. We use python to implement all the comparing algorithms. We use python package Matplotlib to generate our figures.

5.1.4 Simulation Instance. We use $\left(\vec{s}, \vec{d}, t_s\right)$ to denote a problem instance, where $\vec{s}$ is the vector of sources of the N vehicles in the fleet, $\vec{d}$ is the vector of destinations of the vehicles, and t_s is the pickup time of the first passenger. In our simulation, we choose the first pickup time slot $t_s = t_i := i \times \frac{60}{5}$, where $i = 0, \ldots, 23$, in the i−th hour of one day. For every $t_s = t_i$, we sample 10 source and destination pairs with shortest path longer than 3 from the real world pickup traces in the particular hour t_i in each of the 100 days that we run simulations upon. Once we have all the 10 source destination pairs and the corresponding first-passenger pickup time t_s, we put them together to form one instance $(\vec{s}, \vec{d}, t_s)$. We generate in total more than $24 \times 10 = 240$ instances for simulation.

5.1.5 Schemes for Comparison and Performance Metric. For each instance, we implement the following three algorithms and compare their performance:

- Joint routing: our demand-aware joint request-vehicle assignment and vehicle routing algorithm proposed in Sec. 4.
- Independent routing: our previous demand-aware routing algorithm for single vehicle only [22] and an intuitive uniform request-vehicle assignment scheme for assigning multiple appearing requests to multiple vehicles in the same region.
- Fastest routing: a demand-oblivious fastest routing algorithm and a uniform request-vehicle assignment scheme.

Note that here the uniform request-vehicle assignment scheme means that if there are more than one vehicle that can pick up a request in a region, we assign the request to any of the vehicles uniformly at random.

We evaluate an important performance metric *empirical pickup*, namely the average total pickups of all the vehicles over 100 days. Intuitively, the empirical pickup represents the empirical *service throughput* of the fleet in a region with certain demand distribution. Note that for each instance $(\vec{s}, \vec{d}, t_s)$, we evaluate the performance of a scheme by its average pickup across 100 days.

5.2 Benefits of Demand-Aware Optimization

We evaluate the number of empirical pickups in different hours of a day of the three algorithms described in Sec. 5.1.5. In this evaluation, we set the number of vehicles to be N=50 and the delay tolerance

factor to be $\alpha = 1.3$ as defined in Sec. 2.2 (used in our joint routing scheme and the independent routing scheme).

We recall that a request can be picked up by the fleet of N vehicles if and only if there is at least one vehicle that (i) is in the same 5-min region as the request and (ii) is assigned by the request-vehicle assignment module to pick up the request. If a request appears in a region and is not assigned to a vehicle, then the request cannot be fulfilled due to the maximum waiting time constraints.

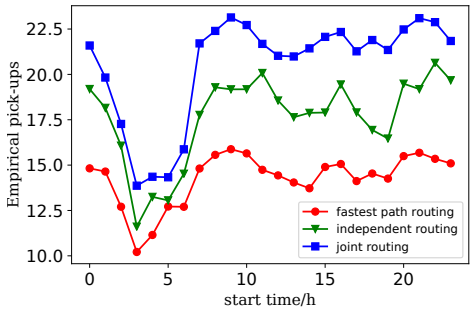


Figure 4: The empirical pickups under the setting of delay factor $\alpha = 1.3$ and fleet size $N = 50$.

The evaluation results are shown in Fig. 4. As seen, our proposed joint request-vehicle assignment and routing algorithm fulfills significantly more requests than the independent routing algorithm and the fastest path routing algorithm throughout the day, especially during the peak hours at noon and evening (in Manhattan). In particular, the daily-average improvement of our demand-aware solution as compared to the demand-oblivious fastest path routing is 46%. This shows that exploiting demand statistics can significantly improve the service throughput of the fleet. Furthermore, the daily-average improvement of our joint assignment and routing solution as compared to the independent routing solution is 19%. This implies that joint optimization at the fleet-level can bring significant service throughput improvement as compared to the selfish optimization at the level of individual vehicles.

5.3 Impacts of the Fleet Size

We fix the delay tolerance factor to be $\alpha = 1.3$ and the time slot to be $t_s = 204$, corresponding to the time window $[17:00, 17:05]$ of the day (when the request statistics is representative). We evaluate how the performance of the three schemes vary as a function of the fleet size N. The results are reported in Fig. 5.

Ideally, for the demand-rich Manhattan area, one would expect the number of pickups should increase as the fleet size N increases. This is indeed the case for all the three algorithms when N is less than 150, as seen from Fig. 5.

Meanwhile, as the fleet size increases beyond 150 in our simulation, the number of pickups of the independent routing and fastest path routing saturate. In comparison, that of our joint routing solution still observes improvement. These two observations highlight two important insights. First, the saturation in service throughput improvements seen by the independent routing and the fastest path routing are not due to insufficient requests in the region. Rather, it

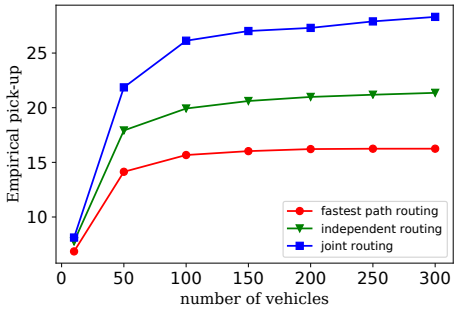


Figure 5: Empirical pickups for $\alpha = 1.3$ and the time slot $t_s = 204$, corresponding to the time window $[17:00, 17:05]$ of the day.

is because neither of them considers load-balancing across regions, which leads to inefficient routing decisions that result in excessive request-vehicle assignment conflicts in some regions while insufficient vehicles for serving requests in other regions. In contrast, our joint assignment and routing solution properly load-balances the fleets (with request-fulfillment deadlines taken into consideration) across regions, allowing the service throughput of the fleet to increase as the fleet size increases.

Second, the results suggest that it is more important to perform intelligent fleet-level optimization for large fleets. This is also intuitive. When the fleet size is small, the limiting factor of the service throughput is the (small) number of vehicles. In contrast, when the fleet size is large, the limiting factor is no longer the number of vehicles, but the routing and assignment efficiency. This explains the particularly superior performance of our joint assignment and routing solution when the fleet size is large. Of course, when the fleet size further increases, one would expect that the limiting factor would change again to be the demand richness in the area, approaching the "service capacity" achievable by any fleets with optimal routing and assignment efficiency.

Overall, this set of results suggest that it is important to perform fleet-level joint optimization to fully release the potential of demand-aware ride-sharing, in particular for large fleets.

5.4 Impacts of the Delay Tolerance Factor

To study the impacts of delay tolerance factor α, we fix $t_s = 204$, corresponding to the time window $[17:00, 17:05]$ of the day. We plot the empirical pickups as a function of the delay factor in Fig. 6. Again, our proposed joint assignment and routing algorithm outperforms the other two significantly. We observe that when the delay factor α increases, the empirical pickups also increase, which is intuitive as the ride-sharing optimization space increases as α increases. We also observe that the pickups of the demand-oblivious fastest path routing algorithm increases slower than the demand-aware solutions.

Furthermore, we observe that the performance gap between our proposed joint assignment and routing algorithm and the independent routing algorithm increase as the delay factor increases. An

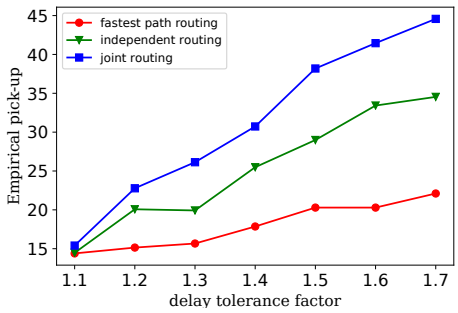


Figure 6: Empirical pickups for $N = 100$ and the time slot $t_s = 204$, corresponding to the time window $[17:00, 17:05]$ of the day.

intuitive explanation is that as the delay factor increases, the ride-sharing optimization spaces for individual vehicles increase. Thus it becomes more likely that different vehicles decide to route through the same demand-rich regions, causing more pickup conflicts and serious service imbalance across regions. On the contrary, our joint assignment and routing approach properly load-balances across the regions and thus maximizes the empirical pickups.

6 CONCLUDING REMARKS

As the first step to explore the demand-aware design, this paper focuses on the snapshot version of the ride-sharing problem. While the problem is already NP-hard, we derive a practical pseudo polynomial-time algorithm that achieves an approximation ratio of $(1 - 1/e)$ under the conditions presented in Theorem 8.

We make the following remarks. First, in our joint routing and request-vehicle assignment optimization at the present time, i.e., $t = 0$, the statistical future demand information at $t = 1, 2\ldots$ is already taken into account. Thus our approach is a demand-aware one for the snapshot version of the ride-sharing problem. Second, upon change in vehicle status, e.g., a user is delivered to the destination or an empty car picks up a new user, the system can re-optimize the routing decisions and request-vehicle assignments, so as to optimize the long-term ride-sharing performance in a greedy fashion. The overall solution can serve as a baseline for other demand-aware studies, e.g., the conceivable ones by extending the approach in [13] and [27] to the multi-rider setting and the recent one in [5]. We leave the performance analysis of the overall greedy solution, as well as developing solutions with optimized long-term performance, as interesting and important future directions.

ACKNOWLEDGEMENT

Xiaojun Lin would like to thank the support by NSF Grant ECCS-1509536.

REFERENCES

[1] TLC Trip Record Data. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.
[2] OpenStreetMap. https://www.openstreetmap.org.
[3] New York City Mobility Report. http://www.nyc.gov/html/dot/downloads/pdf/mobility-report-2016-screen-optimized.pdf.
[4] Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research 223*, 2 (2012), 295–303.
[5] Alabbasi, A., Ghosh, A., and Aggarwal, V. Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *arXiv preprint arXiv:1903.03882* (2019).
[6] Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences 114*, 3 (2017), 462–467.
[7] Alonso-Mora, J., Wallar, A., and Rus, D. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *Proc. IEEE/RSJ IROS* (2017), pp. 3583–3590.
[8] Bazaraa, M. S., and Sherali, H. D. On the choice of step size in subgradient optimization. *European Journal of Operational Research 7*, 4 (1981), 380–388.
[9] Bei, X., and Zhang, S. Algorithms for trip-vehicle assignment in ride-sharing. In *Proc. AAAI* (2018).
[10] Biswas, A., Gopalakrishnan, R., Tulabandhula, T., Mukherjee, K., Metrewar, A., and Thangaraj, R. S. Profit optimization in commercial ridesharing. In *Proc. ACM AAMAS* (2017), pp. 1481–1483.
[11] Blaszczyszyn, B., and Giovanidis, A. Optimal geographic caching in cellular networks. In *Proceedings of IEEE International Conference on Communication* (2015), pp. 3358–3363.
[12] Boeing, G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems 65* (2017), 126–139.
[13] Braverman, A., Dai, J., Liu, X., and Ying, L. Fluid-model-based car routing for modern ridesharing systems. In *Proc. ACM SIGMETRICS* (2017), pp. 11–12.
[14] Braverman, A., Dai, J. G., Liu, X., and Ying, L. Empty-car routing in ridesharing systems. *arXiv preprint arXiv:1609.07219* (2016).
[15] Clewlow, R. R., and Mishra, G. S. Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the united states. *University of California, Davis, Institute of Transportation Studies, Davis, CA, Research Report UCD-ITS-RR-17-07* (2017).
[16] Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological 57* (2013), 28–46.
[17] Goemans, M. X., and Williamson, D. P. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM Journal on Discrete Mathematics 7*, 4 (1994), 656–666.
[18] Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In *Proc. SciPy* (2008), pp. 11–15.
[19] Ioannidis, S., Yeh, E., Yeh, E., and Ioannidis, S. Adaptive caching networks with optimality guarantees. *IEEE/ACM Transactions on Networking (TON) 26*, 2 (2018), 737–750.
[20] Jia, Y., Xu, W., and Liu, X. An optimization framework for online ride-sharing markets. In *Proc. IEEE ICDCS* (2017), pp. 826–835.
[21] Korte, B., and Vygen, J. *Combinatorial Optimization: Theory and Algorithms*, 4th ed. Springer Publishing Company, Incorporated, 2010.
[22] Lin, Q., Deng, L., Sun, J., and Chen, M. Optimal demand-aware ride-sharing routing. In *Proc. IEEE INFOCOM* (2018), pp. 826–835.
[23] Lin, Q., Xu, W., Chen, M., and Lin, X. A probabilisitic approach for demand-aware ride-sharing optimization. *Techical Report, http://arxiv.org/abs/1905.00084* (2019).
[24] Ma, S., Zheng, Y., and Wolfson, O. T-share: A large-scale dynamic taxi ridesharing service. In *Proc. IEEE ICDE* (2013), pp. 410–421.
[25] MacKay, D. J. *Information theory, inference and learning algorithms.* Cambridge university press, 2003.
[26] Miller, J., and How, J. P. Predictive positioning and quality of service ridesharing for campus mobility on demand systems. In *Proc. IEEE ICRA* (2017), pp. 1402–1408.
[27] Oda, T., and Joe-Wong, C. Movi: A model-free approach to dynamic fleet management. In *Proceedings of IEEE INFOCOM* (2018), pp. 2708–2716.
[28] Partnership for New York City. $100 Billion Cost of Traffic Congestion in Metro New York. Internet: http://pfnyc.org/wp-content/uploads/2018/01/2018-01-Congestion-Pricing.pdf/, 2018.
[29] Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. A review of dynamic vehicle routing problems. *European Journal of Operational Research 225*, 1 (2013), 1–11.
[30] Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., and Ratti, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences 111*, 37 (2014), 13290–13294.
[31] Yuan, J., Zheng, Y., Zhang, L., Xie, X., and Sun, G. Where to find my next passenger. In *Proc. ACM UbiComp* (2011), pp. 109–118.
[32] Zhang, D., He, T., Liu, Y., Lin, S., and Stankovic, J. A. A carpooling recommendation system for taxicab services. *IEEE Transactions on Emerging Topics in Computing 2*, 3 (2014), 254–266.
[33] Zhang, D., Li, Y., Zhang, F., Lu, M., Liu, Y., and He, T. coRide: Carpool service with a win-win fare model for large-scale taxicab networks. In *Proc. ACM SenSys* (2013).