Re-Balancing Self-Interested Drivers in Ride-Sharing Networks to Improve Customer Wait-Time

Armin Sadeghi, *Student Member, IEEE*, Stephen L. Smith, *Senior Member, IEEE*

***Abstract*—In this paper, we address the problem of controlling self-interested drivers in ride-sharing applications. The objective of the ride-sharing company is to improve the customer experience by minimizing the wait-time before pick-up. Meanwhile, the drivers attempt to maximize their profit by choosing the best location to wait in the environment between the ride requests assigned to them. The objectives of the ride-sharing company and the drivers are not aligned, and the company has no direct control over the waiting locations of the drivers. The focus of this paper is to provide two indirect control methods for the ride-sharing company to optimize the set of waiting locations of the drivers, thereby minimize one of two objectives: 1) the expected wait-time of the customers, or 2) the maximum wait-time of customers. The proposed indirect control methods are 1) sharing information to a subset of the drivers on the location of other waiting drivers, and 2) paying drivers to relocate. We show that the problem of finding the optimal control is NP-hard for both objectives and both control methods. For the information sharing method, we provide an LP-rounding algorithm to minimize the expected wait-time and a 3-approximation algorithm to minimize the maximum wait-time. To incentivize the drivers to relocate with payments, we provide 3-approximation algorithms for both objectives. Finally, we evaluate the proposed control methods on real-world data and show that we can achieve up to 80% improvement for both objectives.**

***Index Terms*—Mobility-On-Demand Systems, Re-Balancing Self-Interested Drivers, Facility Location Problem**

I. Introduction

In recent years, ride-sharing services such as UberX and Lyft have emerged as an alternative mode of urban transportation. The compelling feature of these services compared to conventional taxi services is the improved service quality such as the expected wait time for pick-up [1]. The wait-time of a pick-up for a ride request is a function of the distribution of the drivers and the customers in the environment. Therefore, the objective of the ride-sharing company is to distribute the drivers in the environment to improve the service quality. However, the ride-sharing company does not have control over the positions of the drivers as they are self-interested units maximizing their local objectives. Therefore, the challenge is to ensure the service quality only using indirect controls on the positions of the drivers.

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (a6sadegh@uwaterloo.ca; stephen.smith@uwaterloo.ca)

Fig. 1. A set of drivers in the ride-sharing system and a set of locations with high probability of ride request arrival.

Surge pricing in high demand areas is an instance of a conventional indirect control method for both re-balancing the drivers and increasing the supply of drivers which is employed by Uber. Although this control method reduces the expected response time of servicing the requests by drawing more drivers to the high demand areas, it can draw drivers away from lower demand areas, resulting in higher wait times in those areas and more imbalance [2], [3].

The problem of servicing ride requests in ride-sharing networks consists of two major problems: 1) assignment of the current ride requests to the drivers; and 2) rebalancing of the drivers for the future ride requests. The main focus of this paper is the latter where we rebalance a subset of drivers to service ride requests arriving sequentially in an environment. The transportation network is represented by a graph and the ride requests arrive on the nodes of the graph according to a known arrival rate (see Figure 1). The drivers are self-interested units maximizing their expected profit by choosing their location in the environment to wait for the next ride request. Therefore, the objective of the ride-sharing company is to incentivize the drivers to relocate to a set of waiting locations that maximizes the service quality. We measure the service quality using two objectives: 1) the expected wait-time before pick-up, which provides better service in high demand regions; or 2) the maximum wait-time before pickup, which provides a more balanced service quality across different regions. Note that the relocation of drivers is effective when ride requests are less frequent relative to the number of active drivers (i.e., in light load), and thus there is time to relocate between consecutive rides.

Contributions: The contributions of this paper are four-fold. First, we formulate the ride-sharing problem with self-interested units and two global objectives; minimizing the expected wait time, and minimizing the maximum wait time of the ride request. We prove the NP-hardness of these problems. Second, we propose two indirect control methods to relocate the drivers in the environment: 1) sharing the location of all drivers with a subset of drivers, and 2) paying the drivers to relocate. Third, we develop novel algorithms for each objective and control method combination. For the problem of minimizing the expected wait-time under information sharing, we propose an LP-rounding algorithm which provides near-optimal solutions in an extensive set of experiments. We provide a 3-approximation algorithm for the problem of minimizing the maximum wait-time under information sharing. To find the optimal payment in the second control method, we first cast the problem as a game between the drivers and the service provider, where the service provider seeks to minimize a linear combination of the total amount paid and the global objective. We provide 3-approximation algorithms to find the optimal control for both global objectives. Fourth, we extend the proposed methods to capture uncertainty in drivers behaviour while maintaining the bounds on the solution quality. Finally, we evaluate the performance of the proposed control methods on real-world ride-sharing data for the two global objectives.

A preliminary version of this work appeared in [4]. In this paper, we extend the results in [4] to capture an additional global objective of minimizing the maximum wait-time and we propose two new algorithms to handle this objective. The objective of minimizing the expected wait-time can draw drivers to high demand areas, which results in unbalanced service quality across the environment. In contrast, the min-max objective promotes equal service quality across the entire environment. This type of objective could be particularly important if the ride-share system is offered as a public service, where, for example, a high quality of service is required in both urban and rural areas. We also extend the proposed control methods to capture uncertainty in drivers behaviour.

Related Work: An extensive number of studies consider the problem of optimally assigning taxis to the ride requests arriving sequentially over time [5]–[8]. In contrast, we focus on the problem of optimally assigning waiting locations to the drivers such that the expected wait-time or the maximum wait-time of the customers is minimized. For assignment of the ride requests to the drivers, we borrow the common method employed by the ride-sharing companies which is to assign the requests to the closest available drivers in a first-come-first serve fashion [9].

The problem of rebalancing service units in the environment has been studied for various applications. In the mobility-on-demand problem (MOD) [10]–[12], a group of vehicles are located at a set of stations. The customers arrive at the stations, hire vehicles for ride, and then drop the vehicles off at the station closest to their destination. The objective is to balance the vehicles at the stations to minimize the expected wait time of the customer. In comparison to MOD, we consider the customer wait-time as the time between the request arrival and the pick-up time, which incorporates the distance of the closest available vehicle to the pick-up location. In [13], the authors focus on the intersection of the MOD systems and the public transportation where the ride-sharing company, customers and the municipal transportation authority are self-interested units. In this study, the authors provide a pricing scheme for the ride-sharing company to maximize service quality. The MOD methods consider the macroscopic aspect of the ride-sharing problem where a flow formulation is provided to approximate the average number of vehicles to relocate from a station to others. In contrast to the MOD approaches, our paper captures the microscopic aspect of ride-sharing, focusing on the movement of individual drivers.

A conventional method for relocating the drivers in a ride-sharing network is by the surge pricing method in high-demand areas. The problem of pricing ride requests in a ride-sharing system has been studied recently in the literature [14]–[16]. In [15], the authors study the ride-sharing problem in a ride-sharing network where the service units are a combination of the self-interested drivers and autonomous vehicles. The authors propose a pricing scheme and a payment method for the self-interested drivers to rebalance in the network. However, increasing the price of rides in high-demand areas to incentivize the drivers to relocate to those areas decreases the demand [15]. In contrast, we propose an indirect control method for relocating the drivers by sharing information on the position of the drivers to a subset of them and steer them towards the areas with higher demand and lower supply.

The facility location problem [17], [18] and its extension to the mobile facility location problem (MFL) [19] is the problem of distributing facilities in a set of locations to respond to the demands arriving at different locations. The objective is to minimize the time to respond to the demands and the total cost of opening facilities. A special case of the facility location problem is the k-median problem [18] where the number of open facilities is limited and the cost of opening a facility is zero. In [20], we addressed a multi-stage MFL problem where we relocate a set of autonomous vehicles to minimize expected response time for future requests in a receding horizon manner. However, a key difference in MFL problems is that the objective of the service units are aligned with the service provider, and thus the waiting locations of service units can be directly controlled.

A closely related variant of the facility location problem is that of Voronoi games on graphs [21] where service units are self-interested. The objective of each self-interested service unit is to maximize the number vertices assigned to them. This work shows that the problem of finding the pure Nash equilibrium for the game between the service units on general graphs is NP-hard. In [22], the authors provide the best response strategy for each driver and they approximate Nash equilibria. These studies focus on the strategies of the self-interested service units. In contrast, we focus on finding the optimal policy for the ride-sharing company to optimally respond to the ride requests.

The paper is organized as follows. In Section II, we formulate the problem of minimizing the expected or the maximum wait-time of customers with self-interested drivers. In Section III, we propose the first indirect control method to relocate the drivers based on sharing the information on the drivers with a subset of them. In Section IV-A, we provide the second control method based on incentive pay. Finally in Section V, we evaluate the performance of the proposed control methods on an extensive set of experiments with real-world ride-sharing data.

II. Problem Formulation

In this section, we provide the detailed description of the environment model for the ride-sharing problem and the models for the self-interested drivers and the service provider.

A. Environment Model

Consider a complete metric graph $G = (V, E, c)$ where the vertex set V represents the pick-up and drop-off locations, the edge set E is the set of connections between the vertices and the function $c : E \to \mathbb{R}_+$ assigns a travel time to each edge on the graph. There are m drivers in the environment responding to the ride requests arriving over time. The drivers wait on a subset of the vertices for the next request, which we call the configuration of the drivers Q. The set of all configurations of the drivers is denoted by $\mathcal{Q}$.

The ride requests arrive at each vertex u of the graph according to an independent process. Upon a request arrival, the closest driver to the vertex of the request is assigned to service the request. Let $p_a(u)$ denote the arrival probability, which represents the likelihood of ride request arriving at vertex u. Let the drop-off probability $p_d(\text{dropoff} = w|\text{pickup} = v)$ be the probability of a request with pickup location v and a drop-off location w.

B. Self-interested Drivers' Model

We assume that the drivers in the system act in their self-interest to maximize their profit. A driver i might be aware of the position of a subset of other drivers which we call the information of driver i and denote by I_i. For instance, each driver may be aware of the location of the other drivers in its vicinity. We assume that the information $I_i = \emptyset$ for all drivers unless it is provided by the centralized service provider.

Each driver selects her next waiting location based on her perception of profit at different locations. Driver i's *perception of her expected profit* is a function of her current location q_i, the information I_i provided to her by the service provider on the location of other drivers, environment parameters such as arrival times, drop-off location probabilities and the period of working time B_i, denoted by $\mathcal{V}_i(u, B_i, I_i)$. Hence, the self-interested driver will wait at a location that maximizes its expected profit, i.e.,

$$\arg\max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B_i - c(q_i, u), I_i), \tag{1}$$

where σ is the cost per minute of driving. The drivers following this model are called *deterministic* drivers.

The remainder of this paper and the proposed main control methods do not rely on any specific form of function $\mathcal{V}_i$. We do assume, however, that the ride-sharing company has access to this function, obtained through data of driver behavior. In Appendix B we present one potential model of $\mathcal{V}$, which is then used for simulating the two control methods. In Sections III and IV, we provide a noisy driver model and evaluate the robustness of the proposed control methods to the uncertainty in the behaviour of drivers.

TABLE I
Summary of Algorithms Proposed

Control method	J_{exp}	$J_{\max}$
information sharing	LP-rounding	3-approx. algorithm
pay-to-control$*$	3-approx. algorithm	3-approx. algorithm

$*$For the pay-to-control method we minimize a linear combination of the total amount paid and the global objective.

C. Service Provider's Model

We consider a global objective $J(Q)$ to maximize the service quality. In this paper, we focus on two global objectives in servicing tasks: 1) the expected wait-time and 2) the maximum wait-time of the ride requests.

The expected wait-time objective can be expressed as

$$J_{\text{exp}}(Q) = \sum_{u \in V} \min_{q_i \in Q} p_a(u) c(q_i, u). \tag{2}$$

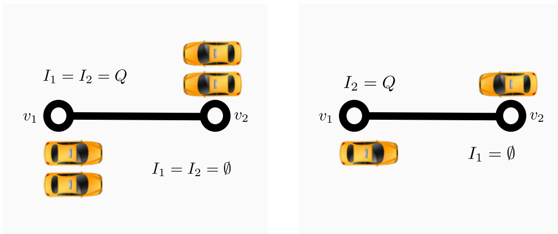
Under global objective (2), the desired configuration of the drivers concentrates on the regions with higher arrival rates.

An alternative global objective is to minimize the maximum wait-time over all ride requests, i.e.,

$$J_{\max}(Q) = \max_{u \in V} \min_{q_i \in Q} c(q_i, u). \tag{3}$$

Under this objective, the drivers provide a more uniform service quality at different locations regardless of the arrival rate at the locations.

The main challenge in optimizing these global objectives is that the drivers are self-interested units and the service provider does not have any direct control over the waiting locations of the drivers. The two indirect control methods proposed in this paper incentivize the drivers to relocate to desired waiting locations. The first control method exploits the dependency of the expected profit of the drivers on their information I_i. The service provider selects a subset of the drivers to share information on the location of drivers and manipulate their decision towards relocating to a desired waiting location. We refer to this as the *information sharing* control method. The second proposed control method, incentivizes the drivers to relocate to desired waiting locations with payments, which we refer to as the *pay-to-control* method. These control methods are applicable to various models of driver behavior $\mathcal{V}$. Table I summarizes the results provided on the proposed two control methods and the two global objectives. The proposed control methods are applied whenever the wait-time for the customers with current configuration of the drivers is surpassing the desired threshold of the service provider. In the event that a ride-request arrives while the drivers are relocating, the closest driver is assigned to service the ride-request. The drivers currently servicing a ride-request are not considered for relocation by the proposed control methods.



(a) Equal information sharing (b) Partial information sharing

Fig. 2. Ride-sharing problem with two vehicles and two request arrival locations.

In the following sections, we provide a detailed description of the two control methods and propose algorithms to find near-optimal controls.

III. Control by Sharing Information

The method proposed in this section exploits the fact that the decision of the self-interested drivers on their optimal waiting location (see Equation (1)) is a function of the information provided them on the position of the other drivers, i.e., I_i.

First we provide an example to demonstrate the importance of the information of the drivers in controlling their configuration in the environment.

Example III.1: Consider a ride-sharing system with two ride request locations, unit distance apart, and two vehicles (see Figure 2). The vehicles are initially located at v_1 and will relocate to the best waiting location, namely optimizing Equation (1). Let the expected profits at both locations are sufficiently larger that σ, i.e., $\sigma \ll \mathcal{V}(v_i, B_i, I_i)$ for $i \in 1, 2$ and any information provided. Figure 2(a) shows the two scenarios where both vehicles are provided the same information, i.e., $I_1 = I_2 = \emptyset$ and $I_1 = I_2 = Q$. In the first scenario, no information is provided to both drivers, therefore, the drivers uninformed about the other driver, will not incur the cost of relocation as they consider that the rides in both locations will be assigned to them. In the second scenario, both drivers are given the information about the position of the other driver. Therefore the expected profit of the drivers will decrease if they wait at v_1, and they will both relocate to v_2 to make sure that the rides at v_2 will be assigned to them. Note that the configuration of the vehicles when they are provided the same information is the worst possible configuration for the global objective J_{exp}. However, illustrated in Figure 2(b), providing the information to a subset of the vehicles results in the optimal configuration for the global objective. •

A. Formal Defintion and Complexity Class

The information sharing problem consists of 1) deciding the subset of drivers we share information with, and 2) deciding what information to share with each driver. The number of different information sets is exponential in the number of drivers. In this work we simplify 1) and 2) into a binary decision for each driver; either share full information or share no information. Theorem III.4 shows that finding the optimal information control even with this binary decision is NP-hard. Moreover, our experiments on real-world ride-sharing data in Section V suggests that even limiting to the binary decision, we achieve significant improvement in the service quality.

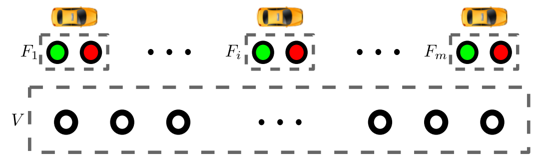


Fig. 3. An instance of Problem III.3. The green vertices represent the desired waiting location of each driver if $\boldsymbol{I_i = \emptyset}$, and the red vertices represent the waiting location of the drivers if $\boldsymbol{I_i = Q}$.

Remark III.2 (Alternate information sets): The approach proposed in this section does not require that the binary choice is between the empty set and the full information set. One can replace this with any two subsets of the information set: For example, for each driver, the binary decision could be between the empty set and the set of all other driver positions within a certain radius. •

Let q'_{i,I_i} be the waiting location selected by driver i from Equation (1) with information I_i, and let $F_i = \{q'_{i,Q}, q'_{i,\emptyset}\}$ be the set of candidate waiting locations for driver i. The formal definition of the problem of sharing information as follows:

Problem III.3: Consider a metric graph $G = (\cup_{i=1}^{m} F_i \cup V, E, c)$. Find a new configuration Q' by picking only one vertex from each F_i, i.e., such that $|Q' \cap F_i| = 1$ for each i, while minimizing the global objective $J_{\mathrm{exp}}(Q')$ or $J_{\max}(Q')$.

Figure 3 shows an instance of the information sharing problem. For each driver, there are two candidate waiting locations based on the information shared with them. The green (resp. red) vertex represents the waiting location of the driver if she has no information (resp. full information) on the position of the other drivers. Let Q' be the solution to Problem III.3 in which if $q_{i,Q} \in Q'$ then the driver i is provided complete information, and no information is available for driver i if $q'_{i,\emptyset} \in Q'$. In the solution to the information sharing problem, if driver i is selected to receive information on the location of drivers, a snapshot of the location of drivers is presented to driver i and the driver can calculate their expected profit based on complete information. We assume that the next waiting location of drivers only depends on the information shared with them. The case of adversarial drivers that leverage the information shared with them to anticipate the information shared with other drivers and adjust their location accordingly is out of the scope of this paper.

First, we analyze the complexity of Problem III.3 for the two global objectives (2) and (3), and then we provide our algorithm for each of the global objectives.

Theorem III.4: The problem of finding the optimal information sharing control, i.e. Problem III.3, with the objective of minimizing the expected wait-time J_{exp} is NP-hard.

Theorem III.5: The problem of finding the optimal information sharing control, i.e. Problem III.3, with the objective of minimizing the maximum wait-time $J_{\max}$ is NP-hard.

The proofs of the Theorems III.4 and III.5 are provided in Appendix A.

B. Minimizing The Expected Wait Time

Given that Problem III.3 is NP-hard for both objectives, we turn our focus to sub-optimal algorithms. In particular, we provide a simple Linear Program (LP)-rounding algorithm for Problem III.3 with the global objective J_{exp}. Although we do not provide bounds on the performance of the LP-rounding algorithm, we evaluate the performance of the algorithm on an extensive set of real-world ride-sharing data in Section V and we show that the proposed algorithm is on average within 0.021% of the optimal.

First we provide an integer linear program (ILP) formulation for Problem III.3 with the global objective J_{exp}. Let binary variable $x_{u,v} \in \{0,1\}$ denote the assignment of a request at v to a driver at vertex u if $x_{u,v} = 1$, and $x_{u,v} = 0$ otherwise. Let the binary variable $y_u \in \{0,1\}$ for all $u \in \cup_i^m F_i$ represent if there is a driver assigned to wait for next request arrival at u. Then we write the ILP for Problem III.3 as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} \sum_{u \in \cup_i^m F_i} p_a(v) c(u,v) x_{u,v} && \text{(4a)}\\
\text{subject to} \quad & \sum_{u \in \cup_i^m F_i} x_{u,v} \geq 1, \forall v \in V, && \text{(4b)}\\
& x_{u,v} \leq y_u, \forall v \in V, \forall u \in \cup_i^m F_i, && \text{(4c)}\\
& y_u + y_v = 1, F_i = \{u,v\}, i \in [m], && \text{(4d)}\\
& y_u, x_{u,v} \in \{0,1\}, \forall v \in V, \forall u \cup_i^m F_i. && \text{(4e)}
\end{aligned}$$

By constraint (4b), a feasible solution assigns each request location to a driver. Equation (4c) ensures that a request is assigned to u only if there is a driver located at u, and finally Equation (4d) shows that in a feasible solution only one of the candidate waiting locations is chosen from each subset F_i, which represent that either the information is provided to a driver or otherwise.

Now we propose our LP-rounding algorithm for Problem III.3 with the global objective J_{exp}. Let $(\mathbf{x}', \mathbf{y}')$ be the solution to the LP relaxation of ILP (4). Given the optimal solution $(\mathbf{x}', \mathbf{y}')$ to the LP relaxation we construct an integer solution to ILP (4) by setting $y_u = 1$ for each vertex u with $y'_u > 1/2$ and $y_u = 0$ otherwise. In a case, $F_i = \{u, v\}$ and $y'_u = y'_v = 1/2$, we set $y_u = 1$ where u is the waiting location of driver i with $I_i = \emptyset$. Finally, we assign each demand vertex $v \in V$ to the closest candidate waiting location u with $y_u = 1$ by setting $x_{u,v} = 1$. Observe that the constructed integer solution solution (x, y) by the LP-rounding algorithm satisfies the constraints of ILP (4). Also, observe that the optimal objective value to the LP relaxation is a lower-bound on the optimal value of ILP (4) and provides a bound on the performance of the LP-rounding algorithm.

C. Minimizing The Maximum Wait-Time

In this section, we propose an algorithm for Problem III.3 with the objective of minimizing the maximum wait-time. We also prove that the solution provided by the proposed algorithm is within a factor of 3 of the optimal control.

The proposed algorithm makes a guess on the optimal maximum wait time of the demands on the vertices, removes the edges longer than the guess, then tries to find a subset of $\cup_i^m F_i$ in the resulting graph such that there is an edge between any $v \in V$ and a vertex in the subset.

Let T be our guess for the maximum wait-time in the optimal solution of Problem III.3. Let $H = (V, E_H)$ be the induced graph of G by deleting the edges longer than T, i.e., $E_H = \{e \in E | c(e) \leq T\}$. Let $\mathcal{N}^H(v)$ denote the vertices in $\cup_{i \in [m]} F_i$ with an edge incident to vertex v in graph H. At any step of the algorithm, if there exist a $v \in V$ such that $|\mathcal{N}(v)| = 0$, then we reached a *conflict*, and we increase our guess T. Let $\mathcal{N}_k(v)$ denote the neighbours of v after adding k vertices to the solution. The approximation algorithm for Problem III.3 with the objective of minimizing the maximum time consists of the following two subroutines:

Subroutine I: For any vertex v with $|\mathcal{N}_k(v)| = 1$, in a sequential manner, we add the vertex $u \in \mathcal{N}_k(v)$ to our solution. At each step, we declare the vertices in V within distance $3T$ of u as serviced. Also, we remove the vertex $w = F_i \setminus \{u\}$ and all the edges incident to it, and we update $\mathcal{N}_k(v)$ for all $v \in V$. If at any stage of this process, there is a v with $|\mathcal{N}_k(v)| = 0$, then there exists a conflict and we increase our guess T.

Subroutine II: Assuming that Subroutine I is completed without any conflicts, then at the start of the Subroutine II of the algorithm all the vertices in V that are not serviced at the end of Subroutine I have $|\mathcal{N}_k(v)| \geq 2$. Starting from an arbitrary vertex $v \in V$, we add a vertex $u \in \mathcal{N}_k(v)$ to the solution and declare all the vertices in V within distance $3T$ of u as serviced. We also, remove $w = F_i \setminus \{u\}$ and all the edges incident to it, and we update $\mathcal{N}_k(v)$ for all $v \in V$. Then we execute the Subroutine I for all the vertices with $|\mathcal{N}_{k+1}(v)| = 1$. Subroutine II continues until all the vertices in V are serviced.

The algorithm performs a binary search on $T \in [\min_{u,v \in V} c(u,v), \max_{u,v} c(u,v)]$ to find the optimal maximum wait-time T^*. Observe that there is no conflict in Subroutine I for any $T \geq T^*$, therefore, in the course of the binary search, the algorithm will reach $T = T^*$ and return no conflicts. Prior to the result on the solution quality, we show the following result on Subroutine II.

Lemma III.6: There is no conflict in Subroutine II.

Proof: The detailed proof is provided in Appendix A. ■

Theorem III.7: The proposed algorithm is a 3-approximation algorithm for Problem III.3 with global objective $J_{\max}$.

Proof: Assume that the algorithm returns a conflict with $T > T^*$. By Lemma III.6, there is no conflict in the execution of Subroutine II, therefore the conflict can only happen in the first execution of Subroutine I. Let H^* be the induced graph by removing edges longer that T^* in E. Since $T > T^*$, then $\mathcal{N}^{H^*}(v) \subseteq \mathcal{N}^H(v)$, i.e., if vertex v is serviced in the optimal solution by $q_i \in \mathcal{N}^{H^*}(v)$, then $q_i \in \mathcal{N}^H(v)$ can service v in time $T > T^*$. Therefore, if there is a conflict in H, there is a conflict in H^*, which is a contradiction. ■

D. Information Sharing for Noisy Drivers

The proposed information sharing algorithms to minimize the J_{exp} and $J_{\max}$ assumes that the drivers pick their next waiting location according to Equation (1). However, the drivers might demonstrate noisy behavior. We model this behavior as follows:

$$\arg\max_{u\in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B_i - c(q_i, u), I_i) + \mathcal{Z}_u, \quad (5)$$

where $\mathcal{Z}_u$ is the zero mean noise which represents the uncertainty in the behavior of the drivers. We refer to the drivers following this model as noisy drivers.

In this section, we extend the proposed algorithms to capture the uncertainty in the behavior of the drivers. Observe that the set $F_i = \{q_{i,Q}, q_{i,\emptyset}\}$ consists of the waiting locations of driver i given full information or no information on the position of the other drivers. In the presence of uncertainty in the behavior of the drivers, $q_{i,\emptyset}$ (resp. $q_{i,Q}$) represents the *expected* waiting location of driver i given no information (resp. full information). Let $p^i_w(u, I)$ be the probability that driver i picks u as her next waiting location given information I according to Equation (5). Then the expected cost of servicing a demand at v by driver i given information I is $c(q_{i,I}, v) = \sum_{u\in V} p^i_w(u, I)c(u, v)$. To capture the noisy behavior of the drivers, $q_{i,\emptyset}$ and $q_{i,Q}$ in F_i for $i \in [m]$ become the *expected* waiting locations. With this modification, the proposed algorithms are applicable to the problem with noisy drivers. Following result shows the performance of the proposed algorithm for the problem of minimizing the maximum wait-time with noisy drivers.

Corollary III.8: The proposed algorithm to minimize the maximum wait-time is a 3-approximation algorithm for Problem III.3 with noisy drivers and global objective $J_{\max}$.
The proof of Corollary III.8 is provided in Appendix A.

IV. Pay to control

In this section, we propose another in-direct control method to relocate the drivers. The idea is that to incentivize a driver to relocate, the difference between the driver's expected profit at the waiting location from Equation (1) and the expected profit at the desired waiting location needs to be compensated. We pose the problem between the drivers and the service provider as a leader-follower game [23] and provide our algorithms for finding the optimal policy of the service provider.

A. Service Provider's Game

Let $Q = \{q_1, \dots, q_m\}$ be the current configuration of the drivers. Then, the game between the drivers and service provider consists of the following:

(i) m players and a service provider,
(ii) An action set A_i for each driver i, which is the waiting locations in the graph, i.e. $A_i = V$ for all $i \in [m]$. The action set of the service provider is $\mathcal{Q}$; and
(iii) The utility function of the service provider is $h(Q') = \sum_{i\in m} d_i\sigma c(q_i, q'_i) + \beta J(Q')$, where d_i is the incentive per unit distance offered to driver i and $\beta \geq 0$ is a user-defined parameter. If β is small, the service provider will offer waiting locations close to the driver's desired waiting location in order to minimize incentive pay. If β is large, the service provider will offer larger payments to relocate the drivers to configurations with minimum expected response time.
(iv) The profit of driver i is the maximum between the expected profit of the offered waiting location including incentive pay and the expected profit of the waiting location from Equation (1), i.e.,

$$\max\{(d_i - 1)\sigma c(q_i, q'_i) + \mathcal{V}(q'_i, B_i - c(q_i, q'_i), I_i),$$
$$\max_{u\in V} -\sigma c(q_i, u) + \mathcal{V}(u, B_i - c(q_i, u), I_i)\}.$$

Driver i will accept the offer by the service provider to relocate to q'_i only if the offered incentives surpass the best-expected profit of the driver. Since the profit functions of the drivers are known to the service provider, then the minimum d_i in which the drivers will accept the offer to move to configuration Q' is

$$d_i = \frac{\max_{u\in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u), I_i)}{\sigma c(q_i, q'_i)} - \frac{\mathcal{V}_i(q'_i, B_i - c(q_i, q'_i), I_i)}{\sigma c(q_i, q'_i)} + 1. \quad (6)$$

Knowing this minimum d_i, the objective of the service provider becomes

$$h(Q') = \sum_{i\in m} \sigma c(q_i, q'_i) - \mathcal{V}_i(q'_i, B_i - c(q_i, q'_i), I_i) + \beta J(Q') + \sum_{i\in m} \max_{u\in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u), I_i). \quad (7)$$

Remark IV.1 (Equilibrium): The optimal solution to the problem $\min_{Q'} h(Q')$ is the equilibrium of the leader-follower game between the service provider and the drivers. Since any other configuration will increase the cost function of the service provider. In addition, By Equation (6), waiting in a location other than the one suggested by the service provider will decrease driver's expected profit.

B. Minimizing The Expected Wait-Time

First, observe that finding the optimal configuration Q' in Equation (7) is independent of $\sum_{i\in m} \max_{u\in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u), I_i)$. Therefore, the problem of minimizing the utility function of the service provider with the global objective $J_{\exp}$, has the mobile facility location (MFL) problem as a special case where $\mathcal{V}_i(v, B_i - c(u, v)) = 0$ for all $u, v \in V$ and $i \in [m]$. The MFL is a well-known NP-hard problem [24] where given a metric graph $G = (F \cup D, E, c)$, mapping $\mu : D \to \mathbb{R}_+$ and a subset $Q \subseteq F \cup D$ of size m. The objective is to find a subset $Q' = \{q'_1, \dots, q'_m\} \subseteq F$ minimizing $\sum_{i\in[m]} c(q_i, q'_i) + \sum_{u\in D} \mu_u \min_{q'\in Q'} c(u, q')$.

Let $w_{q'_i, I_i} = \frac{1}{\sigma}\big[\max_{u\in V} \sigma c(q_i, u) - \mathcal{V}_i(u, B - c(q_i, u)) + \mathcal{V}(q'_i, B_i - c(q_i, q'_i), I_i)\big]$, then the utility function of the service provider becomes

$$h(Q') = \sigma\Big[\sum_{i\in[m]} \big(c(q_i, q'_i) - w_{q'_i, I_i}\big) + \frac{\beta}{\sigma} J_{\exp}(Q')\Big].$$

We now propose a constant factor approximation for the minimum pay-to-control problem, namely minimizing Equation (7). The algorithm follows by a reduction from the minimum pay-to-control problem to MFL.

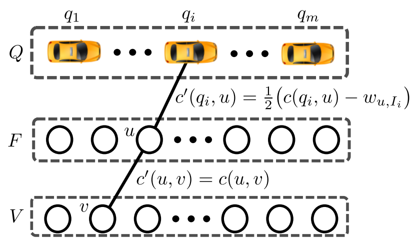


Fig. 4. Constructed MFL instance

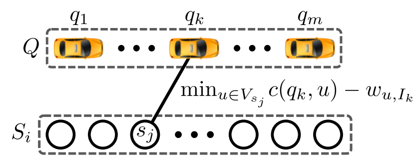


Fig. 5. Constructed bipartite graph for the problem of minimizing service provider's utility function with the global objective of $J_{\max}$.

Given an instance of the minimum pay-to-control problem we construct an MFL instance as follows:

(i) A graph $G=(Q\cup F\cup V, E, c')$ where F is the set of possible waiting locations for the drivers
(ii) There is an edge between $q_i\in Q$ and $q'\in F$ with cost $c'(q_i,q')=\frac{1}{2}\big(c(q_i,q')-w_{q',I_i}\big)$,
(iii) There is an edge between $q'\in F$ and $v\in V$ with cost $c'(q',v)=c(q',v)$.
(iv) The objective is to find a subset of $Q'\subseteq F$ with $|Q'|=m$ such that minimizes

$$C(Q')=\sum_{i\in m} c'(q_i,q_i')+\frac{\beta}{2\sigma}\sum_{u\in V} p_a(u)\min_{i\in[m]} c'(q_i',u).$$

Figure 4 shows the constructed MFL instance. Suppose Q' is a solution to the MFL instance, we let Q' be the solution of the minimum pay-to-control problem and provide the following result on the cost of the solution.

Theorem IV.2: Given an α-approximation algorithm for the MFL problem, the reduction above provides an α-approximation for the pay-to-control problem with objective J_{exp} and any $\beta>0$.

Proof: For any $Q'\subseteq F$, by the construction of the MFL instance, we have $h(Q')=2\sigma C(Q')$. Therefore, given an α-approximation algorithm for the MFL problem, and Q' obtained from the constructed MFL instance, we select Q' as a solution to the minimum pay-to-control problem. Hence, $h(Q')\le 2\alpha\sigma\min_{Q^*\subseteq F} C(Q^*)=\alpha\min_{Q^*\subseteq F} h(Q^*)$. ■

By the result of Theorem IV.2, the $3+o(1)$-approximation algorithm for the MFL problem in [24] applies to the minimum pay-to-control problem.

C. Minimizing The Maximum Wait-Time

In this section, we propose an algorithm for minimizing the service provider's utility function $h(Q)$ in Equation (7) where the measure of the service quality is the maximum wait-time, i.e., $J_{\max}(Q)$. Therefore, the utility function of the service provider becomes

$$\begin{aligned}h(Q')=&\sum_{i\in m}\sigma c(q_i,q_i')+\beta\max_{u\in V}\min_{q'\in Q'} c(u,q')\\&-\sum_{i\in m}\mathcal{V}_i(q_i',B_i-c(q_i,q_i'),I_i)\\&+\sum_{i\in m}\max_{u\in V}-\sigma c(q_i,u)+\mathcal{V}_i(u,B-c(q_i,u),I_i).\end{aligned}$$

Similar to the previous section, observe that the minimization of $h(Q')$ (see Equation (7)) is independent of the term $\sum_{i\in m}\max_{u\in V}-\sigma c(q_i,u)+\mathcal{V}_i(u,B-c(q_i,u),I_i)$. Therefore, the problem of minimizing the service provider's utility function has the metric k-center problem [25] as a special case with $\sigma=0$ and $\sum_{i\in m}\mathcal{V}_i(q_i',B_i-c(q_i,q_i'),I_i)=0$. The metric k-center problem is a well-known NP-hard problem.

Now consider the graph $G=(V,E,c)$ on the demand vertices. Without the loss of generality, we assume $c(e_1)\le c(e_2)\le\ldots\le c(e_{|V|^2})$ where $e_i\in E$ for all $i\in\{1,\ldots,|V|^2\}$. Now consider a set of sub-graphs $\{G_1,\ldots,G_{|V|^2}\}$ where $G_i=(V,E_i,c)$ with $E_i=\{e\in E|c(e)\le c(e_i)\}$.

The proposed algorithm for this problem is built on the approximation algorithm for the metric k-center problem in [25]. For each sub-graph G_i, we construct a graph $H_i^2=(V,E^i_{H^2})$ where $E^i_{H^2}=\{(u,v)|\exists w\in V,(u,w)\in E_i,(w,v)\in E_i\}$.

For each graph H_i^2, starting from an empty set S_i, we greedily add a vertex in $u\in V\setminus S_i$ to S_i if there is no edge between u and any vertex in S_i. We continue adding the vertices to S_i until all the vertices in $V\setminus S_i$ have an edge incident to a vertex in S_i. The set S_i is called a *maximal independent set*. We call a maximal independent set S_i *valid*, if $|S_i|\le m$. Then we construct a bipartite graph for each valid S_i by adding vertices in $Q\cup S_i$. We add an edge between $q_k\in Q$ and $s_j\in S_i$ with cost $\min_{u\in V_{s_j}} c(q_k,u)-w_{u,I_k}$ where $V_{s_j}=\{u\in V|c(u,s_j)\le c(e_i)\}$.

After constructing the graph, we find the optimal assignment in the resulting bipartite graph using the Hungarian algorithm [26]. Let $\text{Assgn}(Q,S_i)$ represent the cost of the optimal assignment for each valid S_i. Let $\text{Assgn}(Q,S')+\beta c(e')$ be the smallest value among the valid independent sets. Let $s_j\in S'$ be the vertex assigned to $q_k\in Q$ is the solution of the assignment problem. Then we add $q_j'=\arg\min_{u\in V_{s_j}} c(q_k,u)-w_{u,I_k}$ to the final solution.

Let Q^* be the optimal solution to the problem of minimizing the service provider's utility function, let T^* be the corresponding maximum wait-time in the optimal solution and let Q' be the configuration obtained from the proposed algorithm. Now we prove the following result on the steps of the algorithm.

Lemma IV.3: If the maximal independent set S_i is not a valid independent set, i.e. $|S_i|>m$, then $T^*\ge c(e_i)$.

Proof: Proof of the result is provided in Appendix A. ■

Theorem IV.4: The proposed algorithm provides a 3-approximation for the pay-to-control problem with objective $J_{\max}$ and any $\beta>0$.

Proof: In the course of the algorithm, we have considered the graph G_i where $c(e_i)=T^*$. By Lemma IV.3, we have $|S_i|\le m$. Also note that for each $s_j\in S_i$, there is a vertex $q_k\in Q^*$ in V_{s_j}. Observe that, while constructing the bipartite graph, we considered the minimum payment cost from each

$q \in Q$ to the vertices in V_{s_j}, therefore, $\text{Assgn}(Q, Q') \leq \text{Assgn}(Q, Q^*)$. Since for each $v \in V$, there is a vertex $s_j \in S_i$ such that $c(s_j, v) \leq 2T^*$, therefore, there is a vertex $q_k \in Q'$ such that $c(q_k, v) \leq c(q_k, s_j) + c(s_j, v) \leq 3T^*$. Finally we have $h(Q') = \text{Assgn}(Q, Q') + \beta J_{\max}(Q') \leq \text{Assgn}(Q, Q') + 3\beta T^*$. Hence, $h(Q') \leq 3\text{Assgn}(Q, Q^*) + 3\beta T^* = 3h(Q^*)$ ■

Remark IV.5: In the presence of noisy driver i, the *expected* payment per unit distance d_i required to convince the driver to relocate is

$$\mathbb{E}(d_i) = \frac{\mathbb{E}(\max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u), I_i))}{\sigma c(q_i, q_i')} - \frac{\mathcal{V}_i(q_i', B_i - c(q_i, q_i'), I_i)}{\sigma c(q_i, q_i')} + 1.$$

Therefore, the utility function of the service provider becomes $h(Q') = \sum_{i \in m} \mathbb{E}(d_i)\sigma c(q_i.q_i') + \beta J(Q')$, where $\sum_{i \in m} \mathbb{E}(d_i)\sigma c(q_i.q_i')$ is the *expected* payment to the drivers. The problem formulation with the new utility function for the service provider and the proposed algorithms follows directly from the scenario with deterministic drivers. Also, the result in Theorems IV.2 (resp. Theorem (IV.4)) is valid for the problem of minimizing J_{exp} (resp. $J_{\max}$) with noisy drivers. •

V. Simulation Results

In this section, we evaluate the performance of the two proposed control methods on real-world ride-sharing data for yellow taxis in Manhattan, N.Y. [27]. In our experiments, we consider the ride requests for a randomly chosen day 10/06/2016, since the data-set for the year 2016 is the latest available data-set that provides the coordinates of pick-up and drop-off locations. To reduce the complexity of the large data set with 401464 pick-up locations, we clustered the ride requests using K-means algorithm [28] into 500 clusters. Figure 6 shows the clustered pick-up locations and the arrival rates at each cluster is represented with a bar. The drop-off probability $p_d(w|v)$ is obtained from the average number of ride requests assigned to cluster v with destination closest to the center of cluster w.

We consider two global objectives: 1) the expected wait-time, and 2) the maximum wait-time and two scenarios: 1) random initial configuration where the initial location of the drivers are selected uniformly randomly, and 2) jammed initial configuration where the drivers are initialized at the 20 closest locations to the Rockefeller center.

Observe that the proposed algorithms to find the controls are applicable to various driver models for $\mathcal{V}$. In Appendix B, we provide a driver model $\mathcal{V}$ used in the simulations.

A. Information Sharing

We begin by evaluating the performance of the information sharing control method. Observe that the number of unique locations in the set $\cup_i^m F_i$ (see Section III) improve the possibility of providing better solutions for both global objectives. Also note that the drivers that are located at the same location given the same information have the same candidate waiting location. Therefore, for all but one of these drivers, we replace

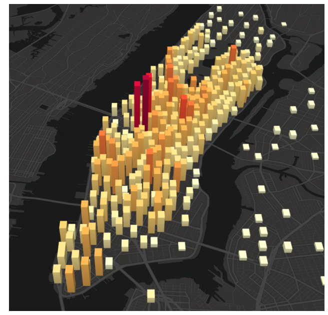


Fig. 6. The set of pick-up and drop off locations in Manhattan. The bars at the locations of clusters represent the ride request arrival rates.

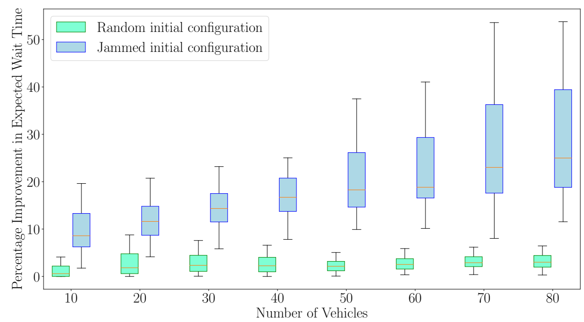


Fig. 7. Improvement in $\boldsymbol{J}_{\textbf{exp}}$ by sharing partial information

the set $F_i = \{q_{i,Q}', q_{i,\emptyset}'\}$ with the set $F_i = \{q_{i,R}', q_{i,\emptyset}'\}$ where R is a random subset of the full information Q.

Figure 7 shows the percentage improvement in the expected wait time for different number of vehicles using the partial information sharing control method of Section III-B in the jammed and random initial configuration scenarios. The results are the average of 200 instances for a varying number of drivers in each scenario. The boxes show the first, second and third quartiles of each set of experiments. Observe that the information sharing algorithm improves the expected wait time of the ride requests by approximately 20% in jammed initial configuration scenario. Also observe that the improvement in the expected wait-time with randomly distributed drivers in the environment is minimal, since the randomly distributed drivers provide a close to optimal expected-wait time, specially in an environment where the arrival rates are not heavily concentrated in a certain area. Therefore, the possibility to improve the expected-wait time with the information sharing algorithm is limited. The expected wait time of the solution obtained from the LP-rounding algorithm of Section III on this set of experiments is on average within 0.021% of optimal. The maximum deviation from the optimal solution is 0.58%. We obtain these bounds by comparing the solution of the LP-rounding algorithm to that from the LP relaxation, which provides an upper bound on the error from optimal.

Figure 8 illustrates the percentage improvement in the maximum wait-time of the ride requests using the partial information sharing control method of Section III-C. Note that the proposed algorithm improves the maximum wait-time of the ride requests by approximately 25% (resp. 5%) in the jammed (resp. random) scenario. In Appendix C, we provide additional results on the performance of the information sharing method

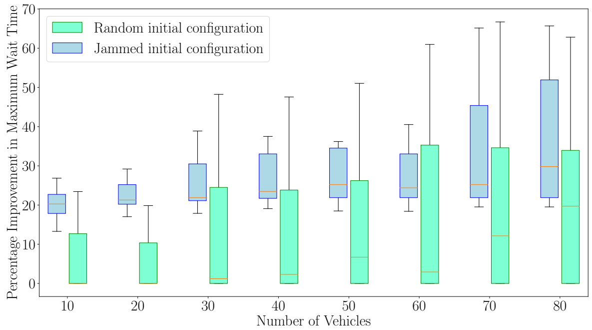


Fig. 8. Improvement in $\boldsymbol{J}_{\mathbf{max}}$ by sharing partial information

in a system of 80 drivers responding to 300 ride requests.

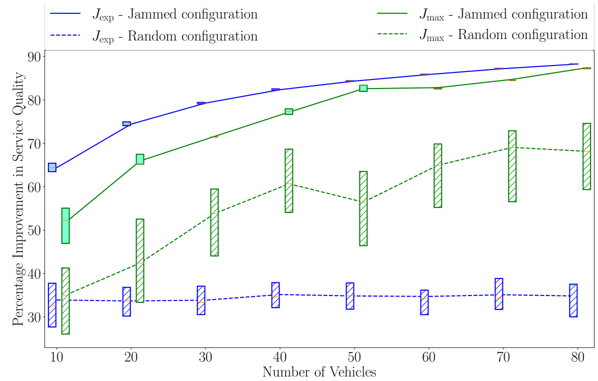
B. Pay to Control

In this section, we evaluate the performance of the pay-to-control method for the jammed and random scenarios. Figure 9(a) illustrates the improvement in the expected and the maximum wait-time for the two scenarios with $\beta = 100$. Note that the proposed algorithm improves the expected wait-time by approximately 75% (resp. 35%) for the jammed (resp.random) scenario. Also observe that the proposed algorithm improves the maximum wait-time by approximately 70% (resp. 50%) in the jammed (resp. random) scenario.

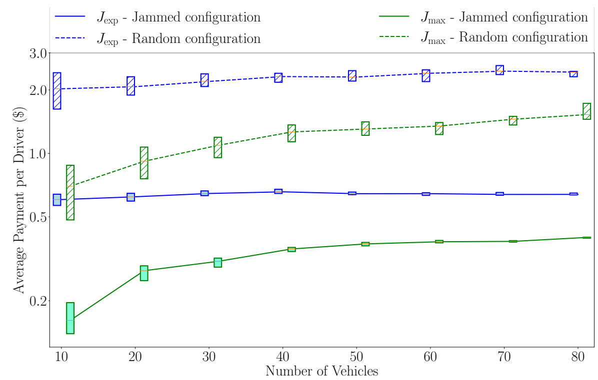
Figure 9(b) shows the payment per driver for the two objectives and the two scenarios. Observe that the expected profit of the drivers in the jammed scenario is smaller than the expected profit of the drivers in the random scenario, therefore, the amount paid to convince the drivers to relocate to desired waiting locations in the jammed scenario is significantly smaller compared to the amount paid in the random scenario.

Figure 10 shows the expected response time of a set of 80 drivers responding to 300 requests arriving over time with the jammed initial configuration. The results are an average of 100 experiments with 300 randomly generated requests for each experiment. The lines represent the average and shaded areas represent the first and third quartiles. The pay-to-control is applied to the system if the expected-wait time is greater than 3 minutes. Observe that the proposed algorithm maintains the low expected wait-time in the course of servicing 300 ride requests. The average pay to maintain the low expected wait-time is 1.95$ per ride request. Figure 11 shows the same experiment with the objective of minimizing the maximum wait-time. The pay-to-control is applied to the system if the maximum wait time is greater than 7 minutes. The average pay to maintain the low expected wait-time is 1.58$ per ride.

In summary, the experiments on the real-world ride-sharing data show that the proposed algorithms significantly improve the expected or the maximum wait-time when the drivers are concentrated in a region. In particular, given a jammed configuration of the drivers, the proposed algorithms improve the service quality significantly with a small number of control inputs and maintain the same quality over-time. The improvements are more evident for the social objective of minimizing the maximum wait-time, since the natural dynamics of the system tend to steer drivers away from the low-demand areas.



(a) The percentage improvement in the service quality



(b) The amount paid to drivers

Fig. 9. Improvement in the global objectives and the amount paid to drivers using pay-to-control method. The blue (resp. green) represents the results for the objective $\boldsymbol{J}_{\mathbf{exp}}$ (resp. $\boldsymbol{J}_{\mathbf{max}}$). The solid bars (resp. hatched bars) represent the jammed (resp. random) initial configuration.

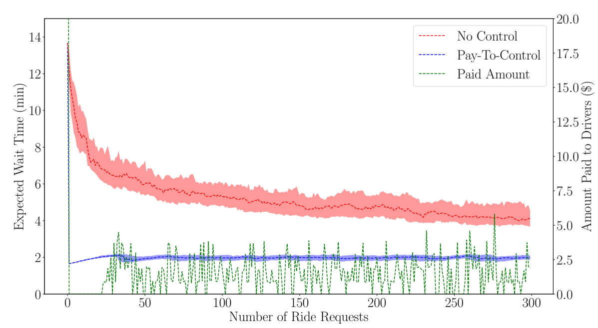


Fig. 10. The expected wait time and the paid amount to a system of $\mathbf{80}$ drivers executing $\mathbf{300}$ ride requests under the pay-to-control method

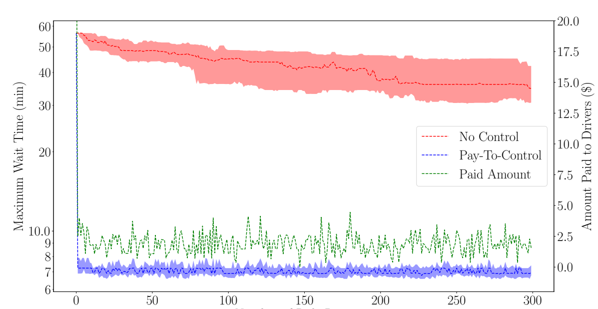


Fig. 11. The maximum wait time and the paid amount to a system of $\mathbf{80}$ drivers executing $\mathbf{300}$ ride requests under the pay-to-control method

Therefore, the control from the proposed algorithm is required to maintain equal service quality across different regions.

VI. Discussion

The proposed control methods do not assume any specific form for the driver model, however, they assume that the driver model is known. Later, we extended the algorithms to capture uncertainty in the behavior of the drivers. For the information sharing method, we substituted the cost of servicing a ride request by a driver with the expected cost of servicing the ride request, and for the pay-to-control method we replaced the compensation with the expected payment to convince the drivers to relocate. A disadvantage to this approach is that if the uncertainty in the driver model is high, then the expected cost of servicing a ride requests and the expected payment can become prohibitively high, and the proposed algorithms will opt to not input any control into the system.

Another short-coming of the proposed methods is the scalability with the number of drivers in the systems due to the combinatorial nature of our proposed approach. The alternative approach to these problems is to use a flow-model. However, this approach suffers scalability issues with the number of regions (stations) in the system. In the flow-model approaches, the environment is usually divided to a small number of regions. A promising direction is to utilize the proposed methods in junction with a flow-model approach, where the solution to the flow-model approach provides the number of drivers in each region, and our proposed methods optimally distribute the drivers within the regions.

VII. Conclusion

This paper considered the problem of controlling self-interested drivers in ride-sharing applications. Two indirect control methods were proposed and for each, a near-optimal algorithm was presented. The extensive results show significant improvement in the expected wait-time and the maximum wait-time on real-world ride-sharing data. In addition, we hope to extend the results to capture vehicles with different capacities and to different ride-sharing applications.

References

[1] L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero, “App-based, on-demand ride services: Comparing taxi and ridesourcing trips and user characteristics in san francisco university of california transportation center (uctc),” *University of California, Berkeley*, 2014.

[2] N. Diakopoulos. How Uber surge pricing really works. [Online]. Available: https://www.washingtonpost.com/news/wonk/wp/2015/04/17/how-uber-surge-pricing-really-works/

[3] A. Rosenblat and L. Stark, “Algorithmic labor and information asymmetries: A case study of Ubers drivers,” *International Journal Of Communication*, 2016.

[4] A. Sadeghi and S. L. Smith, “On re-balancing self-interested agents in ride-sourcing transportation networks,” in *IEEE 58th Conference on Decision and Control*, 2019, pp. 5119–5125.

[5] W. Zhang, S. Guhathakurta, J. Fang, and G. Zhang, “The performance and benefits of a shared autonomous vehicles based dynamic ridesharing system: An agent-based simulation approach,” in *Transportation Research Board 94th Annual Meeting*, no. 15-2919, 2015.

[6] M. Hyland and H. S. Mahmassani, “Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests,” *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 278–297, 2018.

[7] M. Chang, D. S. Hochbaum, Q. Spaen, and M. Velednitsky, “DISPATCH: an optimal algorithm for online perfect bipartite matching with i.i.d. arrivals,” *CoRR*, vol. abs/1805.02014, 2018.

[8] M. Maciejewski, J. Bischoff, and K. Nagel, “An assignment-based approach to efficient real-time city-scale taxi dispatching,” *IEEE Intelligent Systems*, vol. 31, no. 1, pp. 68–77, 2016.

[9] Uber. Driving with Uber, wait less, earn more. [Online]. Available: www.uber.com/ca/en/price-estimate/

[10] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, “Robotic load balancing for mobility-on-demand systems,” *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012.

[11] M. Tsao, R. Iglesias, and M. Pavone, “Stochastic model predictive control for autonomous mobility on demand,” *arXiv preprint arXiv:1804.11074*, 2018.

[12] G. C. Calafiore, C. Novara, F. Portigliotti, and A. Rizzo, “A flow optimization approach for the rebalancing of mobility on demand systems,” in *IEEE International Conference on Decision and Control*, 2017, pp. 5684–5689.

[13] M. Salazar, F. Rossi, M. Schiffer, C. H. Onder, and M. Pavone, “On the interaction between autonomous mobility-on-demand and public transportation systems,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2262–2269.

[14] G. P. Cachon, K. M. Daniels, and R. Lobel, “The role of surge pricing on a service platform with self-scheduling capacity,” *Manufacturing and Service Operations Management*, vol. 19, no. 3, pp. 368–384, 2017.

[15] Q. Wei, J. A. Rodriguez, R. Pedarsani, and S. Coogan, “Ride-sharing networks with mixed autonomy,” in *American Control Conference*, 2019, pp. 3303–3308.

[16] S. Banerjee, R. Johari, and C. Riquelme, “Pricing in ride-sharing platforms: A queueing-theoretic approach,” in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 2015, pp. 639–639.

[17] D. B. Shmoys, “Approximation algorithms for facility location problems,” in *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 2000, pp. 27–32.

[18] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, “Local search heuristics for k-median and facility location problems,” *SIAM Journal on computing*, vol. 33, no. 3, pp. 544–562, 2004.

[19] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam, “Minimizing movement,” *ACM Transactions on Algorithms (TALG)*, vol. 5, no. 3, p. 30, 2009.

[20] A. Sadeghi and S. L. Smith, “Re-deployment algorithms for multiple service robots to optimize task response,” in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2356–2363.

[21] S. Bandyapadhyay, A. Banik, S. Das, and H. Sarkar, “Voronoi game on graphs,” *Theoretical Computer Science*, vol. 562, pp. 270–282, 2015.

[22] R. Salhab, J. Le Ny, and R. P. Malhamé, “A dynamic ride-sourcing game with many drivers,” in *55th Annual Allerton Conference on Communication, Control, and Computing*, 2017, pp. 770–775.

[23] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*. Siam, 1999, vol. 23.

[24] S. Ahmadian, Z. Friggstad, and C. Swamy, “Local-search based approximation algorithms for mobile facility location problems,” in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2013, pp. 1607–1621.

[25] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.

[26] R. Jonker and T. Volgenant, “Improving the hungarian assignment algorithm,” *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, 1986.

[27] (2016) TLC yellow taxi trip data set. [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] R. Schuler, “An algorithm for the satisfiability problem of formulas in conjunctive normal form,” *Journal of Algorithms*, vol. 54, no. 1, pp. 40–44, 2005.

[30] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] The Canadian Automobile Association. Driving costs. [Online]. Available: www.carcosts.caa.ca/

APPENDIX

A. Proof of Results

Proof: [Proof of Theorem 3.4] We prove the NP-hardness of Problem III.3 for minimizing the expected wait-time $J_{\exp}$ with a reduction from CNF-SAT [29] as follows:

Consider an instance of CNF-SAT with m Boolean variables and n clauses. Now we construct an instance of Problem III.3.

(i) For each variable x_i of the CNF-SAT instance, we create a set $F_i = \{v_i^T, v_i^F\}$, where v_i^T will correspond to setting the x_i to true and v_i^F will correspond to setting it to false.
(ii) We let V contain n vertices, one representing each clause in the SAT formula.
(iii) Let E contain an edge for each $v \in \cup_{i=1}^m F_i$ and $w \in V$.
(iv) For each $e = (v, w) \in E$, we set its cost to 1 if the literal v appears in the clause w, and 2 if the literal does not. Note that the costs are metric.
(v) Let $p_a(u) = 1/n$ for each $u \in V$.

Now, we solve the instance of Problem III.3 with the objective of minimizing the expected wait-time $J_{\exp}$. If it returns a subset of $\cup_{i=1}^m F_i$ with cost exactly 1, then for vertex $c \in V$, there is a vertex v in $\cup_{i=1}^m F_i$ with edge cost of 1 to c. Vertex c corresponds to a clause and vertex v corresponds to a literal in c. This implies that the literal chosen from each subset in the partition of $\cup_{i=1}^m F_i$ gives a satisfying truth assignment for the SAT instance. If the subset returned has a cost greater than 1, then there exists a clause $w \in V$ for which every chosen literal has an edge cost of 2. Thus, this clause is not satisfied and no satisfying instance exists. ■

Proof: [Proof of Theorem 3.5] The proof of hardness for Problem III.3 with the objective $J_{\max}$ follows the same steps as the proof of Theorem III.4 with exception of step (v) where $p_a(u) = 1$ for each $u \in V$. □ ■

Proof: [Proof of Lemma 3.6] By contradiction assume there is a conflict in Subroutine II. Therefore, at some step of the execution, there is a vertex v such that $|\mathcal{N}_k(v)| = 0$. Prior to this step, $|\mathcal{N}_{k-1}(v)| = 1$, and there should have been another vertex w with $|\mathcal{N}_{k-1}(w)| = 1$, otherwise the algorithm would have added the vertex in $\mathcal{N}_{k-1}(v)$ to the solution. Observe that the event of $|\mathcal{N}_{k-1}(w)| = 1$ and $|\mathcal{N}_{k-1}(v)| = 1$ shows that at the start of Subroutine II, $\mathcal{N}_i(w) \cap \mathcal{N}_i(v) \neq \emptyset$ for an $i < k$. Let $q_1 \in \mathcal{N}_{k-1}(v)$, $q_2 \in \mathcal{N}_{k-1}(w)$ and $z \in \mathcal{N}_i(w) \cap \mathcal{N}_i(v)$. Without loss of generality, assume that q_i, q_j and q_l correspond to the expected waiting location of the drivers i, j and l where no information is provided to them. Then the cost of servicing v with giving no information to driver j is $c(v, q_j) \leq c(w, q_j) + c(w, v)$ and also note that $c(w, v) \leq c(w, q_l) + c(v, q_l)$ by the triangle inequality.

Therefore, the time to service v by driver j with no information is $c(v, q_j) \leq c(w, q_j) + c(w, v) \leq c(w, q_j) + c(w, q_l) + c(v, q_l) \leq 3T$. Thus, v would have been marked as serviced prior to conflict. ■

Proof: [Proof of Corollary 3.8] We omit the detailed proof due to space constraints, however, the proof directly follows from the proof of Lemma III.6 and Theorem III.7 with only the following modification in the proof of Lemma III.6:

$$
\begin{aligned}
c(v, q_j) &= \sum_{z \in V} p_w^j(z, \emptyset) c(z, v) \leq \sum_{z \in V} p_w^j(l, \emptyset) c(l, w) + c(w, v) \\
&\leq c(w, q_j) + \sum_{z \in V} p_w^l(l, \emptyset) c(z, v) + \sum_{l \in V} p_w^l(z, \emptyset) c(z, w) \\
&\leq c(w, q_j) + c(w, q_l) + c(v, q_l) \leq 3T.
\end{aligned}
$$

■

Proof: [Proof of Lemma 4.3] Suppose, $T^* < c(e_i)$, then for vertex $v \in V$ there is a vertex in $q \in Q^*$, denoted by $\ell_{Q^*}(v)$, with $c(q, v) \leq T^*$. Observe that for any $s_j, s_k \in S_i$, we have $\ell_{Q^*}(s_j) \neq \ell_{Q^*}(s_k)$, otherwise, there is an edge between s_j, s_k in graph H_i^2. This is a contradiction since S_i is a maximal independent set. Therefore, for each vertex in S_i, there is a unique vertex in Q^*. This is a contradiction, since $|Q^*| = m$. ■

B. Drivers' model

A driver model is a function for evaluating the expected profit of different locations at each time instance. The driver model of the drivers in Section V takes into account the environmental parameters such as arrival rates and drop-off probabilities and the information shared with the driver. Let $p_i(v, u)$ be the probability that a ride-request at v is assigned to driver i positioned at u with information I_i. Then, we find the perception of expected profit of a driver as follows:

$$
\begin{aligned}
\mathcal{V}_i(u, B_i, I_i) = \sum_{v, w \in V} & p_d(w|v) \big[p_i(v, u) \big(\max\{0, \sigma' c(w, v) \\
& - \sigma c(u, v) + \mathcal{V}_i(w, B_i - c(u, v) - c(v, w), I_i)\} \big) \big], \quad (8)
\end{aligned}
$$

Note that the calculation of the expected profit of drivers for each time step is computationally expensive, thus we trained a Random Forest Regressor [30] implemented by [28] to approximate the values of $\mathcal{V}$ for each number of vehicles in the system with training data over 10000 instances with work-day B_i of 15 average length rides, fare $\sigma' = \$0.81$ per kilometer [9] and driving cost $\sigma = \$0.15$ per kilometer [31].

We refer the reader to [4] for a detailed description of the model and the proposed method for computing $p_i(v, u)$.

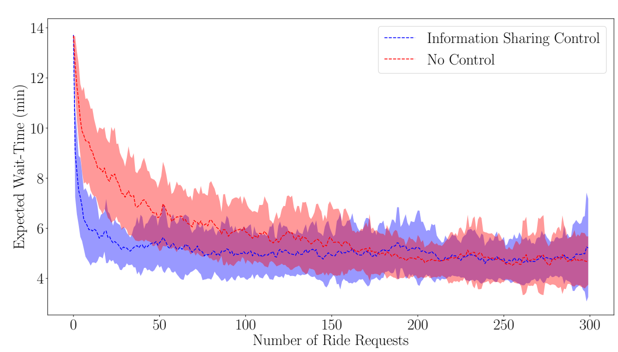


Fig. 12. The expected wait-time of ride requests in a system of **80** drivers executing **300** ride requests arriving over time under the partial information control method.

Fig. 13. The maximum wait-time of ride requests in a system of 80 drivers executing 300 ride requests arriving over time under the partial information control method.

TABLE II
INFORMATION SHARING FOR NOISY DRIVERS

Initial Configuration	J_{exp}		$J_{\max}$	
	Avg.	Std. Dev.	Avg.	Std. Dev.
Random	2.4	0.3	21.8	10.1
Jammed	43.2	4.0	54.7	6.1

C. Information Sharing in a Horizon

Figure 12 shows the expected response time of a set of 80 drivers responding to 300 requests arriving over time with the jammed initial configuration. The figure summarizes 100 experiments with 300 randomly generated requests for each experiment. The lines represent the average and shaded areas represent the first and third quartiles. In the transient, the information sharing algorithm improves the expected response time rapidly. However, in the steady-state the performance is very similar to the no control case. Thus, the information sharing method serves to more quickly disperse the drivers from the initial jammed configuration. The information sharing algorithm is applied whenever the expected wait time is larger than 6 minutes. Figure 13 shows the similar experiment with the objective of minimizing the maximum wait-time. Observe that the proposed algorithm improves the maximum wait-time in the initial jammed configuration and maintains the quality service over the course of responding to 300 requests. To limit the number of information sharing control inputs to the system, we only use the information sharing method when the maximum wait time in the current driver configuration is larger than 30 minutes.

D. Information Sharing for Noisy Drivers

In this section, we evaluate the performance of the proposed information control method in the presence of noisy drivers. We model the uncertainty in the behaviour of the driver (see Equation (5)) with a zero mean uniform noise $\mathcal{Z}_u$ at each vertex where the maximum deviation from $\mathcal{V}_i(u, B_i, I_i)$ is 20%. Table II shows the improvement in the expected wait-time J_{exp} and the maximum wait-time $J_{\max}$ for the two scenarios. The results are the average of 200 trials for each global objective and each initial configuration of the drivers. Observe that the proposed algorithms for the information sharing problem with noisy drivers provide similar performance to the ones with deterministic drivers under random initial configuration. However, the performance of the algorithms improve with uncertainty in the behavior of the drivers under jammed initial configuration. Observe that in the jammed initial configuration with deterministic drivers, the drivers initialized at the same location have similar optimal waiting locations if full-information is provided to them. Therefore, the algorithms provide the information to only one of the drivers positioned at each vertex. However, in the presence of noisy drivers, if full information is provided to multiple drivers position at the same location, the drivers relocate to different waiting locations. Therefore, the algorithms opt to provide information to more drivers resulting in a better distribution of drivers in the environment.