On Re-Balancing Self-Interested Agents in Ride-Sourcing Transportation Networks

Armin Sadeghi   Stephen L. Smith

Abstract— **This paper focuses on the problem of controlling self-interested drivers in ride-sourcing applications. Each driver has the objective of maximizing its profit, while the ride-sourcing company focuses on customer experience by seeking to minimizing the expected wait time for pick-up. These objectives are not usually aligned, and the company has no direct control on the waiting locations of the drivers. In this paper, we provide two indirect control methods to optimize the set of waiting locations of the drivers, thereby minimizing the expected wait time of the customers: 1) sharing the location of all drivers with a subset of drivers, and 2) paying the drivers to relocate. We show that finding the optimal control for each method is NP-hard and we provide algorithms to find near-optimal control in each case. We evaluate the performance of the proposed control methods on real-world data and show that we can achieve between 20% to 80% improvement in the expected response.**

I. Introduction

In recent years, ride-sourcing services such as UberX and Lyft have emerged as an alternative mode of urban transportation. The reduced wait times for pickup of these services is the compelling feature compared to the conventional taxi services [1]. A key factor that affects the response time of the service is where the drivers wait to respond to the next ride request. Ride-sourcing companies do not have control over the position of the drivers as they are self-interested units maximizing their objectives. Therefore, a challenge is to ensure the drivers are distributed throughout the city in order to minimize the expected wait time of the customers. This must be done either by providing information to drivers, or through incentives (payment) that make relocation attractive.

As a method for both re-balancing and increasing the supply of drivers, Uber introduced surge pricing in high demand areas. This reduces the expected response time of servicing the requests by drawing more drivers to those areas. However, the surge pricing 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 requests in ride-sourcing networks can be divided into two major problems: 1) assignment of the ride request to the drivers; and 2) re-balancing of the drivers for future ride requests. In this paper, we focus on the re-balancing problem for a subset of drivers to service ride requests that arrive sequentially in an environment. The drivers' motion in the environment is captured as a road-map (i.e., graph), and each ride request arrives at a node of the graph according to a known arrival rate (see Figure 1). The drivers are self-interested units maximizing the expected profit of their workday. Hence, the ride-sourcing company has no direct control over the waiting locations of the drivers. The objective of the ride-sourcing company is to incentivize the drivers to relocate to a set of waiting locations that minimizes the expected wait-time of the customers.

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)

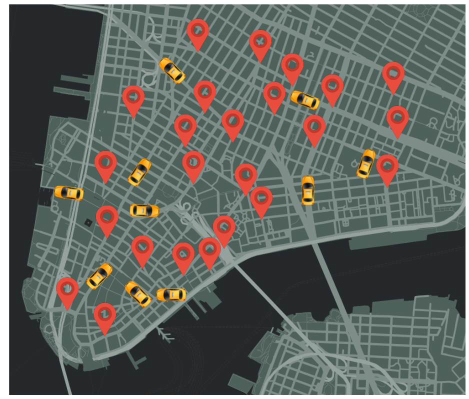


Fig. 1: A set of drivers in the ride-sourcing system and a set of locations with high probability of ride request arrival.

Related Work: The problem of dispatching taxis to service ride requests has been the subject of extensive research [4], [5], [6], [7]. These studies focus on policies to optimally assign the ride requests to the taxis. In contrast, we focus on the waiting locations of the drivers that minimize the expected wait time of the customers. We assume the ride requests are assigned in a first-come-first-serve fashion to the closest available driver which is the common method employed by the ride-sourcing companies [8].

The problem of re-balancing service units in the environment has been studied for various applications. In mobility-on-demand problem (MOD) [9], [10], [11], a group of vehicles are located at a set of stations. The customers arrive at the stations, hire the vehicles for ride, and then drop the vehicles off at their destination stations. The objective is to balance the vehicles at the stations to minimize the expected wait time of the customer. In contrast, 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.

The facility location problem [12], [13] and its extension to the mobile facility location problem (MFL) [14] 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 [13] where the number of open facilities is limited and the cost of opening a facility is zero. In [15], 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.

In the aforementioned studies, the actions of service units are controlled by a central unit and the objective of the service units is aligned with the global objective. However, we consider self-interested units maximizing their own profit such that the ride-sourcing companies have no direct control on their decisions. A closely related problem is that of Voronoi games on graphs [16] where requests arrive on the vertices of a graph and the objective of each self-interested service unit is to maximize the number vertices assigned to them. They cast the problem as a game between the service units prove that the problem of finding the pure Nash equilibrium on general graphs is NP-hard. In [17], the authors provide the best response strategy for each driver and they approximate Nash equilibria, which can be utilized by the ride-sourcing companies to incentivize the drivers in a way to maximizes a global objective. These studies focus on the strategies of the self-interested service units, in contrast, we focus on finding the optimal policy for the ride-sourcing company to optimally respond to the ride requests.

Contributions: The contributions of this paper are three-fold. First, we formulate the re-balancing problem of self-interested service units. Second, we propose two indirect method to relocate the service units to minimize the expected response time and provide algorithms with near-optimal solutions. Third and finally, we evaluate the performance of the control methods on real-world ride-sourcing data.

The paper is organized as follows. In Section II, we formulate the problem of minimizing customer wait-time with self-interested service units. In Section III, we provide the first indirect control method based on sharing the information on the location of the drivers. Section IV.I consists of the second control method based on incentive pay to relocate the drivers to desired waiting locations. Finally in Section V, we provide an extensive set of experiments, on real-world ride-sourcing data, characterizing the performance of the two control methods.

II. Problem Formulation

Consider a set of m drivers and a set of pick-up and drop-off locations V. Let $G = (V, E, c)$ be a metric graph on vertices V, let E be the set of edges between the locations and $c : E \to \mathbb{R}_+$ be the function assigning a travel time to each edge of the graph. 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 $\mathcal{Q}$. Each driver i is aware of the position of a subset of the drivers $I_i \subseteq Q$, for instance, each driver may be aware of the location of the other drivers in its vicinity.

We assume that the requests arrive at each vertex u according to an independent Poisson process with arrival rate λ_u. 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 is the ratio of number of requests arriving at u to the total number of requests arriving in a period of time. Let the drop-off probability $p_d(\text{dropoff} = w|\text{pickup} = v)$ be the probability that a request with pickup location at vertex v has a drop-off location at vertex w.

Driver i's *perception of her expected profit* is a function of her information I_i on the location of other drivers, environment parameters such as arrival times, her waiting location q_i and the period of working time B_i, denoted by $\mathcal{V}_i(u, B_i)$. For the development of our main control methods we do not assume any specific form of this function. We do assume, however, that the ride-sourcing company has access to this function, obtained through data of driver behavior. In Section V we present one potential model of $\mathcal{V}_i(u, B_i)$, which is then used for simulating the two control methods.

Drivers' objective: Each driver is a self-interested unit, therefore, they will wait at a location that maximizes their expected profit, i.e.,

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

where σ is the cost per minute of driving.

Global objective: In addition to the objective of each driver, there is a global objective for the service providers to maximize the service quality by minimizing the expected wait time of the customers until pick-up, i.e.,

$$\min_{Q \in \mathcal{Q}} D(Q) = \sum_{u \in V} \min_{q_i \in Q} p_a(u) c(q_i, u), \tag{2}$$

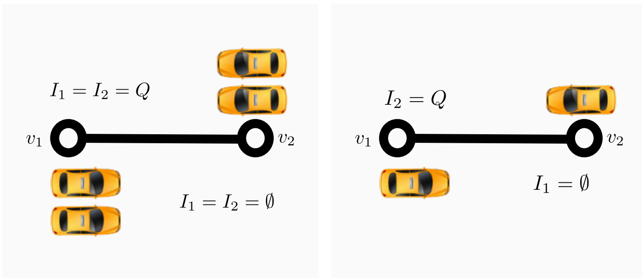
The main challenge in optimizing the global objective is that the drivers are self-interested units and the service provider does not have any direct control on the configuration of the vehicles. Therefore, the service provider is not able to minimize the expected response time to the requests directly. 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 can share more information on the location of drivers with a subset of them to manipulate their decision towards relocating to a desired waiting location. We refer to this as the *sharing information* control. 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 behaviour $\mathcal{V}$.

In the following sections, we provide a detailed description of the two control methods.

III. Control by Sharing Information

In this section, we provide an indirect control on the configuration of the drivers exploiting the fact that the optimal waiting location for each driver in Equation (1) is a function of the information provided to the driver regarding the position of the other drivers, i.e., I_i.

Figure 2 demonstrates the importance of information on an instance of the ride-sourcing problem with two vehicles and two request locations. The locations are within unit distance apart and the arrival rate at locations v_1 and v_2 are 0.1 and 0.2, respectively. The vehicles are initially located at v_1 and will relocate to the best waiting location, namely optimizing



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

Fig. 2: Instance of ride-sourcing problem with two vehicles and two request arrival locations.

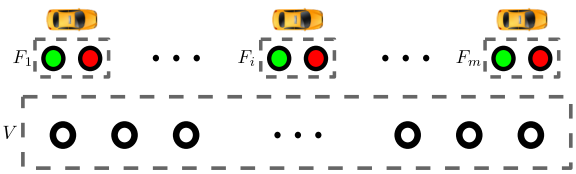


Fig. 3: An instance of Problem III.1. The green vertices represent the desired waiting location of each driver if $I_i = \emptyset$, and the red vertices represent the waiting location of the drivers if $I_i = Q$.

Equation (1). Figure 2a shows the two scenarios where both vehicles are provided the same information, i.e., $I_1 = I_2 = \emptyset$ and $I_1 = I_2 = Q$. Note that the configuration of the vehicles with the same information is the worst possible configuration for the global objective. However, illustrated in Figure 2b, providing the information to a subset of the vehicles results in the optimal configuration for the global objective.

The information sharing problem consists of deciding the subset of drivers we share information with and the information shared with each driver. However, for this work, we consider the binary decision where either full information or no information is provided.

Let $q'_{i,Q}$ (resp. $q'_{i,\emptyset}$) be the new waiting location selected by driver i from Equation (1) with information $I_i = Q$ (resp. $I_i = \emptyset$). Let $F_i = \{q'_{i,Q}, q'_{i,\emptyset}\}$ be the set of candidate waiting locations for driver i. If there exist $i, j \in [m]$ such that $q'_{i,I_i} = q'_{j,I_j}$, then we create a duplicate vertex u for q'_{j,I_j} such that $c(q'_{i,I_j}, v) = c(u, v)$ for all $v \in V$. The formal definition of the problem of sharing information is given as follows:

Problem III.1. 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 $D(Q') = \sum_{u \in V} \min_{q_i \in Q'} p_a(u) c(q, u)$.

Figure 3 shows an instance of the information sharing problem. The green vertices are the waiting locations of the drivers if they have no information on the position of the other drivers, and the red vertices are the waiting locations of the drivers if the information is provided to each driver by the service provider. Let Q' be the solution to Problem III.1 in which if $q_{i,Q} \in Q'$ then the driver i is provided complete information of the position of the other drivers, and no information is available for driver i if $q'_{i,\emptyset} \in Q'$.

First, we analyze the complexity of Problem III.1, and then we provide an LP-rounding algorithm to find the controls.

A. Proof of Hardness

We now prove the NP-hardness of Problem III.1 with a reduction from CNF-SAT [18] as follows: Consider an instance of CNF-SAT with n Boolean variables and m clauses. We will reduce this problem to Problem III.1.

(i) Let $\cup_{i=1}^m F_i$ contain $2n$ vertices, partitioned into n sets of size two. The set F_i contains two vertices, $\{v_i^T, v_i^F\}$, where v_i^T corresponds to setting the ith SAT variable to true (i.e., the positive literal) and v_i^F corresponds to setting it to false (i.e., the negative literal).

(ii) We let V contain m 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 $p_a(u) = 1/m$ for each $u \in V$.

Now, we solve the instance of Problem III.1. If it returns a subset of $\cup_{i=1}^m F_i$ with cost exactly 1, then for each clause $c \in V$, there is literal in $\cup_{i=1}^m F_i$ with edge cost of 1 to 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 edge cost of 2. Thus, this clause is not satisfied and no satisfying instance exists.

B. Linear-Program Rounding Algorithm

Given that Problem III.1 is NP-hard, we turn our focus to suboptimal algorithms. In particular, we provide a simple Linear Program (LP)-rounding algorithm for Problem III.1. Although we do not provide bound on the performance of the LP-rounding algorithm, we evaluate the performance of the algorithm on an extensive set of real-world ride sourcing data in Section V and we show that the proposed algorithm is on average within 0.014% of the optimal.

First we cast Problem III.1 as an integer linear program (ILP), then we propose a rounding algorithm based on the solution to the relaxation of the ILP. Let integer parameter $x_{u,v} \in \{0, 1\}$ 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 integer parameter $y_u \in \{0, 1\}$ 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.1 as follows:

$$\text{minimize} \quad \sum_{v \in V} \sum_{u \in \cup_i^m F_i} p_a(v) c(u, v) x_{u,v} \tag{3}$$

$$\text{subject to:} \sum_{u \in \cup_i^m F_i} x_{u,v} \geq 1, \quad \forall v \in V \tag{4}$$

$$y_u \geq x_{u,v}, \quad \forall v \in V, u \in \cup_i^m F_i \tag{5}$$

$$y_u + y_v = 1, \quad F_i = \{u, v\}, i \in [m] \tag{6}$$

$$y_u, x_{u,v} \in \{0, 1\}, \quad \forall v \in V, u \cup_i^m F_i$$

By constraint (4), a feasible solution assigns each request location to a driver. Equation (5) ensures that a request is assigned to u only if there is a driver located at u, and finally Equation (6) 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.1. Let $(\mathbf{x}', \mathbf{y}')$ be the solution to the LP relaxation of ILP (3). Without loss of generality for all $F_i = \{u, v\}, i \in [m]$, let $y_u \geq 1/2$ and $y_v \leq 1/2$. Given solution $(\mathbf{x}', \mathbf{y}')$ we construct an integer solution to ILP (3) by setting $y_u = 1$ for each vertex u with $y'_u > 1/2$ and $y_v = 0$. In a case, $F_i = \{u, v\}$ and $y'_u = y'_v = 1/2$, we set $y_u = 1$ where u is the optimal waiting location of driver i with $I_i = \emptyset$. Then we assign each vertex $v \in V$ to the closest vertex u in $\cup_{i=1}^{m} F_i$ with $y_u = 1$ by setting $x_{u,v} = 1$. Note that the constructed solution (x, y) satisfies the constraint of ILP (3), therefore, it is a feasible solution to Problem III.1. Also, observe that the optimal objective value to the LP relaxation is a lower-bound on the optimal value of ILP (3) and provides a bound on the performance of the LP-rounding algorithm.

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. This method employed at each time step and presents information to a driver if there is an opportunity to improve the expected response time.

The information sharing method indirectly controls the configuration of the drivers by providing information to a subset of them, however, the possible configurations are limited to the candidate waiting locations of the drivers. In the following section, we provide the details on the pay-to-control method for the service provider.

IV. Pay to control

Each driver as a self-interested unit chooses its waiting location by maximizing the profit in Equation (1). To convince the vehicles to relocate to another configuration, the service provider needs to compensate for the difference between their expected profit of the new location and their expected profit for the waiting location from Equation (1). First, we pose the problem between the drivers and the service provider as a game. Then we provide an approximation algorithm to find the optimal policy for the service provider.

A. Service Provider's Game

Let d_i be the incentive per unit distance offered to driver i. Let $Q = \{q_1, \ldots, q_m\}$ (resp. $Q' = \{q'_1, \ldots, q'_m\}$) be the configuration of the drivers before (resp. after) the incentive pay. The game between the drivers and service provider consists of the following:

- A set of m players and a service provider,
- An action set A_i for each driver i, which is the waiting locations in the graph, i.e. $A_i = V \; \forall i \in [m]$. The action set of the service provider is $\mathcal{Q}$; and
- The profit function of the service provider is $h(Q') = \sum_{i \in m} d_i \sigma c(q_i, q'_i) + \beta D(Q')$, where $\beta \geq 0$ is a user-defined parameter that indicates the importance of the service quality for the service provider with respect to the incentive pay. For a small value of β, the incentive pay is in the priority, thus the service provider will offer the waiting locations close to the drivers' desired waiting locations, however, for large values of β, the service provider accepts high incentive pay to relocate the drivers to a configuration with minimum expected response time.
- The profit of driver i is the maximum of the expected profit of the offered waiting location with incentive pay and the expected profit of the waiting location from Equation (1), i.e.,

$$\max\{(d_i - 1)\sigma c(q_i, q'_i) + \mathcal{V}(q'_i, B_i - c(q_i, q'_i)), \\ \max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}(u, B_i - c(q_i, u))\}. \quad (7)$$

This is an instance of a leader-follower game [19]. The service provider offers an incentive based on its utility and the drivers as followers either take the offer or reject it. The service provider is aware of the best action of the drivers given any action taken by the service provider (i.e., incentive pay and the offered waiting location). The objective is to find the optimal strategy for the service provider to minimize a linear combination of the incentive pay and the expected response time by relocating the drivers to the desired configuration.

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

$$d_i = \frac{\max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u))}{c(q_i, q'_i)} \\ - \frac{\mathcal{V}_i(q'_i, B_i - c(q_i, q'_i))}{\sigma c(q_i, q'_i)} + 1. \quad (8)$$

In the equation above, $\max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u))$ is the maximum expected profit of the driver i by relocating to a new waiting location, and $V_i(q'_i, B_i - c(q_i, q'_i))$ is the expected profit of driver i by waiting at the location q'_i offered by the service provider. Knowing this minimum d_i, the objective of the service provider becomes

$$h(Q') = \sum_{i \in m} \sigma c(q_i, q'_i) + \beta \sum_{u \in V} p_u \min_{i \in [m]} c(q'_i, u) \\ + \sum_{i \in m} \max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u)) \\ - \sum_{i \in m} \mathcal{V}_i(q'_i, B_i - c(q_i, q'_i)). \quad (9)$$

Observe that $\sum_{i \in m} \max_{u \in V} -\sigma c(q_i, u) + \mathcal{V}_i(u, B - c(q_i, u))$ is independent of the optimization parameters. Therefore, the problem of minimizing the utility function of the service provider h has the mobile facility location (MFL) problem

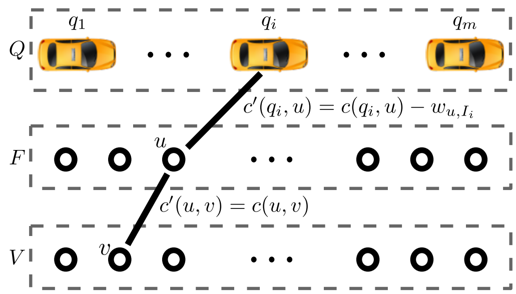


Fig. 4: Constructed MFL instance for optimizing the utility of the service provider. An instance of the edges between the subsets is shown with their respective costs.

Fig. 5: The set of pick-up and drop off locations in Manhattan N.Y. The bars represent the arrival rate for the clusters.

as a special case where $V_i(v, B_i - c(u, v)) = 0$ for all $u, v \in V$ and $i \in [m]$. The MFL is a well-known NP-hard problem [20] where given a metric graph $G = (F \cup D, E, c)$, mapping $\mu : D \rightarrow \mathbb{R}_+$ and a subset $Q \subseteq F \cup D$ of size m. The objective is to find a subset $Q' = \{q'_1, \ldots, q'_m\} \subseteq F$ minimizing $\sum_{i \in [m]} c(q_i, q'_i) + \sum_{u \in D} \mu_u \min_{q' \in Q'} c(u, q')$.

Remark IV.1 (Equilibrium). Observe that the solution to the problem $\min_{Q'} h(Q')$ provides the minimum cost for the objective of the service provider. Furthermore, by Equations (7) and (8), waiting in locations other than the suggested locations by the service provider decrease drives' expected profit. Therefore, an optimal solution to the problem $\min_{Q'} h(Q')$ is an equilibrium for the leader-follower game between the service provider and drivers.

B. Approximation Algorithm

We now propose a constant factor approximation for the minimum pay-to-control problem, namely minimizing Equation (9). Let $w_{q'_i, I_i} = \frac{1}{\sigma} \max_{u \in V} \sigma c(q_i, u) - \mathcal{V}_i(u, B - c(q_i, u)) + V(q'_i, B_i - c(q_i, q'_i))$, then the utility function of the service provider becomes

$$h(Q') = \sum_{i \in [m]} \left(c(q_i, q'_i) - w_{q'_i, I_i} \right) + \beta \sum_{u \in V} p_a(u) \min_{i \in [m]} c(q'_i, u).$$

The algorithm follows by a reduction from the minimum pay-to-control problem to MFL. 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 $c'(q_i, q') = c(q_i, q') - w_{q', I_i}$,
(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 set of m vertices in F such that minimizes $C(Q') = \sum_{i \in m} c'(q_i, q'_i) + \beta \sum_{u \in V} p_a(u) \min_{i \in [m]} c'(q'_i, u)$.

Figure 4 shows an instance of 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.

Lemma IV.2. *Given an α-approximation algorithm for the MFL problem, the reduction above provides an α-approximation for the minimum pay-to-control problem.*

Proof. For any $Q' \subseteq F$, by the construction of the MFL instance, we have $h(Q') = \sigma C(Q')$. 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. Therefore, $h(Q') = \sigma C(Q') \leq \alpha \sigma \min_{Q^* \subseteq F} C(Q^*) = \alpha \min_{Q^* \subseteq F} h(Q^*)$. $\square$

By the result of Lemma IV.2, the $3 + o(1)$-approximation algorithm for the MFL problem in [20] applies to the minimum pay-to-control problem.

V. Simulation Results

We evaluate the performance of the two proposed indirect controls on ride-sourcing data from Uber [21]. The data set consists of the pick-up time and locations from April to September 2014 in New York City, primarily Manhattan. To reduce the complexity of the large data set with 914 pick-up locations, we cluster the close pick-up locations into 125 clusters such that no two pick-up locations in a cluster are farther than 500 meters apart. The drop-off location for each ride is selected from the same set of clusters with equal probability. Figure 5 shows the clustered pick-up locations and the arrival rates for ride requests at each cluster is represented with a bar. The performance of the proposed control methods are evaluated in two scenarios: 1) the initial location of the drivers are selected uniformly randomly, and 2) jammed scenario where the drivers are initialized at 20 closest locations to the Rockefeller center in Manhattan.

Observe that the proposed algorithms to find the controls are applicable to various driver models for $\mathcal{V}$. In the following section, we propose a behavior model $\mathcal{V}$ for the drivers.

A. Drivers' model

The driver model $\mathcal{V}$ is a function assigning an expected profit to each vertex at each time instance. Suppose driver i is located at vertex u. If the driver is the kth closest to a vertex v among the drivers in I_i, then in the model, the driver expects it will wait for k arrivals at v before servicing a request (i.e., driver i is the kth driver in a queue for v). Its expected profit

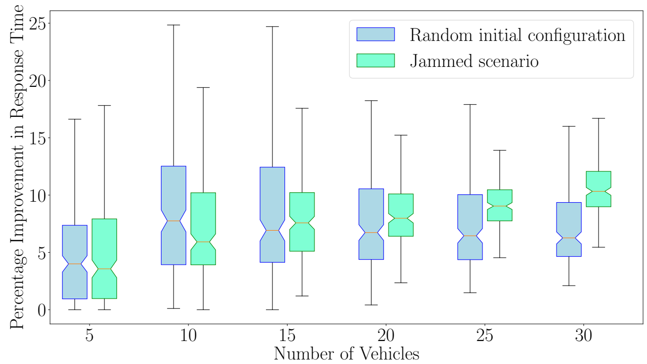


Fig. 6: Improved expected response time via the information sharing control.

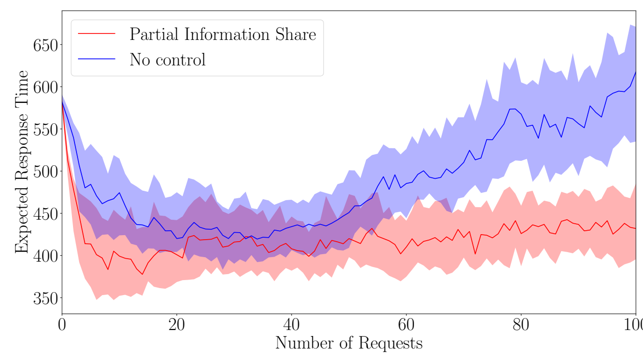


Fig. 7: The expected response time of a system of 20 drivers executing 100 ride requests arriving over time. The drivers are initialized under the jammed scenario.

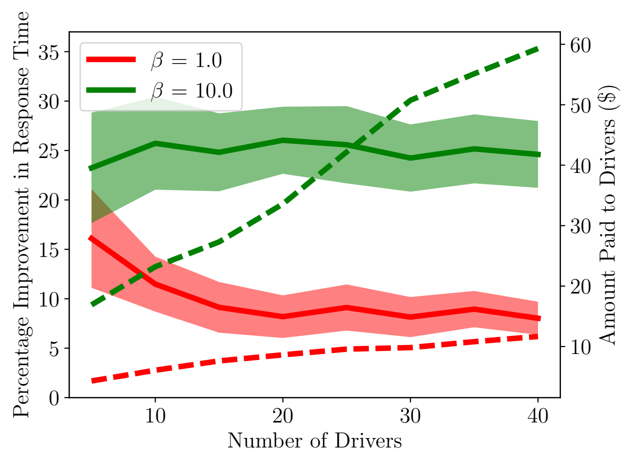


Fig. 8: The percentage improvement in expected response time and the incentive pay. The solid lines represent the average improvement in the expected response time and the dashed lines represent the total amount paid to the drivers.

is then calculated by probabilistically considering which queue it will reach the front of first, and the expected profit when servicing a ride at that vertex along with the expected future profit from that ride's dropoff location. Due to space constraints, we omit a detailed description of the driver model and refer the reader to the extended version of this paper [22], which contains all further details.

B. Partial Information Sharing

Figure 6 shows the percentage improvement in the expected response time for different number of vehicles using the partial information sharing control method of Section III in the two scenarios. Observe that as the number of drivers increases, the average improvement in the expected response time increases. However, with a large number of drivers randomly placed in the environment, the expected response time decreases and the possibility to further optimize it with information sharing is limited. On the other hand, in the jammed scenario where the drivers are concentrated in an area, the information sharing method improves the expected response time by 10% on average. The information sharing method shares information on the position of the other drivers with 31.0% and 41.7% of the drivers in the random initial configuration and jammed scenarios, respectively. The results are the average of 1000 instances for different number of drivers and scenarios. The boxes show the first, first and third quartiles of each set of experiments. The expected response time of the solution obtained from the LP-rounding algorithm of Section III on this set of experiments is within 0.014% on average of the solution of the LP relaxation of the information sharing problem. The maximum deviation from the optimal solution of the LP relaxation is 0.42%.

Figure 7 shows the expected response time of 20 drivers responding to 100 requests arriving over time with the jammed initial configuration. In this experiment, the maximum arrival rate on the vertices is 0.03 per minute, therefore, we assumed that there exist enough time to relocate between the ride request arrivals. Note that the information sharing method maintains a low expected response time over the course of responding to 100 requests compared to the same set of drivers with no control on their waiting locations. The results are the average of 100 experiments with 100 random requests for each experiment. The lines represent the average and shaded areas represent the first and third quartiles.

C. Pay to Control

In this section, we evaluate the performance of the PAY-TO-CONTROL method for the two scenarios. Figure 8 illustrates the improvement in the expected response (solid lines) time and the total amount paid to the drivers (dashed lines) to relocate for different β values in the random initial configuration scenario. The shaded area represents the first and third quartiles of 1000 random instances for each number of vehicles. The amount paid is proportional to the average cost of riding UberXL, i.e., \$0.3 per minute [23]. For larger β, the expected response time is more important than the amount paid for relocation. Therefore, with a larger number of vehicles, the PAY-TO-CONTROL method increases the amount paid to the drivers to minimize the expected response time. Notice that with $\beta = 10$ the expected response time has improved by 25% for \$1 per driver.

Next, we evaluate the performance of the pay-to-control algorithm in the jammed scenario. Figure 9 shows that with a larger number of vehicles concentrated in a small area,

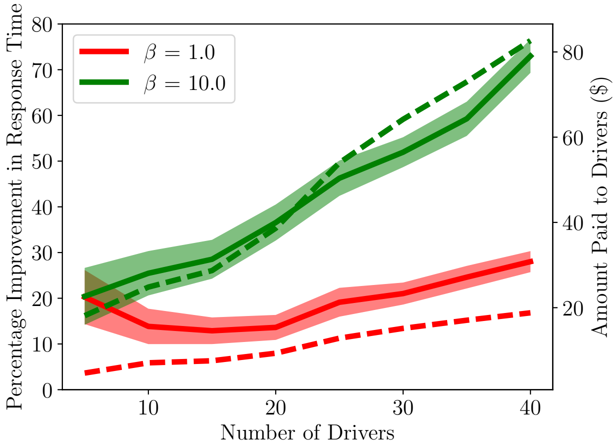


Fig. 9: The percentage improvement in the expected response time and the incentive pay in the jammed scenario. The solid lines represent the average improvement in the expected response time and the dashed lines represent the total amount paid to the drivers.

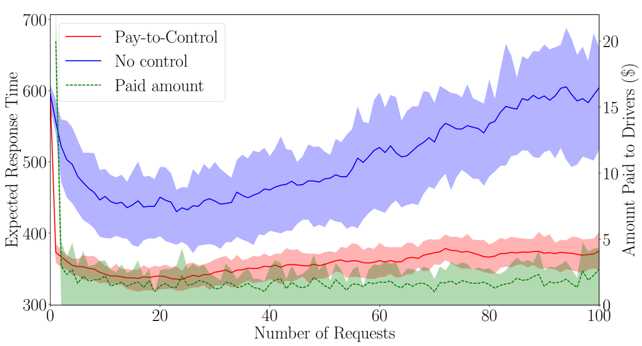


Fig. 10: The expected response time and the paid amount to 20 drivers, initialized under the jammed scenario, servicing 100 ride requests arriving over time under the PAY-TO-CONTROL method.

the pay-to-control algorithm improves the expected response time significantly with limited amounts paid to the drivers. Notice that with $\beta = 10$ the expected response time has improved by 70% for $2 per driver.

Figure 10 shows the expected response time and the amount paid to a set of 20 drivers responding to 100 requests arriving over time with the jammed initial configuration. Notice that the PAY-TO-CONTROL method with $\beta = 1$ maintains a low expected response time over the course of responding to 100 requests compared to the same set of drivers with no control input on their waiting locations. The total amount paid to the drivers over the course of responding to 100 requests is $\$1.87$ per request. The results are an average of 100 experiments with 100 randomly generated requests for each experiment. The lines represent the average and shaded areas represent the first and third quartiles.

VI. CONCLUSION

This paper considered the problem of controlling self-interested drivers in ride-sourcing 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 response time on real-world ride-sourcing data. In addition, we hope to extend the results to capture vehicles with different capacities and 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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Uber. Driving with Uber, wait less, earn more. [Online]. Available: https://www.uber.com/info/get-trips-without-waiting/

[9] 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.

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

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

[20] 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.

[21] (2014) Uber TLC FOIL response. [Online]. Available: https://github.com/fivethirtyeight/uber-tlc-foil-response

[22] A. Sadeghi and S. L. Smith, "On re-balancing self-interested agents in ride-sourcing transportation networks," *arXiv preprint arXiv:1909.04615*, 2019.

[23] Ride sharing driver. How much does uber cost? Uber fare estimator. [Online]. Available: https://www.ridesharingdriver.com/