New England University Transportation Center
77 Massachusetts Avenue, E40-279
Cambridge, MA 02139
utc.mit.edu

Year 25 Final Report

Grant Number: DTRT13-G-UTC31

Project Title:

Modeling the Impact of One-Way Car Sharing: An integrated data- and optimization-driven approach

Project Number:	Project End Date:	Submission Date:
MITR25-6	5/31/2016	5/24/2017

Principal Investigator Emilio Frazzoli
Title: Professor
University: MIT
Email: frazzoli@mit.edu
Phone: +1 617 852 2360

Co-Principal Investigator
Title:
University:
Email:
Phone:

Modeling the Impact of One-Way Car Sharing: An integrated data- and optimization-driven approach

The main objective of this project was to identify and evaluate alternative approaches for next-generation one-way car-sharing systems, and their effect on personal transportation in urban areas. What are the economic viability and potential social impacts of different forms of car-sharing systems, and of their operations, in scenarios of interest? (e.g., American metropolitan region, smaller cities, suburban developments, Asian or South American mega-city, but also university campus, industrial complex, etc.) In particular, in this project we focused primarily on New York City, based on publicly-available data, but the methodology will have general applicability.

We considered two spatial queueing-theoretical models for car-sharing systems that capture salient dynamic and stochastic features of customer demand. A spatial queueing model entails an exogenous dynamical process that generates “transportation requests” at spatially localized queues. Specifically, the first model, referred to as the “distributed” model, transforms the problem of controlling a set of spatially localized queues into one of controlling a single “spatially-averaged” queue and al- lows the determination of analytic scaling laws that can be used to select important system parameters (e.g., fleet size). The second model, referred to as the “lumped” model, exploits the theory of Jackson networks and allows the computation of key performance metrics and the design of system-wide coordination algorithms. Using techniques from receding horizon control, combinatorial optimization, and integer programming, we design control strategies ensuring the stability of the system.

In early work we made an initial contribution to ride-sharing by considering a MoD system, in which a shared fleet of vehicles, each capable of carrying two passengers at a time, is used to transport passengers. Inherent to the formulation are two important attributes: (i) the need to rebalance empty vehicles and (ii) the ability to identify lucrative ridesharing corridors by means of trip chaining. Note that although the later functionality is essential to capture ridesharing in its most general form, it is absent from the majority of existing works that, for a variety of reasons, limit the extent to which rides may be shared. We present a mixed-integer linear programming (MILP) formulation of the problem and show how an heuristic (feasible) solution to the problem can be obtained in polynomial-time by independently solving the ride-matching and rebalancing problems. This approximate solution can be used as a initial guess when solving the coupled problem via a MILP solver.

In the most recent work we considered an approach to uncover ridepooling opportunities in a shared-vehicle, mobility-on-demand system. In this setup, a fleet of self-driving vehicles, each capable of serving up to two passenger requests at a time, is used to trans- port passengers. In line with existing work, a station based model is used to represent the movement of passengers and vehicles in an urban environment. The problem of how to best route vehicles, to minimizes either (i) the total system travel time, (ii) total passenger travel time, or (iii) a weighted combination of the two, is given as the solution of a integer linear program. The formulation captures the need to rebalance vehicles from popular destination stations to popular origin stations. Moreover, it allows trips to be chained together, such that a passenger may board a vehicle with one passenger already on board. Theoretical results address the stability of such systems in terms of fleet size and characteristics of the travel demand. Finally, a simulation of a hypothetical shared mobility system, using real taxi trip data from NYC, demonstrates the efficiency gains possible by combining trips using our routing policy.

A copy of a paper summarizing all the finding of this project, submitted to the 2017 Intelligent Transportation Systems Conference is included in this report.

Ridepooling with Trip-Chaining in a Shared-Vehicle Mobility-on-Demand System

Samitha Samaranayake[a], Kevin Spieser, Harshith Guntha[b] and Emilio Frazzoli[c]

***Abstract*— This work considers an approach to uncover ridepooling opportunities in a shared-vehicle, mobility-on-demand system. In this setup, a fleet of self-driving vehicles, each capable of serving up to two passenger requests at a time, is used to transport passengers. In line with existing work, a station based model is used to represent the movement of passengers and vehicles in an urban environment. The problem of how to best route vehicles, to minimizes either (i) the total system travel time, (ii) total passenger travel time, or (iii) a weighted combination of the two, is given as the solution of a integer linear program. The formulation captures the need to rebalance vehicles from popular destination stations to popular origin stations. Moreover, it allows trips to be chained together, such that a passenger may board a vehicle with one passenger already on board. Theoretical results address the stability of such systems in terms of fleet size and characteristics of the travel demand. Finally, a simulation of a hypothetical shared mobility system, using real taxi trip data from NYC, demonstrates the efficiency gains possible by combining trips using our routing policy.**

I. Introduction

The recent emergence of ride-hailing services (e.g., Uber, Lyft, Via etc.) that use a fleet of vehicles to satisfy passenger requests on-demand, have made a significant impact on the urban transportation eco-system. Using a cellphone app, users may request rides, typically from their current location to their destination, and service is generally punctual. In this way, vehicle sharing provides a service comparable to owning a private car, while removing many of the more burdensome aspects of ownership, e.g., searching for a parking space. Moreover, by collectively time-sharing use of the fleet, significant fixed costs may be distributed over a large user base, drastically reducing the cost to access mobility.

Although ride-hailing services offer a number of benefits, they are plagued by an endemic inefficiency. Namely, empty vehicles must be rebalanced to ensure the supply of vacant vehicles remains aligned with the demand for transport. Compared to the private car ownership model, rebalancing has the unfortunate side-effect of increasing the total vehicle mileage driven throughout the system[1]. This raises concerns about worsening congestion on city streets. Recognizing that many vehicles can carry two or more passengers, one way to curb this effect is to incorporate ridepooling into the service, permitting vehicles to transport multiple passengers simultaneously. Note that neither the origins nor the destinations of passengers collocated in a vehicle need be the same. Services such as uberPOOL, Lyft Line, and Via speak to, relatively recent, commercial interest in building a ride-hailing platform capable of leveraging the efficiency gains of ridepooling.

Despite progress in resolving the core issues of ride-hailing and carpooling independently, the associated results rarely extend to services that fuse ride-hailing and carpooling functionality. For example, much of the carpooling literature does not apply to ride-hailing systems because it assumes private vehicle ownership and avoids the issue of how to rebalance empty vehicles in a fleet setting. At the same time, much of the ride-hailing work assumes unit capacity vehicles. This work considers an autonomous mobility-on-demand (AMoD) system, in which a shared fleet of self-driving vehicles, each capable of servicing two passenger requests at a time[2], is used to transport passengers. Inherent to the formulation are two important attributes: (i) the need to rebalance empty vehicles and (ii) the ability to identify lucrative ridesharing corridors by means of trip chaining. Note that although the later functionality is essential to capture ridesharing in its most general form, it is absent from the majority of existing works that, for a variety of reasons, limit the extent to which rides may be shared.

This remainder of this paper is organized in sections. Section II reviews the relevant literature. The notation and terminology used describe an AMoD system with ridepooling is provided in Section III. Section IV describes a steady-state model for an AMoD system with ridepooling that incorporates both rebalancing and trip-chaining. The analytical results of the paper are contained in Section V, where the ramifications of ridesharing are discussed in the context of system performance and stability. In Section VI, the aforementioned model

[a]School of Civil and Environmental Engineering, Cornell University, samitha@cornell.edu

[b]Department of Electrical Engineering, Indian Institute of Technology, Madras

[c]Department Aeronautics & Astronautics and Laboratory for Information and Decision Systems, MIT

[1]Not accounting for the additional distance traveled looking for parking, which admittedly can be large in certain situations.

[2]The ideas presented can be extended to the case of vehicles capable of carrying three or four passengers, with the solution complexity increasing by a factor of N per increase in occupancy, where N is the number of stations in the network.

is used to develop routing policies that are optimal with respect to (i) average passenger travel time, (ii) average vehicle travel time, and (iii) a combination of the two. A simulation-based analysis using NYC Taxi data shows the tradeoff between these objectives and how the model compares to (i) single-capacity ride-hailing and (ii) ride-pooling in the absence of chaining. Finally, Section VII recaps the key ideas of the paper and enumerates a collection of future work items.

II. Related Work

Ridesharing research is unified in its ambition to maximize efficiency gains at the system-level by servicing multiple trips with a single vehicle. However, the techniques used to do so differ widely based on problem specifics. For example, methods are often tailored to reflect (i) the timescale on which the transportation demand is known, (ii) the passenger capacity of vehicles, and (iii) if vehicles are privately owned or belong to a shared fleet. A more detailed taxonomy of such systems can be found in [1]. Simulations using data from large cities indicate significant gains with respect to efficeincy metrics such as total vehicle miles driven and system throughput may be achieved with ride-pooling [2].

The lexicon of ridesharing continues to evolve in both academic and everyday usage. For example, when trips repeat frequently at known (typically daily) time scales, ride-pooling amounts to carpooling. Trip requests that are reported in a more haphazard fashion require more sophisticated, often automated, methods. Ridesharing is termed static when trips are conglomerated assuming a fully-known demand model. Note in this context, static does not imply time-invariant. Ridesharing is termed dynamic when travel demand is revealed in realtime and trips are planed in an online manner.

When private vehicles are used for ridesharing, the owner remains in the vehicle whenever it is moving. Private vehicle ridesharing schemes for static demand models are reported in [3]. In [4], the role of driver or rider associated with each demand is determined at runtime to minimize total vehicle mileage. Although ridesharing with private vehicles imposes restrictions, where vehicles end up is of no consequence. In contrast, operating a shared fleet requires routing both passenger-filled and empty vehicles. The process of realigning the supply of empty vehicles with the travel demand in shared vehicle systems is termed rebalancing, and is the topic of [5] and [6]. There has also been a growing literature studying AMoD systems from a queuing-theoretic perspective [7]. The core problems in ride-hailing and carpooling have historically been distinct [1], yet become highly coupled when customer trips are shared using a shared vehicle.

The remainder of this section highlights relevant contributions to the study of such systems. Software solutions that exploit ridesharing opportunities in a shared vehicle system under a dynamic model are discussed in [3], [8], and [9]. Demands appear in real time with specified windows for pickup and dropoff. The challenge in doing so lies in deciding which of potentially thousands of vehicles are viable candidates to accept an additional rider and further refining this list to uphold quality of service timing constraints. Because this is a challenging problem [10], solutions often rely on heuristics to make decisions in a timely manner. A door-to-door MoD service tacitly assumes passengers do not need to switch vehicles. Nevertheless, the impact of customer transfers on ridesharing, which is pertinent for multi-mode problems, is discussed [11].

For systems that service no more than two requests at a time with a single vehicle, efficient matching algorithms can be used to identify "optimal" (under some assumptions) trip pairings [2]. However, this approach does not allow for the chaining of trips and does not account for the rebalancing cost of the matching, components that need to be modeled when considering AMoD systems with ridesharing. Even under this model, for capacities of three or more, the reported matching algorithms are non-polynomial in the number of trips. Finally, in very recent work, [12], it was shown that an anytime optimal algorithm can solve large-scale ridepooling problems in real-time, backed by an experimental analysis for NYC.

In this work, we assume a deterministic steady state system where passengers and vehicles are modeled as discrete elements. While this problem can also be studied using a corresponding (more general) queuing theoretic framework, this initial study focuses on the simpler deterministic model. The contributions of this article are as follows. First, we present a model for solving the steady-state ridepooling problem with rebalancing and trip-chaining. Second, we provide some analytical bounds on the minimum fleet size required for the stability of the system, under policies that minimize either the vehicle miles traveled or passenger miles traveled. Finally, we evaluate the system using real world demand data from the NYC Taxi dataset. We compare system performance under different objective functions and compare our routing algorithm's performance against (i) single capacity ride-hailing and (ii) ridepooling without trip chaining.

III. Notation and Terminology

This section presents the notation and terminology used to describe an AMoD system with ridepooling. We begin by describing the general functionality of the AMoD system. The fixed infrastructure of an AMoD system is described by a network $G = (V, E)$ of fully connected stations. Here, $V = \{1, 2, \ldots, N\}$ is a set of N stations and $E \subset V^2$ is the set of station pairs such that $(i, j) \in E$, $\forall i \neq j$. The time required to travel from station i to j is $T_{ij} > 0$ and the distance that must be traveled to do so is $d_{ij} > 0$. For stations i and j that are not directly connected in the physical world, $ij \in E$ represents the shortest physical path between the stations.

Passengers enter and exit the system at stations, and are transported between stations by a fleet of m self-driving vehicles. Each vehicle is able to transport up to two passenger requests (potentially more than two passengers) at a time. Passengers that enter a vehicle may depart the vehicle only upon reaching their destination. This stipulation prevents passengers from being transferred among vehicles while en route, a known deterrent to the adoption of transit systems [13][3]. Vehicles traveling while servicing zero, one, or two passenger requests are referred to as empty or rebalancing, single-occupancy, and dual-occupancy vehicles, respectively. Single occupancy vehicles may be diverted to pickup a second passenger request. We wish to find optimal vehicle and passenger flows (for di erent optimality criteria) by computing the corresponding rebalancing, single-occupancy, and dual-occupancy flows along each $ij \in E$. As mentioned previously, a deterministic steady-state model is used to analyze the movement of passengers and vehicles through G. The variables of interest are described below.

- x_i^k is the rate of empty vehicles traveling from node i to node $k \in V \setminus i$. The restriction $k \neq i$ prevents the senseless case of a station sending empty vehicles to itself.
- y_i^k is the rate of single-occupancy vehicles transporting a passenger from node i to that passenger's destination node $k \in V \setminus i$.
- $\tilde{y}_i^{jk}$ is the rate of single occupancy vehicles en route from i to j with a passenger destined for k.
- $z_i^{k,k'}$ is the rate of dual-occupancy vehicles traveling from node i to node k, for which one of the passengers is destined for node $k \in V \setminus i$, and the other passenger is destined for node $k' \in V \setminus i$. Note that $k = k'$ is permitted as it corresponding to the case where both passengers are traveling to the same destination.
- λ_i^k is the rate of passengers arriving at node i, destined for node k. The set $\{\lambda_i^k : i,\ k \neq i \in V\}$ is referred to as the travel demand of the system. For convenience, we write $\lambda_i = \sum_k \lambda_i^k$, $\lambda^k = \sum_i \lambda_i^k$, and $\lambda = \sum_i \lambda_i = \sum_k \lambda^k$.
- d_{ij} is the distance of link $ij \in E$. The average distance of all requests is $\bar{d}_{od} = \frac{1}{\lambda} \sum_{i,k} \lambda_i^k \cdot d_{ik}$.

Important quantities associated with the AMoD system may be expressed in terms of the above variables and previously mentioned system parameters. At this time, we list two quantities that will feature prominently in subsequent sections. The total number of vehicles in the system, denoted N_V, is given by

$$N_V = \sum_{i \in V} \sum_{k \in V \setminus i} T_{ik} \big(x_i^k + y_i^k + \sum_{j \in V \setminus \{i,k\}} \tilde{y}_i^{jk} + \sum_{k' \in V \setminus i} z_i^{kk'} \big). \quad (1)$$

Keeping in mind that each vehicle in the flow $z_i^{k,k'}$ contains two requests, if the total requests entering the system in the time period of concern is $\tilde{\lambda}$, the average time a passenger spends in the system, denoted $\bar{T}_S$, is,

$$\bar{T}_S = \frac{1}{\tilde{\lambda}} \sum_{i \in V} \sum_{k \in V \setminus i} T_{ik} \big(y_i^k + \sum_{j \in V \setminus \{i,k\}} \tilde{y}_i^{jk} + 2 \sum_{k' \in V \setminus i} z_i^{kk'} \big). \quad (2)$$

Equipped with a common vocabulary, the next sections consider the relationships and tradeo s that exist in a deterministic AMoD system with ridepooling under steady-state conditions.

IV. A Steady-State Model of Ridepooling in AMoD Systems

In reality, the passenger demand in an AMoD system, as well as the corresponding passenger and vehicle movements, will vary with time. However, in this work, we first focus on understanding ridepooling in an AMoD system assuming steady-state conditions. The idea being that any insight gained in this context will streamline analysis in the dynamic setting[4].

In steady state, neither passengers nor vehicles accumulate within the system. Consequently, for each node i, the rate at which vehicles enter i must equal the rate at which vehicles leave i, i.e., $\forall i \in V$,

$$\sum_{j \in V \setminus i} \big(x_j^i + y_j^i + \sum_{k \in V \setminus j} (z_j^{ik} + \tilde{y}_j^{ik}) \big) = \sum_{j \in V \setminus i} \big(x_i^j + y_i^j + \sum_{k \in V \setminus i} (z_i^{jk} + \tilde{y}_i^{jk}) \big). \quad (3)$$

Similarly, for each node i, the passengers entering the system, exiting the system, and just passing through must be conserved, i.e., $\forall i \in V,\ k \in V \setminus i$, we have

$$\lambda_i^k + \sum_{j \in V \setminus i} (z_j^{ik} + \tilde{y}_j^{ik}) = y_i^k + \sum_{j \in V \setminus \{i,k\}} \tilde{y}_i^{jk} + \sum_{k' \in V \setminus \{i,k\}} \Big(z_i^{kk'} + z_i^{k'k} \Big) + 2 z_i^{kk}. \quad (4)$$

We remark that in any sensible routing strategy, z_k^{ik} is zero, because there is no reason for a passenger destined for and already present at k to go on an excursion to i.

We also require that the variables x, y, z and consequently the inputs λ are integer valued, since the system will otherwise lead to trivial non-physical solutions where each capacity two vehicle is cut in half to create two capacity one vehicles. Finally, as per our earlier restriction, we do not want passengers to be transferred between vehicles while en route to their destination. Let

$$\kappa_i^k = \sum_{j \in V \setminus i} (z_j^{ik} + \tilde{y}_j^{ik}). \quad (5)$$

The following condition enforces this requirement:

$$z_i^{kk} \leq \begin{cases} \kappa_i^k + \frac{1}{2}(\lambda_i^k - \kappa_i^k) & \text{, if } \lambda_i^k \geq \kappa_i^k \\ \lambda_i^k & \text{, otherwise} \end{cases}, \quad (6)$$

[3]This restriction is added to make the system more practically applicable and can be easily removed.

[4]We have already extended this work to the dynamic setting and obtained preliminary results, which will be provided in an extended version of this article.

$\forall i \in V,\ k \in V \setminus i$. If $\lambda_i^k \geq \kappa_i^k$, then z_i^{kk} is maximized when as many of the λ_i^k demands get paired with the incoming passengers heading to k and the remaining $\lambda_i^k - \kappa_i^k$ share rides in pairs. If $\lambda_i^k < \kappa_i^k$, then at most λ_i^k vehicles will have two passengers heading to k. Conveniently, (6) may be written as the following pair of inequalities:

$$z_i^{kk} \leq \lambda_i^k \tag{7}$$

$$z_i^{kk} \leq \frac{\lambda_i^k}{2} + \frac{1}{2}\kappa_i^k. \tag{8}$$

Because the number of flow variables originating at node i, i.e., x_i^k, y_i^k, and z_i^{jk}, scales as $O(N+N+N^2)$, where $|V| = N$, the total number of variables in the system scales as $O(N^3)$. In practice, however, it is unlikely that the variables z_i^{jk} need to exist for all $(i, jk) \in V \times E$. For example, in NYC, $z_i^{jk} = 0$ for the case where i is in Midtown Manhattan, j is in the Bronx, and k is in Lower Manhattan. Therefore, in practice, the number of variables is likely to obey a more favorable scaling, which we exploit in our numerical experiments.

In the steady-state setting, e cient ridepooling amounts to finding vehicles flows that satisfy (3)–(6) and optimize a sensible performance metric. Identifying optimal flows is the subject of Section VI. As an interlude, the next section describes some of the theoretical consequences of (3)–(6).

V. Analysis of Ridepooling in AMoD Systems

In the case of ride-hailing without pooling, a steady-state formulation mandates arriving passengers be greeted by a car and transported to their destination along the fastest route. Routing then assumes the single-minded objective of minimizing N_V or, equivalently, empty vehicle miles traveled [5]. The fundamental *extra work* of rebalancing can also be related to the Earth Movers Distance [14], an established quantity that may be expressed in terms of origin and destination demands (λ_o, λ_d)[5].

In ridepooling, establishing a notion of fundamental extra work is nebulous because ridepooling permits vehicles to be deflected from the fastest path in favor of longer routes that promote sharing and, in turn, smaller fleet sizes. Accordingly, we advocate the correct framework to understand performance limits in ridepooling systems is in terms of the $(N_V, \bar{T}_S)$–tradeo curve obtained by minimizing the objective $\alpha N_V + (1-\alpha)\bar{T}_S$, subject to (3), (4), and (6), over a spectrum of α in $[0,1]$.

We reiterate that this tradeo between N_V and $\bar{T}_S$ does not materialize in the case of systems without ridepooling. Characterizing the $(N_V, \bar{T}_S)$-curve for select (λ_o, λ_d) can be done by solving the corresponding minimization problem, and is straightforward. In general, however, describing or even bounding the curve, as a function of (λ_o, λ_d), has proven challenging. Given these challenges and the preliminary nature of this work, we devote the following section to a numeric study of the $N_V - \bar{T}_S$ curve for a hypothetical ride-sharing system based on actual transport data, using both our model and prior solutions. The remainder of this section presents a handful of ridepooling results that hold irrespective of (λ_o, λ_d).

Intuitively, AMoD systems with ridepooling require a number of cars, N_V, no greater than what would be required for a system where pooling is not allowed. To explore the extent to which N_V can be reduced via ridepooling, we first provide a series of stability results relating N_V to properties of the travel demand and the desired objective.

A. Earth Mover's Distance

In ride-hailing systems with time-invariant pickup and dropo distributions φ_o and φ_d, respectively, $\mathrm{EMD}(\varphi_o, \varphi_d)$ is a lower bound on the minimum distance a vehicle must travel, on average, to transition from one job to the next [14]. Formally, given a set X and two distributions φ_1 and φ_2, where $\varphi_i : X \to \mathbb{R}_{\geq 0}$ for $i = 1, 2$, and a distance metric D on X, the Earth Mover's Distance, denoted $\mathrm{EMD}(\varphi_1, \varphi_2)$, is the first Wasserstein distance [15]. Mathematically,

$$\mathrm{EMD}(\varphi_1, \varphi_2) = \inf_{\gamma \in \Gamma(\varphi_1, \varphi_2)} \int_{X \times X} D(x_1, x_2)\, d\gamma(x_1, x_2), \tag{9}$$

where (φ_1, φ_2) is the set of all measures with marginals φ_1 and φ_2 on the first and second factor, respectively. Informally, if φ_1 and φ_2 represent two piles of "dirt" (i.e., earth), then $\mathrm{EMD}(\varphi_1, \varphi_2)$ is the minimum work (dirt $\times$ distance) required to reshape φ_1 into φ_2, or vice versa.

For conciseness, and compatibility with the standard definition of EMD, we provide the results with respect to constant vehicle velocities. The results, however, can be generalized to variable speeds by defining a corresponding Earth Mover's Time (EMT) instead of distance.

B. Performance Limits for a AMoD System with Ridepooling

We will work from the following definition of stability.

Definition 1: Consider an AMoD system. Let $Q_i(t)$ denote the number of passengers waiting for a vehicle at station i at time t and $Q(t) = \sum_{i \in V} Q_i(t)$. The system is *stable* if there exists a constant $M < \infty$ such that

$$Q(0) \leq \infty \Rightarrow Q(t) \leq M,\ t \geq 0. \tag{10}$$

In other words, stability requires the number of passengers waiting at stations be bounded at all times. A routing policy, π, is stable if it results in a stable AMoD system.

Proposition 1: Let $\pi \in\ _N$ be a stable routing policy that minimizes N_V, and $\bar{d}_{od}$ be the average distance of

[5] (λ_o, λ_d) are vectors that specify the total demand leaving each station and entering each station respectively.

each request. In the fluid limit of the system, the fleet size $m(\pi)$ satisfies

$$\frac{\lambda}{2v} \leq \frac{m(\pi)}{\left(\bar{d}_{od} + EMD(\lambda_o, \lambda_d)\right)} \leq \frac{\lambda}{v}. \tag{11}$$

Proof: Recall the the EMD specifies the minimum amount of work that needs to be done when rebalancing vehicles for demand and supply distributions (λ_o, λ_d). In the case without ridepooling where each vehicle transports at most one passenger at time, we have:

$$\lambda\left(\bar{d}_{od} + EMD(\lambda_o, \lambda_d)\right) \leq mv, \tag{12}$$

This is a restatement of the result in [14] that holds regardless of the routing strategy used and whether it is a fluid limit or the actual discrete system. It corresponds to the case where no ridepooling is performed and each vehicle transports at most one passenger at time. In the case of an optimal rebalancing strategy [5] and the fluid limit, this inequality holds with equality[6]. Therefore, when an optimal routing strategy is used, $(\lambda/v)\left(\bar{d}_{od} + EMD(\lambda_o, \lambda_d)\right)$ gives an upper bound on the required fleet size for the fluid system.

At the alternate extreme, the required fleet size is minimized when each passenger can always share their ride with another passenger without increasing the total distance traveled. This occurs either when every request can be paired up with another rider that has the same origin and destination (either because all the demands come in pairs or if passengers can wait for the next request with the same origin and destination). In this case, the total work required per request is halved. The lefthand side of (13) corresponds to this alternate extreme and provides a lower bound on the fleet size required for system stability. This bound holds regardless of whether we consider the fluid limit or the actual system. Unsurprisingly, doubling the capacity of vehicles halves the lower bound of the necessary fleet size. ■

Proposition 2: Let $\pi \in \;_T$ be a stable routing policy that minimizes T_S. In the fluid system, there exists a $\pi \in \;_S$ such that the fleet size $m(\pi)$ satisfies

$$\frac{\lambda}{2v} \leq \frac{m(\pi)}{\left(\bar{d}_{od} + EMD(\lambda_o, \lambda_d)\right)} \leq \frac{\lambda}{v}. \tag{13}$$

Proof: The solution above in which each passenger is transported individually gives us on the upper bound on the fleet size, since $\bar{T}_S$ is clearly minimized in this case. The lower bound is achieved when each passenger shares their ride with another passenger without having to deviate from the shortest path. Thus, the lower bound also remains the same. Furthermore, if any waiting time at the origin (prior to pickup) is ignored, one can trivially achieve the lower bound for fleet size by simply waiting and the origin until another demand with the same destination arrives. As long as this time is bounded, the system will still be stable. ■

[6]This simply requires solving a min cost bi-partite matching problem.

Unfortunately, bounding these quantities in the actual system without a fluid approximation is challenging, making it hard to quantify the tradeo between N_V and $\bar{T}_S$ in real-world systems. Therefore, we evaluate this tradeo experimentally in the next section.

VI. Simulations results

In this section, we show how the proposed ridepooling policies perform in di erent scenarios. We first compare this system against a single capacity ride-hailing system [16] and a system based on the matching algorithm presented in [2]. Then we experimentally explore the tradeo between fleet size and the average travel time. Our experimental results are based on a hypothetical AMoD installation in NYC using real demands obtained from NYC Taxi data. We first discuss our dataset and outline the experimental methodology, before describing the simulation results.

The demand model was populated using data from the open source NYC taxi data for 2013 [17]. The dataset contains trip level data with attributes such as origin location, destination location, pickup time and travel duration. After limiting the boundaries of the origin and destination to remain within lower Manhattan Island, the total number of trips made in the selected day (March 11, Monday) was nearly 39,306. Collectively, we assume that this demand is representative of an initial user base for the hypothetical ridepooling system.

To make this data compatible with the proposed model, origin and destination locations of all demands were clustered using a k-means algorithm and mapped to 40 stations [16]. To measure the model's performance through a day's data, the demands of the day were aggregated into 24 buckets corresponding to their hour of origin. Once aggregated, to make the demands more realistic and challenging for the model, the time window was compressed from 1 hour to 5 minutes ($\lambda_i^k = \lceil \lambda_i^k(\text{hour})/12 \rceil$). Thus, we obtain a steady state demand distribution with a 5 minute time step. The simulation assumes vehicles travel at a constant speed of 30km/h and move directly between stations. Upon mapping origin and destination points to the nearest station, and discarding trips that start and end the same node, we were left with 36,915 trips.

The proposed model results in an MILP that was implemented in Python and solved using the Gurobi Optimizer. The objectives of interest are to (1) minimize total fleet size (N_V), (2) minimize average customer travel time ($\bar{T}_S$), and (3) a convex combination of these objectives. The constraints are given by equations (3) – (6). Since the problem is an MILP, convergence is not always absolute. The largest MIP gap (di erence between lower and upper bounds in the solver) in our simulations is 6.22% with a maximum solver runtime of 10 minutes.

The first set of experiments were run with the objective of minimizing fleet size (N_V) for all hours of the selected day. We demonstrate the performance of the

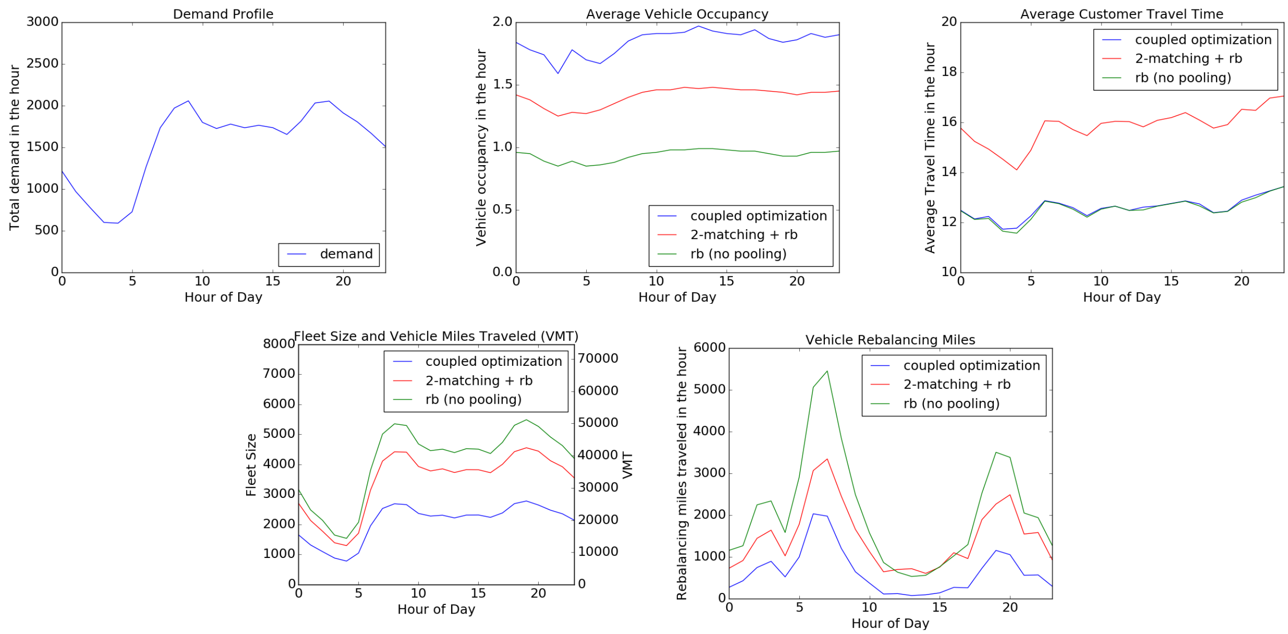

Fig. 1: Experimental results comparing three ridesharing models, 1) optimal rebalancing without ridepooling, (i.e., rb (no pooling)), 2) optimal ride matching with decoupled optimal rebalancing (i.e., 2-matching + rb), and 3) the new coupled optimization model (i.e., coupled optimization), with New York City taxi data over a 24 hour period.

α	|V|	X	Y	Z	VMT	X%	AVO	ATT
0*	1902	271	0	1631	17730	14.25	1.71	17.02
0	1921	202	143	1575	17904	10.53	1.71	12.98
0.1	1919	205	141	1574	17888	10.69	1.71	12.95
0.2	1919	200	157	1562	17891	10.40	1.71	12.93
0.3	1920	202	155	1563	17891	10.53	1.71	12.92
0.4	1918	192	178	1547	17875	10.03	1.70	12.90
0.5	1954	214	219	1521	18217	10.96	1.67	12.85
0.6	1945	211	205	1528	18125	10.86	1.67	12.85
0.7	1961	206	249	1506	18280	10.51	1.66	12.85
0.8	1945	201	226	1518	18129	10.34	1.67	12.85
0.9	1964	212	244	1508	18310	10.78	1.66	12.85
1	3805	542	3262	0	35461	14.25	0.85	12.85

(a) Hour 6

α	|V|	X	Y	Z	VMT	X%	AVO	ATT
0*	2197	29	0	2168	20476	1.30	1.97	16.06
0	2217	9	48	2160	20660	0.40	1.97	12.59
0.1	2215	5	56	2154	20642	0.21	1.97	12.58
0.2	2217	6	67	2143	20660	0.29	1.96	12.55
0.3	2219	7	66	2146	20683	0.31	1.96	12.56
0.4	2245	8	139	2098	20921	0.34	1.93	12.49
0.5	2253	8	156	2089	20998	0.34	1.92	12.49
0.6	2265	8	181	2077	21113	0.33	1.91	12.49
0.7	2256	11	154	2090	21023	0.50	1.92	12.49
0.8	2287	11	217	2058	21315	0.50	1.89	12.49
0.9	2285	10	216	2059	21299	0.44	1.90	12.49
1	4394	57	4337	0	40952	1.30	0.98	12.50*

(b) Hour 13

TABLE I: Tradeoff analysis between minimizing the total Vehicle Miles Traveled (VMT) and Average Passenger Travel Time (ATT) in minutes for two representative hours of the day. The objective function is $Z = (1-\alpha)Z_{VMT} + \alpha Z_{ATT}$. The column descriptions are as follows: $|V|$ is the fleet size, X is the #empty (or rebalancing) trips, Y is the #single occupancy trips, Z is the #dual occupancy trips, X% is the percentage of empty trips over all trips, and AVO is the average vehicle occupancy. The $\alpha = 1$ case for hour 13 is slightly larger than $\alpha = 0.9$ due to rounding errors.

model by comparing it with two alternate models. The first model assigns each request to a dedicated vehicle (no ridepooling) and minimizes the total empty vehicle (or rebalancing) miles as described in [5] and [16]. The second model, based on [2], solves an optimal matching problem to allocate requests to vehicles with a limit of two (as in our model). This algorithm, however, does not consider rebalancing the vehicle fleet and does not have the capability of trip chaining. The address the rebalancing issue, which is necessary for system stability, we use a sequential optimization approach where the optimal rebalancing is performance after the matching is done. The two stages of this approach in more detail are: 1. perform a maximum matching of the aggregated demands and pair them accordingly, and 2. rebalance the new trip set using the algorithm given in [16]. In contrast, our model is able to integrate the rebalancing problem with the matching problem and solve a coupled optimization problem. It should be noted, however, the the approach in [2] does not require a station based model and is more general in that sense[7].

Figure 1 shows the results obtained from comparing our model with the two models described above over a 24 hour period. As expected, the minimum required fleet size is proportional to the total demand in all three models with our coupled optimization approach requiring the smallest fleet size. The temporal variation of the fleet

[7]For a comparison with respect to non-station based fleet management we refer the reader to [12].

size is also minimized in our approach with the maximum required fleet size being almost half of the requirement with the decoupled 2-pooling approach. Note that the minimum required fleet size and total vehicle miles traveled are proportional to each other in the steady state models. The higher average vehicle occupancy in our model, made possible by the additional sharing due to trip chaining, leads to the smaller fleet requirement. By exploiting these trip chaining opportunities, our model is also able to significantly reduce the total number of empty vehicle (or rebalancing) miles in the system, which are non-revenue-generating. Somewhat surprisingly, for the case of this particular data set, the average travel in our approaches is almost as good as with no ridepooling. While theoretically, one would need to tradeo between VMT and passenger travel time, the trip chaining in our method is able to very e ectively exploit the overlapping nature of the trips in this dataset.

The second set of experimental results correspond to a tradeo analysis between two competing objectives of interest: (1) total Vehicle Miles Traveled (or total fleet size) Z_{VMT} and (2) Average Passenger Travel Time Z_{ATT}. We consider a spectrum of objectives that span the linear combinations of these two objectives with a parameter $\alpha \in [0,1]$, where $Z = (1-\alpha)Z_{VMT} + \alpha Z_{ATT}$. The corresponding results for two hours of the day (hours 6 and 13) are presented in Table 1. The extra data point labeled with $\alpha = 0^*$ is a special case of interest when all demands are forced to travel as coupled demands (i.e. as z_i^{kk} flows) by forcing each customer to wait at the source until a perfect match is found. As seen in Table 1, this leads to a significantly longer ATT, but does not reduce VMT by any significant amount compared to when $\alpha = 0$. The fleet size and VMT is minimized when $\alpha = 0$, but grows very slowly as α increases. Correspondingly the ATT also decreasing at a slow rate. In this case, the trip chaining algorithm is able to find e cient routes for demand and keep travel time increase at a minimum. For this dataset, with a fairly uniformly distributed demand, the sweet spot appears to be when α is at 0.9. We expect this point to be di erent across di erent datasets.

While not a viable framework for operating a real-time Mobility-on-Demand (MoD) system with ridepooling, the proposed model can be used as a fast and simple methodology for evaluating the potential for such systems. It can be used as a tool by both fleet operators and public agencies to evaluate the benefits of the simple case of capacity two ridepooling.

VII. Conclusions and Future Directions

This paper presented a steady state analysis for ridepooling with trip chaining in a Mobility-on-Demand (MoD) system. A new model that captures both the e ects of trip chaining and vehicle rebalancing was developed and used to discover more e cient ridepooling solutions. We show how the minimum fleet size for system stability, in the case of the fluid approximation of the system, can be bounded using the Earth Mover's Distance, both in the case of minimizing (i) fleet size and (ii) on-trip delay. The system imposes a fundamental tradeo between minimizing the total vehicle miles traveled and minimizing the total passenger miles traveled. While an analytical expression for this tradeo proved elusive, a detailed experimental analysis for Manhattan, based on real travel data from NYC taxis, was presented. The results show that our solution is more e cient than previously reported schemes and able to decrease vehicle miles traveled with very little impact on the average travel time. Our current work is limited to the case of a deterministic steady state system, but we are expanding this work to a) study the deterministic dynamic system and b) the stochastic steady state system.

References

[1] N. Agatz, A. Erera, M. Savelsbergh, and X. Wing, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, 2012.

[2] P. Santi, G. Resta, M. Szell, S. Sobolvesky, S. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *PNAS*, 2014.

[3] P. d'prey, R. Fernandes, and M. Ferreira, "Empirical evaluation of a dynamic and distributed taxi-sharing system," *Proc. of Conference on Intelligent Transportation Systems*, 2012.

[4] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Dynamic ride-sharing: A simulation study in metro atlanta," *Transportation Research Part B: Methodological*, 2012.

[5] M. Pavone, S. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *International Journal of Robotics Research*, 2012.

[6] D. Jorge and G. Correia, "Carsharing systems demand estimation and defined operations: a literature review," *European Journal of Transport and Infrastructure Research*, 2013.

[7] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: a queueing-theoretical perspective," *Proc. of Robotics: Science and Systems Conference*, July 2014.

[8] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large scale dynamic taxi ridesharing service," *Proc. of IEEE ICDE*, 2013.

[9] M. Horn, "Fleet scheduling and dispatching for demand-responsive passenger services," *Transportation Research Part C: Emerging Technologies*, 2002.

[10] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," *The European Journal of Operational Research*, 1990.

[11] B. Coltin and M. Velosa, "Scheduling for transfers in pickup & delivery problems with very large neighborhood search," *Proc. of Twenty-Eighth Conference on Artificial Intelligence*, 2014.

[12] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, 2017.

[13] Z. Guo and N. H. Wilson, "Assessing the cost of transfer inconvenience in public transport systems: A case study of the london underground," *Transportation Research Part A: Policy and Practice*, vol. 45, no. 2, pp. 91–104, 2011.

[14] K. Treleaven and M. Pavone, "Asymptotically optimal algorithms for pickup and delivery problems with application to large-scale transportation systems," *IEEE Transactions on Automatic Control*, 2013.

[15] L. Ruschendorf, "The Wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, 1985.

[16] K. Spieser, S. Samaranayake, W. Gruel, and E. Frazzoli, "Shared-vehicle mobility-on-demand systems: A fleet operator's guide to rebalancing empty vehicles," in *Transportation Research Board 95th Annual Meeting*, no. 16-5987, 2016.

[17] B. Donovan and D. Work, "New york city taxi trip data (2010-2013)," in *University of Illinois at Urbana-Champaign*, 2016.