

Analysis, Control, and Evaluation of Mobility-on-Demand Systems: a Queueing-Theoretical Approach

Rick Zhang, Federico Rossi, and Marco Pavone*

Abstract—This paper presents a queueing-theoretical approach to the analysis, control, and evaluation of mobility-on-demand (MoD) systems for urban personal transportation. A MoD system consists of a fleet of vehicles providing one-way car sharing service and a team of drivers to rebalance such vehicles. The drivers then rebalance themselves by driving select customers similar to a taxi service. We model the MoD system as two coupled closed Jackson networks with passenger loss. We show that the system can be approximately balanced by solving two decoupled linear programs and exactly balanced through nonlinear optimization. The rebalancing techniques are applied to a system sizing example using taxi data in three neighborhoods of Manhattan. Lastly, we formulate a real-time closed-loop rebalancing policy for drivers and perform case studies of two hypothetical MoD systems in Manhattan and Hangzhou, China. We show that the taxi demand in Manhattan can be met with the same number of vehicles in a MoD system, but only require 1/3 to 1/4 the number of drivers; in Hangzhou, where customer demand is highly unbalanced, higher driver-to-vehicle ratios are required to achieve good quality of service.

I. INTRODUCTION

Car sharing promises to be a cost effective and sustainable alternative to private urban mobility by allowing a split of hefty ownership costs, increasing vehicle utilization, and reducing urban infrastructure needed for parking [1]. One type of vehicle-sharing service, called mobility-on-demand (MoD), consists of stacks or racks of light electric vehicles parked at many different stations throughout a city [1]. Each customer arrives at a station, takes a vehicle to the desired destination, and drops off the vehicle at that station.

MoD systems have been advocated as a key step toward sustainable personal urban mobility in the 21st century [1]. They present, however, a number of operational challenges. In particular, due to the asymmetry of customer demands, vehicles tend to aggregate at some stations and be depleted elsewhere, causing the system to become *unbalanced* [2] and leading to an overall reduction in quality of service. Rebalancing approaches in car sharing systems are typically categorized into (1) user-based rebalancing and (2) operator-based rebalancing. User-based approaches typically introduce financial incentives to influence trip origins and destinations as well as encourage ride sharing or splitting [3]. However, these strategies typically cannot meet all of the rebalancing needs of the system, since it is difficult to model and

control user behavior [4]. Operator-based rebalancing, the main focus of this paper, involves sending hired drivers to different parts of the cities to rebalance the vehicles. Previous works on operator-based rebalancing strategies often formulate the problem as a mixed-integer linear program to maximize the profit generated by the system [5], [6] and subject to fixed rebalancing costs. However, these formulations do not directly account for the rebalancing of the drivers themselves, if, say, public transit is not readily available. The drivers may be rebalanced using shared shuttles [7] or by ferrying passengers to their destinations, much like a taxi service [8]. It is worth noting that rebalancing for MoD systems has also been studied in the context of autonomous vehicles under a fluidic model [2], a queueing network model [9], and a decentralized Gaussian Process-based model [10].

The objective of this paper is to develop a queueing-theoretical framework for the analysis, control, and evaluation of (human-driven, non-autonomous) mobility-on-demand systems. We then apply the insights from this queueing framework to develop real-time closed-loop policies to control such systems. On the modeling and analysis side, we consider a model similar to the one proposed in [8], where drivers are themselves rebalanced by driving a portion of the customers to their destinations. In this way, the MoD system can be viewed as a one-way customer-driven car sharing service mixed with a taxi service. The model presented in [8] hinges upon the optimization of rebalancing *rates* and is studied under a fluidic approximation (where customers, drivers, and vehicles are modeled as a continuum). While this model offers insights into the minimum number of vehicles and drivers required in a MoD system, it does not provide key performance metrics in terms of quality of service (i.e., the availability of vehicles at stations or the customer wait times). These shortcomings are addressed by [9] for an autonomous MoD system, where the system is modeled as a stochastic queueing network from which key performance metrics are derived. This paper can be viewed as an extension of the models in [8], [9] to human-driven MoD systems taking into account both vehicles and rebalancing drivers. On the control side, real-time closed-loop policies for one-way car sharing systems have been studied in [5] and [6] with the objective of maximizing profit, where the rebalancing of vehicles is modeled as a cost. Our paper differs from these works in two key respects: 1) in addition to minimizing cost, our key objective is quality of service for customers in terms of vehicle availabilities and wait times, and 2) we explicitly control the movement of rebalancing drivers which makes the system self-contained (e.g., drivers do not need to rely on public transit to rebalance themselves).

Our contribution in this paper is fourfold. First, we model a MoD system within a queueing-theoretical framework that takes into account the *coupled* rebalancing of vehicles and drivers. Specifically, our approach is to model a MoD system

Rick Zhang is with Zoox Inc., 325 Sharon Park Drive, Menlo Park, CA 94025, USA. Email: rick@zoox.com.

Federico Rossi and Marco Pavone are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA. Email: {frossi2,pavone}@stanford.edu.

This research was supported by the National Science Foundation under CAREER Award CMMI-1454737, by the Toyota Research Institute (TRI), and by the Dr. Cleve B. Moler Stanford Graduate Fellowship. Travel demand data is courtesy of the New York City Taxi and Limousine Commission and of Didi Chuxing. The authors would like to acknowledge insightful discussions with S. Beiker, E. Frazzoli, D. Rus, M. Schwager, and S. Smith.

* Corresponding author.

as two coupled closed Jackson networks with passenger loss. Second, we present two approaches for the open-loop control of a MoD system. In the first approach, the optimal rebalancing parameters are solved by two decoupled linear programs, and are therefore efficient to compute, but only approximately guarantee balance of the system. In the second approach, nonlinear optimization techniques are used (with higher computational cost) to balance the system exactly. Third, we apply such approaches to the problem of system sizing and test them on a case study in Manhattan, showing that the optimal vehicle-to-driver ratio in a MoD system deployed in the city should be between 3 and 5. Finally, leveraging the aforementioned open-loop control strategies, we devise a real-time *closed-loop* rebalancing policy and demonstrate its performance for case studies of two hypothetical MoD systems in Manhattan and Hangzhou, China. In particular, we show that a MoD system can satisfy all existing taxi demands in Manhattan with around the same number of vehicles as current taxis (approximately 11,000), but only needs 1/3 to 1/4 the number of drivers. In Hangzhou, where demand is highly unbalanced, we show that a higher driver-to-vehicle ratio of 1/3 to 1/2 is required to achieve good quality of service.

The rest of this paper proceeds as follows: Section II reviews some key results in the theory of Jackson networks. Section III describes in detail our queueing network model of a MoD system. Section IV offers the approaches for the open-loop control of a MoD system. The rebalancing techniques are then applied to a system sizing example based on taxi data in Manhattan. In Section V we introduce a real-time closed-loop control policy useful for practical systems, and in Section VI we study its performance on MoD systems in Manhattan and Hangzhou, China. Finally, in Section VII we draw our conclusions and provide directions for future research.

A preliminary version of this paper appeared as [11]. In this revised and extended version, we provide as additional contributions proofs of all results, an expanded description of the real-time closed-loop rebalancing algorithm, and two new case studies for MoD systems operating in Manhattan and in Hangzhou, China.

II. BACKGROUND MATERIAL

In this section we recall a few fundamental concepts from the theory of queueing networks, and in particular Jackson networks, that will be used throughout the rest of the paper. We refer the interested reader to [9], [12] for more in-depth discussions.

A Jackson network is a class of Markovian queueing networks whereby the routing distribution (the probability of transitioning to node j from node i), r_{ij} , is stationary and the service rate at each node i , $\mu_i(x_i)$, only depends on the number of agents at that node, x_i [12, p. 9]. In equilibrium, the throughput at each node (i.e., the average number of agents passing through the node per unit time), $\{\pi_i\}_{i=1}^{|V|}$, of a *closed* Jackson network (i.e., with a fixed number of agents moving among the nodes, with no external arrivals or departures) satisfies the balance equations

$$\pi_i = \sum_{j \in V} \pi_j r_{ji}, \quad \text{for all } i \in V. \quad (1)$$

Note that (1) does not yield a unique solution and only determines $\pi = (\pi_1 \ \pi_2 \ \dots \ \pi_{|V|})^T$ up to a constant factor.

Accordingly, π is referred to as the *relative* throughput. The stationary probability distribution of a closed Jackson network with m agents is given by

$$\mathbb{P}(x_1, x_2, \dots, x_{|V|}) = \frac{1}{G(m)} \prod_{j=1}^{|V|} \pi_j^{x_j} \prod_{n=1}^{x_j} \mu_j(n)^{-1},$$

where $G(m)$ is a normalization constant required to make $\mathbb{P}(x_1, x_2, \dots, x_{|V|})$ a probability measure. It turns out that many performance metrics of the network can be expressed in terms of the normalization constant $G(m)$. Two such performance metrics are of interest to us: 1) the actual throughput of each node (see [12, p. 27]) is given by

$$\Lambda_i(m) = \pi_i G(m-1)/G(m), \quad (2)$$

and 2) the probability that a node has at least one agent, referred to as the *availability* of node i ([13], [14]), is given by

$$A_i(m) = \gamma_i G(m-1)/G(m), \quad (3)$$

where $\gamma_i = \pi_i/\mu_i(1)$ is referred to as the relative utilization of node i . In general, solving for $G(m)$ is quite computationally expensive, especially when m is large. A well-known iterative technique called mean value analysis (MVA) [15] enables us to compute the mean values of performance metrics without explicitly solving for $G(m)$. The MVA algorithm is described in detail in [9], [16].

III. MODEL DESCRIPTION AND PROBLEM FORMULATION

A. MoD system model

In this section we formally describe the MoD system under consideration and cast it within a queueing network framework by modeling the system as two coupled, closed Jackson networks. We consider N stations with unlimited parking capacity placed in a given geographical area, m_v vehicles that can be rented by customers for one-way trips between stations, and m_d “rebalancing” drivers employed to rebalance the vehicles by driving them to the stations where they are needed. After rebalancing the vehicles, the drivers themselves become unbalanced – they need to get back to locations with an excess of vehicles. To “rebalance” the drivers, we propose a mechanism whereby the drivers drive a portion of customers to their destinations, effectively operating as a taxi service. This requires each driver to *always* have access to a vehicle since the driver’s task involves driving a vehicle with or without a customer. (A driver left at a station without a vehicle is effectively “stranded”.) We therefore pose the constraint $v_i \geq d_i$, where v_i is the number of vehicles at station i and d_i is the number of drivers at station i (note that in this framework, we do not allow multiple drivers to occupy the same vehicle). With this requirement, we may view the MoD system as two systems operating in parallel – a one-way customer-driven car sharing service with $m_v - m_d$ vehicles and a taxi service with m_d vehicles. It is worth noting that there are other, more elaborated ways of managing a MoD system which we do not address in this paper. For example, in [8], the authors also consider customers potentially riding with multiple drivers. One could also envision a system where drivers can drive other drivers or take public transportation to stations with excess cars. In these cases, a key challenge is the explicit modeling of the movement of the drivers, which could be represented using a separate queueing network. If drivers

can take other forms of transportation not explicitly modeled by the queueing network, it may be included as an additional rebalancing cost, similar to [5]. The extension of our model to such cases is an interesting avenue for future research.

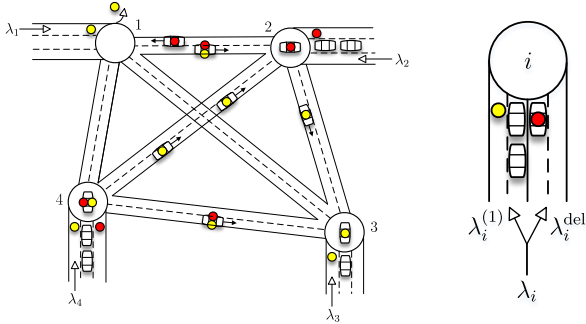


Fig. 1. Left: MoD system model. Yellow dots represent customers and red dots represent rebalancing drivers. Customers can drive themselves or ride with a rebalancing driver. Customers are lost if no vehicles are available (station 1). Right: each customer arriving at station i is delegated to either System 1 (customer-driven vehicles) or System 2 (taxi system).

Customers arrive to station i according to a Poisson process with parameter λ_i . Upon arrival at station i , the customer selects a destination j with probability p_{ij} , where $p_{ij} \geq 0$, $p_{ii} = 0$, and $\sum_j p_{ij} = 1$. Furthermore, we assume that the probabilities $\{p_{ij}\}_{ij}$ constitute an irreducible Markov chain. The customer can travel to his/her destination in one of two ways: 1) the customer drives a vehicle to his/her destination, or 2) the customer is taken to his/her destination by a rebalancing driver. The travel time from station i to station j is an exponentially distributed random variable with mean $T_{ij} > 0$. The assumptions of Poisson arrivals and exponential travel times not only simplify the problem, but have been shown to be reasonable approximations in terms of their predictive accuracy in similar spatial queueing models for vehicle routing [9]. We employ a “passenger loss” model similar to [9], [14], where if a vehicle is not available upon the arrival of a customer, the customer immediately leaves the system. However, due to the additional complexity of our MoD model (a one-way car sharing service and a taxi service in parallel) the passenger loss assumption is more involved. We assume that upon arrival at a station, a customer is delegated to one of the two parallel systems by the MoD service operator (see Fig. 1). The customer is lost if there are no available vehicles in the system to which he/she was delegated. For example, if a customer is delegated to the taxi system and no taxis are immediately available, the customer cannot switch over to the other system and drive himself/herself to the desired destination. This assumption is needed to maintain tractability in the Jackson network model. The modeling consequences of this assumption will be further discussed in the next section. The performance criterion of interest in this case is the probability a customer will find an available vehicle (both empty vehicles and taxis) at each station. In Section V we will relax the passenger loss assumption and investigate the more realistic scenario where customers form a queue to wait for available vehicles. The performance of the system is then measured by customer wait times.

B. Jackson network model of a MoD system

We now formally cast the MoD model described in the previous section within a queueing network framework. The key is to construct an abstract queueing network where the

stations are modeled as single-server (SS) nodes and the roads as infinite-server (IS) nodes, as done in [9], [14]. Vehicles form a queue at each SS node while waiting for customers and are “serviced” when a customer arrives. The vehicle then moves from the SS node to the IS node connecting the origin to the destination selected by the customer. After spending an exponentially distributed amount of time (with mean T_{ij}) in the IS node, the vehicle moves to the destination SS node. With this setup, we have described a *closed Jackson network with respect to the vehicles*. To capture the idea that the MoD system consists of two systems (customer-driven system and taxi system) operating in parallel, we model the MoD system as two coupled closed Jackson networks. More formally, let System 1 represent the Jackson network of $m_v - m_d$ customer-driven vehicles, and System 2 represent the network of m_d taxis. Let $S^{(k)}$ represent the set of SS nodes and $I^{(k)}$ represent the set of IS nodes in the k^{th} Jackson network, where $k = \{1, 2\}$. For each network, each SS node is connected to every other SS node through an IS node. Thus, each network consists of $N + N(N - 1) = N^2$ nodes (the IS node from station i to itself is not represented since $p_{ii} = 0$). For each IS node $i \in I^{(k)}$, let $\text{Parent}(i)$ and $\text{Child}(i)$ be the origin and destination of i , respectively. The routing matrix $\{r_{ij}^{(k)}\}_{ij}$ in Jackson network k can then be written as

$$r_{ij}^{(k)} = \begin{cases} p_{il}^{(k)} & i \in S^{(k)}, j \in I^{(k)}, i = \text{Parent}(j), l = \text{Child}(j), \\ 1 & i \in I^{(k)}, j \in S^{(k)}, j = \text{Child}(i), \\ 0 & \text{otherwise,} \end{cases}$$

where the first case is the movement from a SS node to an IS node and the second case is from an IS node to its unique destination SS node. The service times at each node are exponentially distributed with mean service rates

$$\mu_i^{(k)}(n) = \begin{cases} \lambda_i^{(k)} & \text{if } i \in S^{(k)}, \\ \frac{n}{T_{jl}} & \text{if } i \in I^{(k)}, j = \text{Parent}(i), l = \text{Child}(i), \end{cases}$$

where n is the number of vehicles in the IS node. With this formulation we have defined two closed Jackson networks of the same form as in [9], amenable to analysis. Fig. 2 illustrates the topology of a simple system cast as a closed Jackson network.

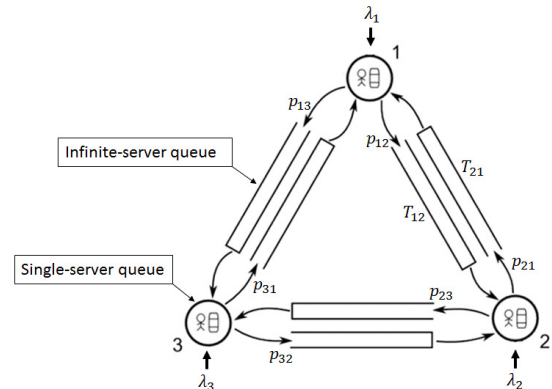


Fig. 2. A three-station system (System 1 or System 2) cast as a closed Jackson network. Circles represent SS nodes and rectangles represent IS nodes.

We now return to the customer arrival process and the loss model assumption. Recall that customers arrive at station i according to a Poisson process with rate λ_i . Upon arrival, customers at station i going to station j are split into either System 1 or System 2, with fixed probability. This can be seen

as a Bernoulli splitting of the customer arrival process into two Poisson processes for each desired destination. Denote by $\lambda_i^{(1)}$ the total rate of customers delegated to System 1, by $p_{ij}^{(1)}$ the routing probabilities associated with System 1 ($p_{ij}^{(1)} \geq 0$, $p_{ii}^{(1)} = 0$, $\sum_j p_{ij}^{(1)} = 1$), by λ_i^{del} the total rate of customers delegated to System 2, and by η_{ij} the routing probabilities associated with System 2. We have the relationship $\lambda_i = \lambda_i^{(1)} + \lambda_i^{\text{del}}$ for each station i . We define q_i to be the total fraction of customers delegated to System 1 at station i , i.e., $q_i := \lambda_i^{(1)}/\lambda_i$. We can also write $1 - q_i = \lambda_i^{\text{del}}/\lambda_i$. The routing probabilities for the customers are then split up as follows

$$\begin{aligned} p_{ij} &= \mathbb{P}(i \rightarrow j \mid \text{System 1}) q_i + \mathbb{P}(i \rightarrow j \mid \text{System 2}) (1 - q_i) \\ &= p_{ij}^{(1)} q_i + \eta_{ij} (1 - q_i). \end{aligned} \quad (4)$$

We can equivalently say that the Poisson rate of customers originating at station i and headed for station j is $\lambda_i p_{ij}$. The arrival rate of these customers to System 1 is then $\lambda_i^{(1)} p_{ij}^{(1)}$ and the arrival rate to System 2 is $\lambda_i^{\text{del}} \eta_{ij}$. Thus relation (4) can be rewritten as

$$\lambda_i p_{ij} = \lambda_i^{(1)} p_{ij}^{(1)} + \lambda_i^{\text{del}} \eta_{ij}. \quad (5)$$

If the delegation process is known (i.e., λ_i^{del} and η_{ij} are known), the routing probabilities for System 1 can be solved by rearranging (4) as

$$p_{ij}^{(1)} = p_{ij}/q_i - \eta_{ij}(1 - q_i)/q_i. \quad (6)$$

Note that since the delegation process is controlled by the service operator, the rate and probability distribution of customers delegated (λ_i^{del} and η_{ij}) can be viewed as control inputs, and optimized. In Section III-C we will describe in detail how to solve for λ_i^{del} and η_{ij} . Arrival rates $\lambda_i^{(1)}$, routing probabilities $p_{ij}^{(1)}$, and mean travel times T_{ij} fully describe the System 1 Jackson network.

Now we consider the second Jackson network, System 2, which models the m_d vehicles operating as a taxi service. This network must not only provide service to customers but also rebalance the MoD system to ensure quality of service. To incorporate the notion of vehicle rebalancing, we use the concept of ‘‘virtual’’ customers as in [9]. Virtual customers are generated at station i according to a Poisson process with parameter ψ_i and routing probabilities ξ_{ij} , independent from the real customer arrival process. Virtual customers are lost upon arrival if a taxi is not immediately available, just like real customers. In this way, virtual customers promote rebalancing while not enforcing a strict rebalancing rate, which is key to retaining tractability in the model. The overall customer arrival rate (real and virtual) at station i for System 2 is

$$\lambda_i^{(2)} = \lambda_i^{\text{del}} + \psi_i.$$

With respect to the vehicles, $\lambda_i^{(2)}$ is the exponentially distributed service rate at SS node $i \in S^{(2)}$. The routing probabilities for this network can be defined as

$$\begin{aligned} p_{ij}^{(2)} &= \mathbb{P}(i \rightarrow j \mid \text{virtual}) \frac{\psi_i}{\lambda_i^{(2)}} + \mathbb{P}(i \rightarrow j \mid \text{real}) \frac{\lambda_i^{\text{del}}}{\lambda_i^{(2)}} \\ &= \xi_{ij} \frac{\psi_i}{\lambda_i^{(2)}} + \eta_{ij} \frac{\lambda_i^{\text{del}}}{\lambda_i^{(2)}} \\ &= \xi_{ij} p_i + \eta_{ij} (1 - p_i), \end{aligned} \quad (7)$$

where $p_i := \psi_i/\lambda_i^{(2)}$, similar to the definition in [9].

To summarize our Jackson network model, customers arrive at station i headed for station j according to a Poisson process with rate $\lambda_i p_{ij}$. Upon arrival, each customer is delegated to one of two systems, the customer-driven system (System 1) or the taxi system (System 2). The probability of the customer (going from station i to j) being delegated to System 1 is $\frac{\lambda_i^{(1)} p_{ij}^{(1)}}{\lambda_i p_{ij}}$ and the probability of the customer being delegated to System 2 is $\frac{\lambda_i^{\text{del}} \eta_{ij}}{\lambda_i p_{ij}}$ (from (6)). Once the customer has been delegated, if he/she finds the station empty of vehicles, he/she immediately leaves the system. Once delegated, a customer cannot switch from System 1 to System 2 or vice versa. We note that in the same way that ψ_i represents the rebalancing-promoting rate of vehicles in the MoD system, λ_i^{del} represents the rebalancing-promoting rate of the drivers. Together, the parameters ψ_i , ξ_{ij} , λ_i^{del} , and η_{ij} constitute the open-loop controls for our model of a MoD system. The open-loop control problem is formalized and solved in Section IV.

C. Performance criteria

The task to control the MoD system involves optimizing the parameters λ_i^{del} (rebalancing the drivers) and ψ_i (rebalancing the vehicles) as well as the routing probabilities η_{ij} and ξ_{ij} . The key performance metric is the availability of vehicles (the probability that a customer will find an available vehicle), given by (3). In [14] it was shown that for a closed Jackson network of the form described in the previous section, the availability satisfies $\lim_{m \rightarrow \infty} A_i(m) = \gamma_i/\gamma_S^{\text{max}}$, for all $i \in S$, where γ_i is the relative utilization at node $i \in S$, S is the set of station nodes, and $\gamma_S^{\text{max}} := \max_{i \in S} \gamma_i$. As the number of vehicles increases, the set of stations $B := \{i \in S : \gamma_i = \gamma_S^{\text{max}}\}$ will have availability approaching one while all other stations will have availability strictly less than one. Thus, a natural notion of rebalancing, introduced in [9] for autonomous MoD systems, is to ensure that $A_i(m) = A_j(m)$ for all $i, j \in S$ (or equivalently $\gamma_i = \gamma_j$ for all $i, j \in S$, as implied by (3)). The relative utilizations for each Jackson network are defined as follows

$$\begin{aligned} \gamma_i^{(1)} &= \frac{\pi_i^{(1)}}{\mu_i^{(1)}} = \frac{\pi_i^{(1)}}{\lambda_i - \lambda_i^{\text{del}}} \quad \forall i \in S^{(1)}, \\ \gamma_i^{(2)} &= \frac{\pi_i^{(2)}}{\mu_i^{(2)}} = \frac{\pi_i^{(2)}}{\lambda_i^{\text{del}} + \psi_i} \quad \forall i \in S^{(2)}, \end{aligned}$$

where $\pi_i^{(k)}$, $i \in S^{(k)}$, $k = \{1, 2\}$ satisfies (1). For autonomous MoD systems, the constraint $\gamma_i = \gamma_j$ embodies two features: 1) fairness, as characterized by equal availability across all stations, and 2) performance, since the availability at each station approaches 100% as the number of vehicles increases. We will apply this constraint to both Jackson networks in our MoD system as firstly it is a direct generalization of the approach used for autonomous MoD systems and secondly it yields a linear optimization problem (Section IV-A) that is easy to compute and scale to large systems. However, as discussed in Section IV-B, such approach only approximately balances the system (even though the approximation is often remarkably good). We then introduce a modified approach in Section IV-C that relies on nonlinear optimization, which does ensure fairness while maintaining system performance,

but incurs a higher computational cost. Collectively, the open-loop control approaches of Section IV are useful for analysis and design tasks such as system sizing (Section IV-D) and drive the development of closed-loop policies (Section V).

IV. ANALYSIS AND DESIGN OF MoD SYSTEMS

A. Approximate MoD rebalancing

In this section we formulate a linear optimization approach to (approximately) rebalance a MoD system. Specifically, we would like to manipulate the control variables λ_i^{del} , ψ_i , η_{ij} , and ξ_{ij} such that $\gamma_i^{(1)} = \gamma_j^{(1)}$ for all $i, j \in S^{(1)}$ and $\gamma_i^{(2)} = \gamma_j^{(2)}$ for all $i, j \in S^{(2)}$. To minimize the cost of MoD service, we would like to simultaneously minimize the mean number of rebalancing vehicles on the road (minimize energy use and possibly congestion), given by $\sum_{i,j} T_{ij} \xi_{ij} \psi_i$, as well as the number of rebalancing drivers needed, given by $\sum_{i,j} T_{ij} (\xi_{ij} \psi_i + \eta_{ij} \lambda_i^{\text{del}})$. We can state this multi-objective problem as follows:

MoD Rebalancing Problem (MRP): Given a MoD system modeled as two closed Jackson networks, solve

$$\begin{aligned} & \underset{\lambda_i^{\text{del}}, \psi_i, \eta_{ij}, \xi_{ij}}{\text{minimize}} && \sum_{i,j} T_{ij} \xi_{ij} \psi_i \text{ and } \sum_{i,j} T_{ij} (\xi_{ij} \psi_i + \eta_{ij} \lambda_i^{\text{del}}) \\ & \text{subject to} && \gamma_i^{(k)} = \gamma_j^{(k)} \quad i, j \in S^{(k)}, k = 1, 2 \quad (8) \\ & && \sum_j \eta_{ij} = 1, \quad \sum_j \xi_{ij} = 1, \\ & && \eta_{ij} \geq 0, \quad \xi_{ij} \geq 0, \quad \lambda_i^{\text{del}} \geq 0, \quad \psi_i \geq 0 \\ & && \lambda_i^{\text{del}} \eta_{ij} \leq \lambda_i p_{ij} \quad i, j \in \{1, \dots, N\} \end{aligned}$$

Note that the last constraint in the MRP ensures that the number of customers delegated to the taxi system does not exceed the total number of customers.

Remarkably, the MRP can be solved as two decoupled linear optimization problems with the same form as in [8] (which uses a deterministic, fluidic model). This result, stated in Theorem IV.5, constitutes the main contribution of this section. By decoupling the constraints, we can show that the two objectives are indeed aligned, i.e., minimizing the second objective will minimize the first as well. We begin by presenting supporting lemmas that are used in the proof of Theorem IV.5. The first two lemmas establish some structural properties of the model and were introduced in [11]. They are restated here for completeness; their proofs are virtually identical to the proofs in [11] and are omitted. The first lemma allows the balance equations of the Jackson network to be solved by considering only the SS nodes.

Lemma IV.1 (Folding of balance equations). *Consider either System 1 or System 2 from Section III-B. The relative throughputs π 's for the SS nodes can be found by solving the reduced balance equations*

$$\pi_i^{(k)} = \sum_{j \in S^{(k)}} \pi_j^{(k)} p_{ji}^{(k)} \quad \forall i \in S^{(k)}, k = \{1, 2\}, \quad (9)$$

where SS nodes are considered in isolation. The π 's for the IS nodes are then given by

$$\pi_i^{(k)} = \pi_{\text{Parent}(i)}^{(k)} p_{\text{Parent}(i)\text{Child}(i)}^{(k)} \quad \forall i \in I^{(k)}, k = \{1, 2\}. \quad (10)$$

Lemma IV.2. *For any rebalancing policy $\{\psi_i\}_i$ and $\{\xi_{ij}\}_{ij}$, it holds for all $i \in S^{(2)}$*

$$\begin{aligned} 1) & \gamma_i^{(2)} > 0, \\ 2) & (\lambda_i^{\text{del}} + \psi_i) \gamma_i^{(2)} = \sum_{j \in S^{(2)}} \gamma_j^{(2)} (\psi_j \xi_{ji} + \lambda_j^{\text{del}} \eta_{ji}). \end{aligned}$$

Similarly, for System 1,

$$\begin{aligned} 1) & \gamma_i^{(1)} > 0, \\ 2) & (\lambda_i - \lambda_i^{\text{del}}) \gamma_i^{(1)} = \sum_{j \in S^{(1)}} \gamma_j^{(1)} (\lambda_j p_{ji} - \lambda_j^{\text{del}} \eta_{ji}). \end{aligned}$$

In the next two lemmas, we introduce new optimization variables $\{\alpha_{ij}\}_{ij}$ and $\{\beta_{ij}\}_{ij}$ and show that the constraints $\gamma_i = \gamma_j$ in the MRP are equivalent to linear constraints in these new variables. The proofs are similar to the proof of Theorem IV.3 in [9].

Lemma IV.3 (Constraint equivalence for System 1). *Assume that the $\{\beta_{ij}\}$'s are given. Set $\lambda_i^{\text{del}} = \sum_{j \neq i} \beta_{ij}$, $\eta_{ii} = 0$, and for $j \neq i$,*

$$\eta_{ij} = \begin{cases} \beta_{ij} / \lambda_i^{\text{del}} & \text{if } \lambda_i^{\text{del}} > 0, \\ 1 / (N - 1) & \text{otherwise.} \end{cases}$$

With this definition, the constraint

$$\sum_{j \in S^{(1)}, j \neq i} (\beta_{ij} - \beta_{ji}) = \lambda_i - \sum_{j \in S^{(1)}, j \neq i} \lambda_j p_{ji} \quad (11)$$

is equivalent to the constraint

$$\gamma_i^{(1)} = \gamma_j^{(1)}, \quad i, j \in S^{(1)}.$$

Proof. First, rewrite (11) in terms of λ_i^{del} and η_{ij} 's. We then have

$$\lambda_i - \lambda_i^{\text{del}} = \sum_{j \neq i} (\lambda_j p_{ji} - \lambda_j^{\text{del}} \eta_{ji}).$$

Substituting this expression into the last statement of Lemma IV.2, we have

$$\left(\sum_{j \neq i} (\lambda_j p_{ji} - \lambda_j^{\text{del}} \eta_{ji}) \right) \gamma_i^{(1)} = \sum_{j \neq i} \gamma_j^{(1)} (\lambda_j p_{ji} - \lambda_j^{\text{del}} \eta_{ji}). \quad (12)$$

Let $\varphi_{ij} := \lambda_j p_{ji} - \lambda_j^{\text{del}} \eta_{ji}$ and $\zeta_{ij} := \varphi_{ij} / \sum_j \varphi_{ij}$. Note that $\sum_j \varphi_{ij} = \lambda_i - \lambda_i^{\text{del}} = \lambda_i^{(1)} > 0$ by assumption. The variables ζ_{ij} can be considered transition probabilities of an irreducible Markov chain, and (12) can be rewritten in matrix form as $Z\gamma^{(1)} = \gamma^{(1)}$. Matrix Z is an irreducible, row stochastic matrix, so by the Perron-Frobenius theorem [17], the eigenspace associated with the eigenvalue 1 is one-dimensional. Therefore the unique solution to $Z\gamma^{(1)} = \gamma^{(1)}$ (up to a scaling factor) is the vector $(1, \dots, 1)^T$, so $\gamma_i^{(1)} = \gamma_j^{(1)}$ for all i, j . \square

Lemma IV.4 (Constraint equivalence for System 2). *Assume that the $\{\alpha_{ij}\}$'s are given. Set $\psi_i = \sum_{j \neq i} \alpha_{ij}$, $\xi_{ii} = 0$, and for $j \neq i$,*

$$\xi_{ij} = \begin{cases} \alpha_{ij} / \psi_i & \text{if } \psi_i > 0, \\ 1 / (N - 1) & \text{otherwise.} \end{cases}$$

With this definition, the constraint

$$\sum_{j \neq i} (\alpha_{ij} - \alpha_{ji}) = \sum_{j \neq i} (\beta_{ji} - \beta_{ij}) \quad (13)$$

is equivalent to the constraint

$$\gamma_i^{(2)} = \gamma_j^{(2)}, \quad i, j \in S^{(2)}.$$

The proof is essentially identical to the proof of Lemma IV.3 and is omitted. Furthermore, we can substitute (11) into (13) and rewrite (13) as

$$\sum_{j \neq i} (\alpha_{ij} - \alpha_{ji}) = -\lambda_i + \sum_{j \neq i} \lambda_j p_{ji}. \quad (14)$$

With this substitution, we have decoupled the original MRP constraints into those associated with System 1 (λ_i^{del} and η_{ij}) and those associated with System 2 (ψ_i and ξ_{ij}). Using Lemmas IV.3 and IV.4, one can also show that minimizing the second objective in the MRP also minimizes the first objective. We now state the main result of this section.

Theorem IV.5 (Solution to MRP). *Consider the following two decoupled linear optimization problems*

$$\begin{aligned} & \underset{\beta_{ij}}{\text{minimize}} && \sum_{i,j} T_{ij} \beta_{ij} && (15) \\ & \text{subject to} && \sum_{j \neq i} (\beta_{ij} - \beta_{ji}) = \lambda_i - \sum_{j \neq i} \lambda_j p_{ji} \\ & && 0 \leq \beta_{ij} \leq \lambda_i p_{ij} \end{aligned}$$

$$\begin{aligned} & \underset{\alpha_{ij}}{\text{minimize}} && \sum_{i,j} T_{ij} \alpha_{ij} && (16) \\ & \text{subject to} && \sum_{j \neq i} (\alpha_{ij} - \alpha_{ji}) = -\lambda_i + \sum_{j \neq i} \lambda_j p_{ji} \\ & && 0 \leq \alpha_{ij} \end{aligned}$$

These problems are always feasible. Let β_{ij}^* and α_{ij}^* be optimal solutions to problems (15) and (16) respectively. By making the following substitutions

$$\begin{aligned} \lambda_i^{\text{del}} &= \sum_{j \neq i} \beta_{ij}^*, \\ \psi_i &= \sum_{j \neq i} \alpha_{ij}^*, \\ \eta_{ij} &= \begin{cases} 0 & \text{if } i = j, \\ \beta_{ij}^*/\lambda_i^{\text{del}} & \text{if } \lambda_i^{\text{del}} > 0, i \neq j, \\ 1/(N-1) & \text{otherwise,} \end{cases} \\ \xi_{ij} &= \begin{cases} 0 & \text{if } i = j, \\ \alpha_{ij}^*/\psi_i & \text{if } \psi_i > 0, i \neq j, \\ 1/(N-1) & \text{otherwise,} \end{cases} \end{aligned}$$

one obtains an optimal solution to the MRP.

Proof. Problem (16) is an uncapacitated minimum cost flow problem and is always feasible. Problem (15) is a standard capacitated minimum cost flow problem and its capacity constraints can be shown to always permit the existence of a feasible solution [8], [18, p. 191]. The main task of the proof is showing that the constraints $\gamma_i^{(k)} = \gamma_j^{(k)}$ are equivalent to the constraints in (15) and (16), which is shown in Lemmas IV.3 and IV.4. \square

This result allows us to compute the open-loop control very efficiently and can be applied to very large systems comprising hundreds of stations. We apply this technique in the next section to compute the availability of vehicles at each station and in Section IV-D to the problem of ‘‘sizing’’ a MoD system.

B. Availability of vehicles for real passengers

In general, the availability of vehicles at each station in the customer-driven system is different from the taxi system. The approach in the previous section calculates the availability of the two systems separately, but the availability of vehicles in the taxi system applies not only to real customers, but to virtual customers as well. To calculate the availability for all (real) passengers, we must consider both systems concurrently. First, we note that the total throughput of both real and virtual customers for both networks is given by

$$\Lambda_i^{\text{tot}}(m_v, m_d) = \Lambda_i^{(1)}(m_v - m_d) + \Lambda_i^{(2)}(m_d).$$

The throughput of only real passengers is given by

$$\Lambda_i^{\text{pass}}(m_v, m_d) = \Lambda_i^{(1)}(m_v - m_d) + \frac{\lambda_i^{\text{del}}}{\lambda_i^{\text{del}} + \psi_i} \Lambda_i^{(2)}(m_d),$$

where the second term on the right hand side reflects the fraction of real passengers in the taxi network. Thus, the vehicle availability for real passengers is given by

$$A_i^{\text{pass}}(m_v, m_d) = \Lambda_i^{\text{pass}}(m_v, m_d)/\lambda_i.$$

With some algebraic manipulations, $A_i^{\text{pass}}(m_v, m_d)$ can be rewritten as

$$A_i^{\text{pass}}(m_v, m_d) = A_i^{(1)}(m_v - m_d)q_i + A_i^{(2)}(m_d)(1 - q_i). \quad (17)$$

Since q_i is in general not the same for all i , the availability of vehicles for real passengers will *not* be the same for every station. Fig. 3 shows that the rebalancing technique

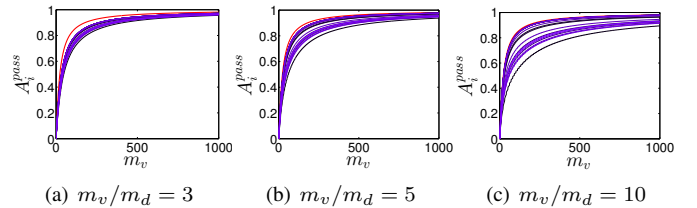


Fig. 3. Overall vehicle availability for passengers for a randomly generated system with 20 stations. Blue lines represent the availability of each station as a function of the number of vehicles. Note that the availability at each station is different, thus the overall system is unbalanced. The red line shows the availability if there were as many drivers as vehicles (as in an autonomous MoD system). 3(a) shows a vehicle-to-driver ratio of 3, 3(b) shows a vehicle-to-driver ratio of 5, and 3(c) shows a vehicle-to-driver ratio of 10.

described in the previous section will produce unbalanced vehicle availabilities for real passengers. Furthermore, the degree of system imbalance grows with the vehicle-to-driver ratio, which intuitively makes sense since there are fewer drivers to rebalance the system when the vehicle-to-driver ratio is high. However, it is important to note that even though the availabilities at each station are not the same, as $m_v \rightarrow \infty$ and $m_d \rightarrow \infty$, the availabilities approach one for *all* stations.

The red line in Fig. 3 shows the availability of the system if there were m_v drivers and m_v vehicles (or equivalently a taxi system or an autonomous MoD system). It is clear that the autonomous MoD system yields better performance both in terms of throughput (high availability) and fairness (same availability at all stations) due to the ability of every vehicle to perform rebalancing trips. This result presents a strong case for the advantages of autonomous MoD systems over current human-driven MoD systems in operation.

C. Exact MoD rebalancing

It is clear that applying the rebalancing constraints separately for the two networks as done in (8) does not yield a balanced system in terms of vehicle availability for all customers. Indeed, the constraints needed to balance availability for the passengers is

$$A_i^{\text{pass}}(m_v, m_d) = A_j^{\text{pass}}(m_v, m_d) \quad \forall i, j \in \{1, \dots, N\}. \quad (18)$$

This set of constraints is dependent on the number of vehicles and the number of drivers in the system, and relative utilizations γ_i can no longer be used to evaluate the constraints in place of real availabilities (3). Thus, constraints (18) cannot be reduced to linear constraints in the optimization variables. Taking into account the modified constraints, we reformulate our problem to the following:

Exact MoD Rebalancing Problem (EMRP): Given a MoD system with N stations, m_v vehicles, and m_d drivers modeled as two closed Jackson networks, solve

$$\begin{aligned} & \underset{\lambda_i^{\text{del}}, \psi_i, \eta_{ij}, \xi_{ij}}{\text{minimize}} && \sum_{i,j} T_{ij} \xi_{ij} \psi_i - c \sum_i A_i^{(2)}(m_v - m_d) && (19) \\ & \text{subject to} && \gamma_i^{(1)} = \gamma_j^{(1)} \\ & && A_i^{\text{pass}}(m_v, m_d) = A_j^{\text{pass}}(m_v, m_d) \\ & && \sum_j \eta_{ij} = 1, \quad \sum_j \xi_{ij} = 1 \\ & && \eta_{ij} \geq 0, \quad \xi_{ij} \geq 0, \quad \lambda_i^{\text{del}} \geq 0, \quad \psi_i \geq 0 \\ & && \lambda_i^{\text{del}} \eta_{ij} \leq \lambda_i p_{ij} \quad i, j \in \{1, \dots, N\}. \end{aligned}$$

The objective function now trades off two objectives that are not always aligned – minimizing the number of rebalancing trips while maximizing the overall availability (note that the first constraint balances and maximizes the availability of the customer-driven system, so to maximize overall availability, we only need to maximize the availabilities in the taxi system). A weighting factor c is used in this trade-off. The constraint $\gamma_i^{(1)} = \gamma_j^{(1)}$ is used in conjunction with (18) to ensure the availability of the customer-driven system remains balanced. The strategy is to use the taxi system to enforce the availability constraint for real customers with the intuition that the system operator has full control over the rebalancing of the taxi system while the rebalancing of the customer-driven system depends on the arrival process of the customers, which is subject to large stochastic fluctuations. If the customer-driven system becomes unbalanced, empty vehicles will accumulate at some stations for extended periods of time, decreasing the effective number of vehicles in the system (see Section V).

The modified availability constraint (18) is nonlinear and involves solving for $A_i^{(2)}$ using MVA at each iteration ($A_i^{(1)}$ is also needed, but only needs to be computed once). For systems of reasonably small size (~ 20 stations and ~ 1000 vehicles), MVA can be carried out quickly (< 1 sec). For larger networks, an approximate MVA technique exists which involves solving a set of nonlinear equations rather than iterating through all values of m [15]. The EMRP can be solved using nonlinear optimization techniques for a given number of vehicles and drivers. We let A^* represent the balanced availability A_i^{pass} obtained by solving the EMRP.

To demonstrate this technique on a realistic system, key system parameters (arrival rates, routing probabilities, and travel times) were extracted from a portion of a data set of New York City taxi trips (courtesy of the New York City Taxi

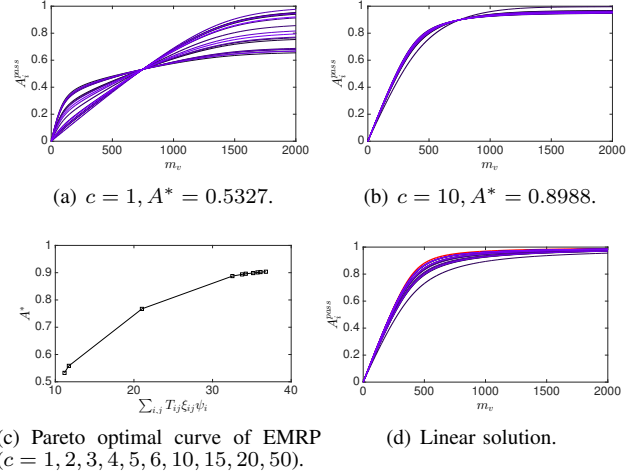


Fig. 4. Nonlinear optimization results for a 20 station system based on Lower Manhattan taxi trip data. The blue lines represent the availability at each station. In 4(a) and 4(b), the availability curves converge to a single value, A^* . This is the point at which the system is balanced. 4(a) shows the optimized availability curves for $c = 1$. 4(b) shows the optimized availability curves for $c = 10$. 4(c) shows the Pareto optimal curve obtained by increasing c from 1 to 50. The x-axis can be interpreted as the average number of rebalancing vehicles on the road. 4(d) shows the linear optimization results for comparison.

and Limousine Commission). Specifically, a 20-station system was created using taxi trips within Lower Manhattan (south of 14th St.) between 10 and 11am on March 1, 2012. The EMRP is solved for this system with 750 vehicles and 150 drivers ($m_v/m_d = 5$). Fig. 4 shows the resulting availability curves and the trade-off between rebalancing rate and system performance.

Fig. 4 shows that as the weighting factor c is increased, vehicle availability increases at the cost of an increased number of rebalancing trips up to a point where it levels off (in this case around 90%). This result compares favorably with the linear solution (4(d)), where at $m_v = 750$, the availabilities range from 0.84 to 0.94. In general, the linear optimization technique appears suitable for computing a first approximation of key design parameters of the system, and the nonlinear technique can be used to further refine the solution. Finally, compared to an autonomous MoD system with the same number of vehicles (red line in Fig. 4(d)), the overall availability is 5% lower (90% vs. 95%). This further shows that autonomous MoD systems would achieve higher levels of performance compared to MoD systems.

D. Application to system sizing

Though the linear programming approach (Section IV-A) does not yield identical availabilities across all stations, it is nonetheless useful for applications such as fleet sizing due to its scalability and efficiency. In this section we provide a simplified example of how to use the MRP approach to gain insight into the optimal vehicle-to-driver ratio (m_v/m_d) of a MoD system. The idea is to find the optimal number of vehicles and drivers that would minimize total cost (or maximize profit) while maintaining an acceptable quality of service. For this simple example, the total cost (normalized by the cost of a vehicle) is

$$C_{\text{total}} = m_v + c_r m_d, \quad (20)$$

where c_r is the cost ratio between a vehicle and a driver. It is reasonable to assume that the cost of a driver is greater than the cost of a vehicle, so $c_r \geq 1$. Three MoD systems are generated using portions of the New York City taxi data: 1) Lower Manhattan (A1), 2) Midtown Manhattan (A2), and 3) Upper Manhattan (A3). Taxi trips within each region are aggregated and clustered into 20 stations, and the system parameters (λ_i , p_{ij} , and T_{ij}) are estimated. Different travel patterns in the three systems allow us to generalize our insights about the optimal m_v/m_d required to minimize cost. For each system with a fixed m_v/m_d , the MRP is solved and the number of vehicles and drivers needed are found such that the lowest availability across all the stations is greater than the availability threshold. Three availability thresholds are investigated (85%, 90%, and 95%). Fig. 5(b) shows the total cost as it varies with the vehicle-to-driver ratio and with c_r for Lower Manhattan with 90% availability threshold. The optimal vehicle-to-driver ratio is the minimum point of each line in 5(b). Fig. 5(c) shows the optimal vehicle-to-driver ratios plotted against the cost ratio c_r for all three Manhattan suburbs and all three availability thresholds.

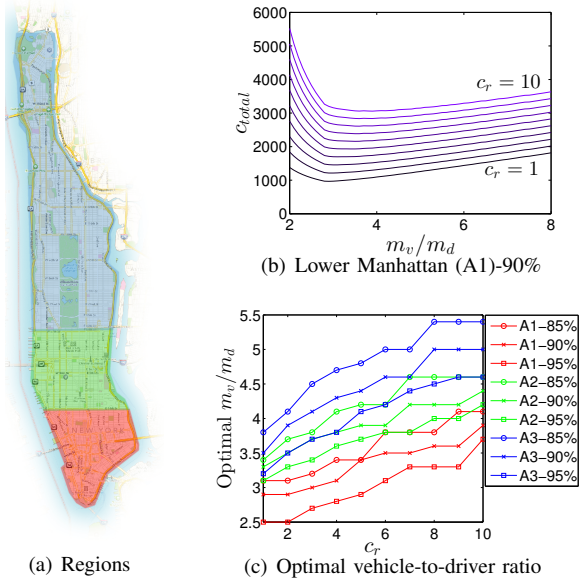


Fig. 5. 5(a) shows the color-coded regions we consider in this case study. 5(b) shows the total cost as a function of the vehicle-to-driver ratio for c_r values ranging from 1 to 10. m_v and m_d values at each point in each curve can be solved using (20). m_v and m_d satisfies the constraint that the availability at each station is greater than the threshold of 90%. 5(c) shows the optimal vehicle-to-driver ratio for the 3 suburbs of Manhattan and 3 availability thresholds (85%, 90%, 95%). For example, the curve A1-90% in 5(c) is constructed by taking the minimum cost of each curve in 5(b).

A few insights can be gained from this example. First, the optimal m_v/m_d ratio does not significantly increase with increasing cost ratio. Second, the optimal m_v/m_d ratio decreases as the availability threshold is raised, consistent with the idea that a high quality of service requires more rebalancing, and thus more drivers. Third, the optimal m_v/m_d ratio is clearly different for each of the Manhattan suburbs (which highlights the important system-dependent nature of this value) but stays between 3 and 5 for a wide range of cost.

V. CLOSED-LOOP CONTROL OF MoD SYSTEMS

In this section we formulate a real-time closed-loop control policy inspired by the open-loop problems presented in Section IV. The closed-loop policy relies on receding-horizon optimization and is targeted towards a practical scenario where customers wait in line for the next available vehicle rather than leave the system. The control policy must perform two tasks: 1) rebalance empty vehicles throughout the network by issuing instructions to drivers, and 2) assign vehicles (with or without driver) to waiting customers at each station. In analogy with the approach in Section IV, we perform these tasks *separately*: that is, we design both a vehicle rebalancing policy and a customer-assignment policy. A vehicle rebalancing policy was introduced in [2] for autonomous MoD (AMoD) systems, which has been shown to be quite effective [9]: hence, we adapt it for our system with little modification. The customer-assignment policy is trickier, and we propose a mixed-integer linear program (MILP) to select the best assignment of customers to customer-driven vehicles or taxis based on the current state of the system. In the proposed policy, customers arriving at each station join a queue of “unassigned” customers. A system-wide optimization problem is solved to try to match as many customers as possible with either an empty vehicle or a taxi while keeping the distribution of the customer-driven vehicles balanced across the system. Once a customer is assigned, he/she moves to the departure queue where he/she will depart from with an empty vehicle or with a taxi. The optimization procedure is performed every time a departure queue is empty and there are unassigned customers. The goal of keeping the distribution of customer-driven vehicles balanced stems from early studies we performed using simple heuristic policies, where we observed customer-driven vehicles aggregating at a small number of stations unused for long periods of time, effectively decreasing the number of vehicles in the system.

Let n_{ij}^v be the number of customers traveling from station i to j to be assigned to drive themselves. Let n_{ij}^d be the number of customers traveling from station i to j to be assigned to a taxi. Denote by v_i^e the number of excess unassigned customer-driven vehicles at station i , v_{ji}^t the number of customer-driven vehicles traveling from station j to i , and v_{ji}^a the number of customer-driven vehicles at station j assigned to travel to station i but that have not yet left the station. Assuming these quantities are known, the number of customer-driven vehicles at a future time step is $v_i^+ = v_i^e + \sum_j (v_{ji}^a + v_{ji}^t + n_{ji}^d - n_{ij}^v)$. We can additionally define a desired vehicle distribution v_i^{des} . For example, an even vehicle distribution would be $v_i^{\text{des}} = (m_v - m_d)/N$. Alternatively, the desired vehicle distribution could be based on demand estimation, for example, $v_i^{\text{des}} = (m_v - m_d)\lambda_i / \sum_j \lambda_j$. The assignment policy is given by solving the following optimization problem:

$$\begin{aligned} & \underset{n_{ij}^d, n_{ij}^v}{\text{minimize}} && \sum_i |v_i^+ - v_i^{\text{des}}| - w \sum_{i,j} (n_{ij}^d + n_{ij}^v) \quad (21) \\ & \text{subject to} && n_{ij}^d + n_{ij}^v \leq c_{ij}^u \\ & && \sum_j n_{ij}^v \leq v_i^e, \quad \sum_j n_{ij}^d \leq d_i^u \\ & && n_{ij}^v \geq 0, \quad n_{ij}^d \geq 0, \quad n_{ij}^v \in \mathbb{Z}, \quad n_{ij}^d \in \mathbb{Z}, \end{aligned}$$

where c_{ij}^u is the number of unassigned customers traveling from i to j , d_i^u is the number of unassigned drivers at station i ,

and w is a weighting factor. The objective function trades off the relative importance of system balance and customer wait times (increasing w would allow the system to assign more customers and reduce wait times). The constraints ensure that the assignment policy is feasible (there are enough vehicles, drivers, and customers). Problem (21) is formulated as a MILP and solved using the IBM CPLEX solver [19].

The overall real-time closed-loop control policy is presented in Algorithm 1. Outstanding customers are assigned to either a customer-driven vehicle or a rebalancer-driven vehicle by solving Problem (21). The AMoD controller in [2] is then used to compute a desired rebalancing policy for empty vehicles. Finally, idle drivers are assigned to drive empty vehicles according to the desired rebalancing policy.

Algorithm 1 Real-time closed-loop control policy

```

procedure REALTIMEREBALANCEANDASSIGN
   $\{n_{ij}^d, n_{ij}^v\} \leftarrow$  solve Problem (21)
  for all  $waitingCustomer(i, j)$  do
    if  $n_{ij}^v > 0$  and vehicle available at  $i$  then
      assign  $waitingCustomer(i, j)$  to empty vehicle
       $n_{ij}^v - = 1$ 
    else if  $n_{ij}^d > 0$  and vehicle and driver avail. then
      assign  $waitingCustomer(i, j)$  to taxi
       $n_{ij}^d - = 1$ 
   $\{RebQueue(s)\} \leftarrow$  AMODREBALANCING( $v_{ij}^e, d_{ij}^u$ )
  for  $s \in Stations$  do
    while  $RebQueue(s) \neq \emptyset$  and driver and vehicle available
    at station  $s$  do
      Send vehicle and driver to a destination in
       $RebQueue(s)$ 
      Remove destination from  $RebQueue(s)$ 
  
```

VI. CASE STUDIES: MOD IN MANHATTAN AND HANGZHOU

We assess the performance and scalability of the real-time closed-loop control policy through case studies, based on real-world transportation data, of two hypothetical MoD systems. The first MoD system operates over all of Manhattan (as opposed to small regions as in Section IV-D) and its customer demand is derived from the dataset used in Section IV. We place 100 stations within Manhattan based on k -means clustering of taxi requests. The resulting station positions are such that customer demands are on average less than 300 meters from the nearest station. The second MoD system operates in Hangzhou, China, and its customer demand is derived from a dataset provided by Didi Chuxing. For this case study customer demand data is only available at an aggregate level, in terms of sixty-six districts—while this dataset has a rather coarse spatial resolution, it is nonetheless interesting as it represents a challenging setting for the operation of a MoD system, as detailed below.

Specifically, the two case studies illustrate the performance of the real-time closed-loop algorithm in markedly different settings: in particular, transportation demand in Hangzhou is significantly more unbalanced than in Manhattan. We measure the level of unbalance of transportation requests with the first Wasserstein distance between the spatial distribution of customer destinations f^d and the spatial distribution of customer origins f^o [20], [21]. The first Wasserstein distance (also known as earth-mover distance) is defined as

$$W(f^d, f^o) := \inf_{\gamma \in \Gamma(f^d, f^o)} \int \mathcal{D}(p^d, p^o) d\gamma(p^d, p^o)$$

where $\Gamma(f^o, f^d)$ is the set of all distributions with marginals f^o and f^d and $\mathcal{D}(x, y)$ is the distance between locations x and y . The average Wasserstein distance between the origin distribution and the destination distribution in Manhattan is 0.509 km; the average Wasserstein distance in Hangzhou is 2.129 km, over four times higher.

For each hour of the day, customer arrival rates λ_i and routing probabilities p_{ij} are estimated by counting the number of trips that originate and end at each station (or district). The travel times T_{ij} are estimated from the average vehicle speed and the Manhattan distance between stations (or districts). Customer arrivals are generated at each station as a Poisson process with mean λ_i and routing probabilities p_{ij} . In contrast to the analysis in the previous section, simulated customers wait at the station until a vehicle has been assigned to them.

Simulations are performed for different vehicle-to-driver ratios, for 24 hours with a time step of 6 seconds. The results of the simulations are shown in Fig. 6.

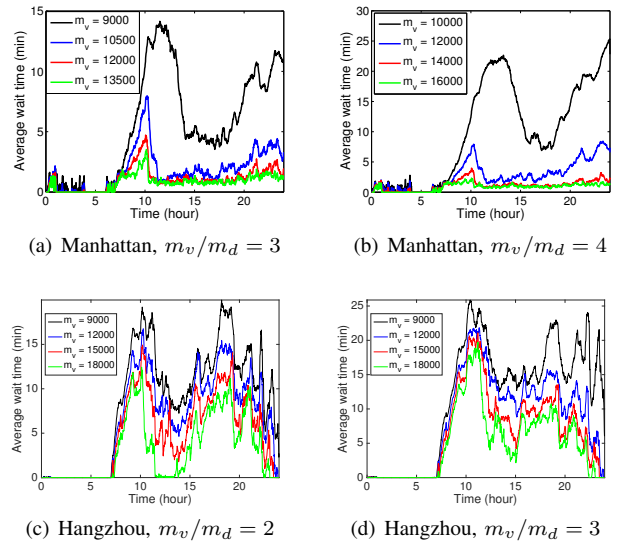


Fig. 6. Average customer wait times throughout the day.

In Manhattan, for a vehicle-to-driver ratio of 3, a system consisting of 12,000 vehicles and 4,000 drivers yielded a maximum average wait time of under 5 minutes, which is indicative of adequate service. For a vehicle-to-driver ratio of 4, satisfactory quality of service is reached between 12,000 and 14,000 vehicles (3,000 to 3,500 drivers). For comparison, New York City has over 13,000 taxis, and 85% of trips are within Manhattan. This suggests that by operating a fraction of the vehicles as a taxi service to maintain system balance, a MoD system can achieve comparable quality of service to taxi systems with only 1/4 to 1/3 the number of drivers.

Instead, in Hangzhou, where transportation demand is significantly more unbalanced, a lower vehicle-to-driver ratio is required to achieve good quality of service. With a vehicle-to-driver ratio of 2, 9,000 cars are sufficient to keep wait times below 20 minutes, and 15,000 cars reduce the peak wait time below 15 minutes; conversely, with a vehicle-to-driver ratio of 3, 18,000 vehicles are required to reduce the peak wait time below 20 minutes and, with a vehicle-to-driver ratio of 4 (not shown), a fleet of 24,000 vehicles is needed to achieve the same peak wait time.

We also compare the simulation results to the queueing network analysis in Section IV. Using the linear approximate rebalancing analysis presented in Section IV-A, we study the availability of vehicles in Manhattan and Hangzhou during the peak period from 9 to 10 am. The availability curves are shown in Fig. 7, and compared to a balanced taxi or autonomous vehicle system (shown in red). In Manhattan, for a MoD system to achieve the same theoretical performance as an autonomous MoD with 8,000 vehicles (which, according to [9], achieves acceptable quality of service, with peak wait times under 5 minutes), 9,789 vehicles are needed for a vehicle-to-driver ratio of 3, and 10,267 vehicles are needed for a vehicle-to-driver ratio of 4. This corroborates well with simulation results in Fig. 6, which suggests that between 11,000 and 12,000 vehicles are needed. In Hangzhou, 7,473 and 7,880 vehicles are required to achieve 95% availability at all stations for vehicle-to-driver ratios of 2 and 3, respectively; such a discrepancy between high availability and long peak customer wait times is most likely due to the spatial coarseness of the data.

The driver assignment optimization problem in the simulations was solved in an average of 0.5 seconds. Since the problem size only scales with the number of stations and the constraints consist mostly of bounding hyperplanes, the feasible set is easy to compute and the problem can be solved in real-time for large scale systems.

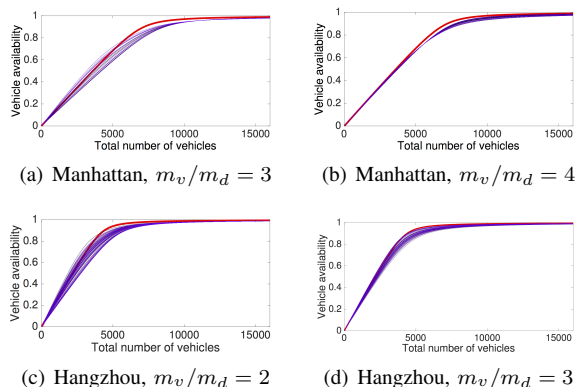


Fig. 7. Vehicle availability during peak demand.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented a queueing network model of a MoD system and developed two open-loop control approaches useful for design tasks such as system sizing. We applied such approaches to a system sizing example for three Manhattan neighborhoods, which showed that the optimal vehicle-to-driver ratio is between 3 and 5. Drawing insights from these techniques, we developed a real-time closed-loop control policy and demonstrated its effectiveness with case studies of two hypothetical MoD systems in Manhattan and Hangzhou.

This work paves the way for several important extensions. First, we plan to include other methods of rebalancing drivers such as allowing them to use public transit or to shuttle multiple other drivers to stations with excess unused vehicles. Second, it is of interest to include congestion effects in our model. A possible strategy is to modify the IS nodes by considering a finite number of servers, representing the capacity of

the road. Third, we plan to test our strategies on microscopic and mesoscopic models of transportation networks. Finally, we would like to incorporate the effects of dynamic pricing incentives for customers on the amount of rebalancing that is required.

REFERENCES

- [1] W. J. Mitchell, C. E. Borroni Bird, and L. D. Burns, *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*. Cambridge, MA: The MIT Press, 2010.
- [2] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *International Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, Jun. 2012.
- [3] M. Barth, M. Todd, and L. Xue, "User-based vehicle relocation techniques for multiple-station shared-used vehicle systems," in *Transportation Research Board 80th Annual Meeting*, Washington, DC, 2004.
- [4] S. Herrmann, F. Schulte, and S. Voß, "Increasing acceptance of free-floating car sharing systems using smart relocation strategies: a survey based study of car2go Hamburg," in *Computational Logistics*. Springer, 2014, pp. 151–162.
- [5] D. Jorge, G. H. A. Correia, and C. Barnhart, "Comparing Optimal Relocation Operations With Simulated Relocation Policies in One-Way Carsharing Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1667–1675, 2014.
- [6] R. Nair and E. Miller Hooks, "Fleet management for vehicle sharing operations," *Transportation Science*, vol. 45, no. 4, pp. 524–540, 2011.
- [7] Clemens Grupp, "The Relocation Planning Problem in Free-Floating Car Sharing: Models and Heuristics." Master's thesis, Technische Universität München, 2015.
- [8] S. L. Smith, M. Pavone, M. Schwager, E. Frazzoli, and D. Rus, "Rebalancing the Rebalancers: Optimally Routing Vehicles and Drivers in Mobility-On-Demand Systems," in *American Control Conference*, Washington, DC, Jun. 2013, pp. 2362–2367.
- [9] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 186–203, Jan. 2016.
- [10] J. Chen, K. H. Low, and C. K. Y. Tan, "Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system," in *Robotics: Science and Systems*, 2013.
- [11] R. Zhang and M. Pavone, "A Queueing Network Approach to the Analysis and Control of Mobility-On-Demand Systems," in *American Control Conference*, Chicago, IL, Jul. 2015, pp. 4702–4709.
- [12] R. Serfozo, *Introduction to stochastic networks*. Springer, 1999, vol. 44.
- [13] R. Zhang and M. Pavone, "Control of Robotic Mobility-On-Demand Systems: a Queueing-Theoretical Perspective," in *Robotics: Science and Systems*, Berkeley, CA, Jul. 2014, pp. 1–9. [Online]. Available: <http://www.roboticsproceedings.org/rss10/p26>
- [14] D. K. George and C. H. Xia, "Fleet-sizing and service availability for a vehicle rental system via closed queueing networks," *European Journal of Operational Research*, vol. 211, no. 1, pp. 198–207, 2011.
- [15] M. Reiser and S. S. Lavenberg, "Mean-value analysis of closed multichain queueing networks," *Journal of the Association for Computing Machinery*, vol. 27, no. 2, pp. 313–322, 1980.
- [16] S. S. Lavenberg, *Computer performance modeling handbook*. Elsevier, 1983, vol. 4.
- [17] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.
- [18] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 4th ed., ser. Algorithmics and Combinatorics. Springer, 2007, vol. 21.
- [19] *ILOG CPLEX User's guide*, ILOG, Mountain View, CA, 1999.
- [20] K. Treleaven, M. Pavone, and E. Frazzoli, "Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2261–2276, 2013.
- [21] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of Autonomous Mobility-on-Demand systems: A case study in Singapore," in *Road Vehicle Automation*. Springer, 2014.