Control of Robotic Mobility-On-Demand Systems: a Queueing-Theoretical Perspective

Rick Zhang
Department of Aeronautics and Astronautics, Stanford University, USA

Marco Pavone*
Department of Aeronautics and Astronautics, Stanford University, USA

Abstract

In this paper we present queueing-theoretical methods for the modeling, analysis, and control of autonomous mobility-on-demand (MOD) systems wherein robotic, self-driving vehicles transport customers within an urban environment and rebalance themselves to ensure acceptable quality of service throughout the network. We first cast an autonomous MOD system within a closed Jackson network model with passenger loss. It is shown that an optimal rebalancing algorithm minimizing the number of (autonomously) rebalancing vehicles while keeping vehicle availabilities balanced throughout the network can be found by solving a linear program. The theoretical insights are used to design a robust, real-time rebalancing algorithm, which is applied to a case study of New York City and implemented on a 8-vehicle mobile robot testbed. The case study of New York shows that the current taxi demand in Manhattan can be met with about 8,000 robotic vehicles (roughly 70% of the size of the current taxi fleet operating in Manhattan). Finally, we extend our queueing-theoretical setup to include congestion effects, and study the impact of autonomously rebalancing vehicles on overall congestion. Using a simple heuristic algorithm, we show that additional congestion due to autonomous rebalancing can be effectively avoided on a road network. Collectively, this paper provides a rigorous approach to the problem of system-wide coordination of autonomously driving vehicles, and provides one of the first characterizations of the sustainability benefits of robotic transportation networks.

Keywords

Self-driving cars, intelligent transportation systems, vehicle routing, queueing networks, autonomous systems.

1. Introduction

According to United Nations estimates, urban population will double in the next 30 years (UN, 2011). Given the limited availability for additional roads and parking spaces in current (mega)-cities, private automobiles appear as an *unsustainable*

* Corresponding author; e-mail: pavone@stanford.edu
A preliminary version of this paper has appeared in the Proceedings of the 2014 Robotics: Science and Systems Conference (Zhang & Pavone, 2014).

solution for the future of *personal* urban mobility (Mitchell et al., 2010). Arguably, one of the most promising approaches to cope with this problem is one-way vehicle sharing with electric cars (referred to as Mobility-On-Demand, or MOD), which directly targets the problems of parking spaces, pollution, and low vehicle utilization rates (Mitchell et al., 2010). Limited-size MOD systems with human-driven vehicles have recently been deployed in several European and American cities (CAR2GO, 2011). However, such systems lead to vehicle *imbalances*, that is some stations become rapidly depleted of vehicles while others have too many, due to some stations being more popular than others. Somewhat surprisingly, even if the transportation network is symmetric (that is, the underlying network topology is a regular grid and arrival rates and routing choices of customers at all nodes are uniform), the *stochastic* nature of customer arrivals to the stations will quickly drive the system out of balance and hence to instability, since the customer queue will grow without bound at some stations (Fricker & Gast, 2012).

The related problem of rebalancing in the increasingly popular bike-sharing systems is solved using trucks which can carry many bikes at the same time, and algorithms have very recently been developed to optimize the truck routes (Di Gaspero et al., 2013; Dell'Amico et al., 2014). Since this approach is not feasible with cars, the work in Smith et al. (2013) considers the possibility of hiring a team of rebalancing drivers whose job is to rebalance the vehicles throughout the transportation network. However, with this approach the rebalancing drivers themselves become unbalanced, and one needs to "rebalance the rebalancers," which significantly increases congestion and costs (Smith et al., 2013). Another option would be to incentivize ride sharing (Barth et al., 2001), which, unfortunately, defeats the purpose of a MOD system to ensure *personal* mobility.

Recently, a transformational technology has been proposed in Pavone et al. (2012); Burns et al. (2013), whereby *driverless* electric cars shared by the customers provide on-demand mobility. Autonomous driving holds great promise for MOD systems because robotic vehicles can rebalance themselves (thus eliminating the rebalancing problem at its core), enable system-wide coordination, free passengers from the task of driving, and potentially increase safety. Indeed, robotic vehicles specifically designed for personal urban mobility are already being tested, e.g., the Induct Navia vehicle (Induct, 2013), the General Motors' EN-V vehicle (GM, 2011), and the Google car (Fisher, 2013). Yet, little is known about how to design and operate *robotic* transportation networks (Pavone et al., 2012).

Statement of contributions: The objective of this paper is to develop a model of, study rebalancing algorithms for, and evaluate the potential benefits of a MOD system where mobility is provided by driverless cars (henceforth referred to as autonomous MOD system). Rebalancing algorithms for autonomous MOD systems have been investigated in Pavone et al. (2012) under a fluidic approximation (i.e., customers and vehicles are modeled as a continuum). While this approach provides valuable insights for the operation of an autonomous MOD system, by its very nature, it does not provide information about the effect of stochastic fluctuations in the system (e.g., due to the customers' arrival process) and, most importantly, it does not allow the computation of key performance metrics such as availability of vehicles at stations and customer waiting times. This motivates the queueing-theoretical approach considered in this paper. In this respect, our work is related to George & Xia (2011); Waserhole & Jost (2013), where a transportation network comprising traditional (i.e., human-driven) shared vehicles is modeled within the framework of Jackson networks (Serfozo, 1999). The key technical difference is that in this paper we address the problem of *synthesizing* a rebalancing policy, rather than *analyzing* the evolution of the vehicle distribution under the customers' routing choices. Our work is also related to the Dynamic Vehicle Routing problem (Bullo et al., 2011; Pavone et al., 2011) and, more specifically, to the Dynamic Pickup and Delivery Problem (DPDP) (Berbeglia et al., 2010; Treleaven et al., 2013). However, current DPDP works do not explicitly consider rebalancing, and system performance results, whenever available, are usually limited to asymptotic cases where the load on the system approaches zero or approaches the capacity of the system (Treleaven et al., 2013).

Specifically, the contribution of this paper is fourfold. First, we propose a queueing-theoretical model of an autonomous MOD system cast within a Jackson network model (valid under any load condition). Second, we study the problem of synthesizing rebalancing algorithms, where the control objective is to minimize the number of (autonomously) rebalancing

vehicles on the roads while keeping vehicle availabilities balanced throughout the network. Remarkably, we show that under certain assumptions an optimal policy can be solved as a linear program. Third, we apply our theoretical results to a case study of New York City and an experimental testbed with 8 mobile robots. The case study of New York City shows that the current taxi demand in Manhattan can be met with about 8,000 robotic vehicles (roughly 70% of the size of the current taxi fleet operating in Manhattan). This shows the potential of autonomous MOD systems. Finally, by leveraging our queueing-theoretical setup, we study the potential detrimental effect of rebalancing on traffic congestion (rebalancing vehicles, in fact, *increase* the number of vehicles on the roads). Our study suggests that while autonomously rebalancing vehicles can have a detrimental impact on traffic congestion in already-congested systems, in most cases this is not generally a concern as rebalancing vehicles can be routed to travel along less congested roads.

A preliminary version of this paper appeared as Zhang & Pavone (2014). This extended and revised version contains as additional contributions: (1) proofs of all results, (2) additional numerical results, discussion, and simulation video (see Extension 1) for the New York case study with an emphasis on waiting time distributions and dependency on the locations and number of stations, (3) hardware experimental results and video (see Extension 2) using a mobile robot testbed, and (4) an extended discussion about congestion effects, a heuristic algorithm to mitigate maximum road utilization due to autonomous rebalancing, and corresponding simulation results. To the best of our knowledge, this is the first paper (together with its conference version (Zhang & Pavone, 2014)) to provide a rigorous, stochastic approach to the problem of system-wide coordination of autonomously driving vehicles.

Organization: The remainder of the paper is structured as follows: In Section 2 we briefly review some well-known results for queueing networks, specifically Jackson networks. In Section 3 we show how to model an autonomous MOD system with rebalancing within a Jackson network model. In Section 4 we formulate the optimal rebalancing problem, we show that it can be solved via a linear program, we provide an iterative algorithm to compute relevant performance metrics (chiefly, vehicle availability at stations), and we use the theoretical insights to design a robust, real-time rebalancing policy. In Section 5 we apply our model and algorithms to a case study of New York City, while in Section 6 we apply the same algorithms to a 8-vehicle hardware testbed. In Section 7 we extend our queueing-theoretical setup to include congestion effects and numerically analyze the impact of rebalancing on congestion through a heuristic scheme. Finally, in Section 8 we draw our conclusions and present directions for future research.

2. Background Material

In this section we review some key results from the theory of Jackson networks, on which we will rely extensively later in the paper. Consider a network consisting of $|\mathcal{N}|$ first-come first-serve nodes, or queues, where $\mathcal{N}$ represents the set of nodes in the network. Discrete agents[1] arrive from outside the network according to a stochastic process or move among the nodes. Agents that arrive at each node are serviced by the node, and proceed to another node or leave the system. A network is called *closed* if the number of agents in the system remains constant and no agents enter or leave the network. A Jackson network is a Markov process where agents move from node to node according to a stationary routing distribution r_{ij} and the service rate $\mu_i(n)$ at each node i depends only on the number of agents at that node, n (Serfozo, 1999, p. 9). For the remainder of this paper, we consider only closed networks. The state space of a closed Jackson network with m agents is given by

$$\Omega_m := \Big\{(x_1, x_2, \ldots, x_{|\mathcal{N}|}) : x_i \in \mathbb{N} \text{ for all } i \in \mathcal{N}, \sum_{i=1}^{|\mathcal{N}|} x_i = m\Big\},$$

[1] Entities moving through the queues are more commonly referred to as "customers" rather than "agents" in the queueing theory literature. We use the term "agents" instead of "customers" to avoid confusion later on, since these entities in our abstract model of an autonomous MOD system will represent self-driving vehicles. In our model, "customers" will represent people requiring transportation within the network.

where x_i is the number of agents at node i. Jackson networks are known to admit a product-form stationary distribution, where the stationary distribution of the network is given by a product of the distribution of each node. In equilibrium, the throughput at each node (average number of agents moving through a node per unit time), denoted by π_i for $i \in \mathcal{N}$, satisfies the traffic equations

$$\pi_i = \sum_{j \in \mathcal{N}} \pi_j r_{ji} \quad \text{for all } i \in \mathcal{N}. \tag{1}$$

For a closed network, equation (1) does not have a unique solution, and $\pi := (\pi_1, \pi_2, \ldots, \pi_{|\mathcal{N}|})^T$ only determines the throughput up to a constant factor, and is therefore called the relative throughput. The stationary distribution of the network is given by

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

The quantity $G(m)$ is the normalization constant needed to make $\mathbb{P}(x_1, x_2, ..., x_{|\mathcal{N}|})$ a probability measure, and is given by

$$G(m) = \sum_{x \in \Omega_m} \prod_{j=1}^{|\mathcal{N}|} \pi_j^{x_j} \prod_{n=1}^{x_j} \mu_j(n)^{-1},$$

where $x := (x_1, x_2, \ldots, x_{|\mathcal{N}|})$. Many performance measures of closed Jackson networks can be expressed in terms of the normalization factor $G(m)$. In Serfozo (1999, p. 27), it is shown that the actual throughput of each node is given by

$$\Lambda_i(m) = \pi_i \, G(m-1)/G(m). \tag{2}$$

One can further define the quantity

$$\gamma_i = \pi_i / \mu_i(1) \quad \text{for all } i \in \mathcal{N}, \tag{3}$$

where γ_i is referred to as the relative utilization of node i. Lavenberg (1983, p. 128) showed that the marginal distribution of the queue length variable X_i at node $i \in \mathcal{N}$ is given by

$$\mathbb{P}(X_i = x_i) = \gamma_i^{x_i} \left[G(m - x_i) - \gamma_i G(m - x_i - 1) \right] / G(m).$$

A quantity of interest is the probability that a node has at least 1 agent, which we refer to as the *availability* of node i, $A_i(m)$. This is given by

$$A_i(m) = 1 - P(X_i = 0) = 1 - \frac{G(m) - \gamma_i G(m-1)}{G(m)} = \frac{\gamma_i G(m-1)}{G(m)}. \tag{4}$$

3. Model Description and Problem Formulation

3.1. Model of autonomous MOD system

In this paper, we model an autonomous MOD system within a queueing theoretical framework. Consider N stations placed within a given geographical area and m (autonomous) vehicles that provide service to passengers. Customers arrive at each station i according to a time-invariant Poisson process with rate $\lambda_i \in \mathbb{R}_{>0}$. Upon arrival, a customer at station i selects a destination j with probability p_{ij}, where $p_{ij} \in \mathbb{R}_{\geq 0}$, $p_{ii} = 0$, and $\sum_j p_{ij} = 1$. Furthermore, we assume that the probabilities $\{p_{ij}\}_{ij}$ constitute an irreducible Markov chain. If there are vehicles parked at station i, the customer takes the vehicle and travels to her/his selected destination. Instead, if the station is empty of vehicles, the customer immediately leaves the system. This type of customer model will be referred to as a "passenger loss" model (as opposed to a model where passengers form a queue at each station). A consequence of the passenger loss model is that the number of customers

at each station at a fixed instant in time is 0 (since customers either depart immediately with a vehicle or leave the system). We assume that each station has sufficiently many parking spaces so that vehicles can always immediately park upon arrival at a station. The travel time from station i to station j is an exponentially distributed random variable with mean equal to $T_{ij} \in \mathbb{R}_{>0}$. The travel times for the different passengers are assumed to constitute an independently and identically distributed sequence (i.i.d.). The vehicles can *autonomously* travel throughout the network in order to rebalance themselves and best anticipate future demand. The performance criterion that we are interested in is the availability of vehicles at each station (the probability that at least one vehicle is at the station or conversely the probability that a customer will be lost).

A few comments are in order. First, our model captures well the setup with impatient customers, not willing to make use of a MOD system if waiting is required. In this respect, our model appears to be suitable for studying the benefits of autonomous MOD systems whenever high quality of service (as measured in terms of average waiting times for available vehicles) is required. From a practical standpoint, the loss model assumption significantly simplifies the problem, as it essentially allows us to decouple the "vehicle process" and the "customer process" (see Section 3.2). Second, travel times, in practice, do not follow an exponential distribution. However we make this assumption as (i) it simplifies the problem considerably, and (ii) reasonable deviations from this assumption have been found not to alter in any practical way the predictive accuracy of similar spatial queueing models used for vehicle routing (Larson & Odoni, 1981). Third, the assumption that the probabilities $\{p_{ij}\}_{ij}$ constitute an irreducible Markov chain appear appropriate for dense urban environments. Finally, our model does not consider congestion, which is clearly a critical aspect for the efficient system-wide coordination of autonomous vehicles in a MOD system. The inclusion of congestion effects will be discussed in Section 7.

3.2. Casting an autonomous MOD system into a Jackson model

The key idea to cast an autonomous MOD system into a Jackson model is to consider an *abstract queueing network* where we identify the stations with single-server (SS) nodes (also referred to as "station" nodes) and the roads with infinite-server (IS) nodes (also referred to as "road" nodes). Assume, first, the simplified scenario where vehicles do not perform rebalancing trips (in which case, the model is essentially identical to the one in George & Xia (2011)). In this case, at each station node, vehicles form a queue while waiting for customers and are "serviced" when a customer arrives. A vehicle departing from a SS node moves to the IS node that connects the origin to the destination selected by the customer. After spending an exponentially distributed amount of time in the IS node (i.e., the "travel time"), the vehicle moves to the destination SS node. According to our model, once a vehicle leaves a SS (station) node i, the probability that it moves to the IS (road) node ij is p_{ij}. The vehicle then moves to SS (station) node j with probability 1 (see Figure 1). Note that with this identification we have modeled a MOD system (at least in the case without rebalancing) as a *closed queueing network with respect to the vehicles*. Note that the road queues are modeled as infinite-server queues as the model does not consider congestion effects (in Section 7 we will see that if congestion is taken into account the road queues become *finite*-server queues).

More formally, denote by S the set of single-server nodes and I the set of infinite-server nodes. Each station is mapped into a SS node, while each road is mapped into an IS node. The set of all nodes in the abstract queueing network is then given by $\mathcal{N} = S \cup I$. Since each SS node is connected to every other SS node, and since $p_{ii} = 0$ (hence the road node ii does not need to be represented), the number of nodes in the network is given by $N(N-1) + N = N^2$, in other words, $|\mathcal{N}| = N^2$. For each IS node $i \in I$, let Parent(i) and Child(i) be the origin and destination nodes of i, respectively. As explained before, vehicles in the abstract queueing network move between SS nodes and IS nodes according to the routing

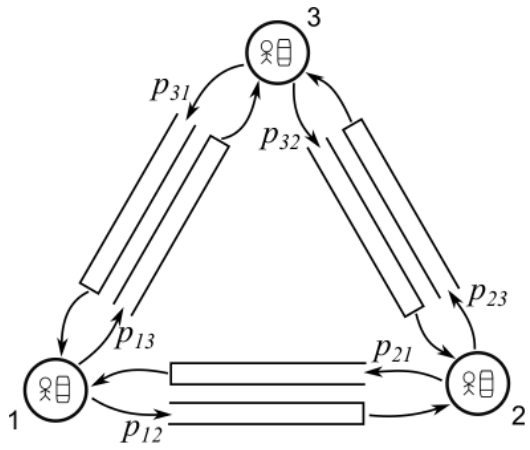


Fig. 1. A 3-station network. Circles represent the SS nodes (stations) and rectangles represent the IS nodes (roads).

matrix $\{r_{ij}\}_{ij}$:

$$r_{ij} = \begin{cases} p_{il} & i \in S, j \in I \quad \text{where } i = \text{Parent}(j), l = \text{Child}(j), \\ 1 & i \in I, j \in S \quad \text{where } j = \text{Child}(i), \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where the first case corresponds to a move from a SS node to an IS node (according to the destination selected by a customer), and the second case to a move from an IS node to the unique SS node corresponding to its destination. Furthermore, the service times at each node $i \in \mathcal{N}$ are exponentially distributed with service rates given by

$$\mu_i(n) = \begin{cases} \lambda_i & \text{if } i \in S, \\ n \cdot \mu_{jk} & \text{if } i \in I, j = \text{Parent}(i), k = \text{Child}(i), \end{cases} \tag{6}$$

where $n \in \{0, 1, \dots\ m\}$ is the number of vehicles at node i, and $\mu_{jk} = 1/T_{jk}$. The first case is the case where vehicles wait for customers at stations, while the second case is the case where vehicles spend an exponentially distributed travel time to move between stations (note that the IS nodes correspond to infinite-server queues, hence the service rate is proportional to the number of vehicles in the queue). As defined, the abstract queueing network is a closed Jackson network, and hence can be analyzed with the tools discussed in Section 2.

Assume, now, that we allow the vehicles to autonomously rebalance throughout the network. To include rebalancing while staying within the Jackson network framework, we focus on a particular class of stochastic rebalancing policies described as follows. Each station i generates "virtual passengers" according to a Poisson process with rate ψ_i, *independent* of the real passenger arrival process, and routes these virtual passengers to station j with probability α_{ij} (with $\sum_j \alpha_{ij} = 1$ and $\alpha_{ii} = 0$). As with real passengers, the virtual passengers are lost if the station is empty upon generation. Such class of rebalancing policies *encourages* rebalancing but does *not* enforce a rebalancing rate, which allows us to maintain tractability in the model.

One can then combine the real passenger arrival process with the virtual passenger process (assumed independent) using the independence assumption to form a model of the same form as the one described earlier in this section while taking into account vehicle rebalancing. Specifically, we consider the same set of SS nodes and IS nodes (since the transportation network is still the same). Let $\{A_t^{(i)}, t \geq 0\}$ be the total arrival process of real *and* virtual passengers at station $i \in S$, and denote its rate with $\tilde{\lambda}_i$. The process $A_t^{(i)}$ is Poisson since it is the superposition of two independent Poisson processes. Hence, the rate $\tilde{\lambda}_i$ is given by

$$\tilde{\lambda}_i = \lambda_i + \psi_i. \tag{7}$$

Equivalently, one can view the passenger arrival process and the rebalancing process as the result of Bernoulli splitting on $A_t^{(i)}$ with a probability p_i satisfying

$$\psi_i = p_i\tilde{\lambda}_i, \quad \lambda_i = (1-p_i)\tilde{\lambda}_i. \tag{8}$$

Let us refer to passengers arriving according the the processes $\{A_t^{(i)}, t \geq 0\}$ as generalized passengers. The probability $\tilde{p}_{ij}$ that a generalized passenger arriving at station i selects a destination j is given by

$$\begin{aligned}\tilde{p}_{ij} &= \mathbb{P}(i \to j \mid \text{virtual})\, p_i + \mathbb{P}(i \to j \mid \neg\text{virtual})\,(1-p_i) \\ &= \alpha_{ij}p_i + p_{ij}(1-p_i),\end{aligned} \tag{9}$$

where $\mathbb{P}(i \to j \mid \text{virtual})$ is the probability that a virtual passenger selects station j as its destination, and $\mathbb{P}(i \to j \mid \neg\text{virtual})$ is the probability that a real passenger selects station j as its destination. One can then identify an autonomous MOD system with rebalancing (for the specific class of rebalancing policies discussed above) with an abstract queueing network with routing matrix and service rates given, respectively, by equations (5) and (6), where p_{il} is replaced by $\tilde{p}_{il}$, λ_i is replaced by $\tilde{\lambda}_i$, and r_{ij} is replaced by $\tilde{r}_{ij}$. In this way, the model is still a closed Jackson network model. For notational convenience, we order γ_i and π_i (as defined in Section 2) in such a way that the first N components correspond to the N stations (for example, γ_i corresponds to station i, or the ith SS node, where $i = 1, 2, \ldots, N$).

As already mentioned, in order to identify an autonomous MOD system with rebalancing with a Jackson queueing model, we restricted the class of rebalancing policies to open-loop, "rebalancing promoting" policies. We will consider *closed-loop* policies in Section 4.3.

3.3. Problem formulation

Within our model, the optimization variables are the rebalancing rates ψ_i and α_{ij} of the rebalancing promoting policies. One might wonder in the first place if and when rebalancing is even required. Indeed, one can easily obtain that, for the case *without* rebalancing (George & Xia, 2011), $\lim_{m\to\infty} A_i(m) = \gamma_i/\gamma_S^{\max}$, for all $i \in S$, where γ_i is the relative utilization of node $i \in S$, $A_i(m)$ is the availability of vehicles at node $i \in S$ (see Section 2) and $\gamma_S^{\max} := \max_{i\in S}\gamma_i$. Hence, as m approaches infinity, the set of stations $B := \{i \in S : \gamma_i = \gamma_S^{\max}\}$ can have availability arbitrarily close to 1 while all other stations have availability *strictly* less than 1 *regardless* of m. In other words, without rebalancing, a MOD system will always experience passenger losses no matter how many vehicles are employed!

The above discussion motivates the need for rebalancing. The tenet of our approach is to ensure, through rebalancing, that the network is (on average) in balance, i.e., $A_i(m) = A_j(m)$ for all $i, j \in S$ (or, equivalently, $\gamma_i = \gamma_j$ for all $i, j \in S$, as implied by equation (4)). The motivation behind this philosophy is twofold (i) it provides a natural notion of service fairness, and (ii) it fulfills the intuitive condition that as m goes to $+\infty$ the availability of *each station* goes to one (since in this case $\gamma_i = \gamma_S^{\max}$ for all i in S). The objective then is to manipulate the rebalancing rates α_{ij} and ψ_i such that all the γ_i's in S are equal while minimizing the number of rebalancing vehicles on the road. Note that the average number of rebalancing vehicles traveling between station nodes i and j is given by $T_{ij}\alpha_{ij}\psi_i$. The rebalancing problem we wish to solve is then as follows:

Optimal Rebalancing Problem (ORP): Given an autonomous MOD system modeled as a closed Jackson network, solve

$$\begin{aligned} \underset{\psi_i,\alpha_{ij}}{\text{minimize}} \quad & \sum_{i,j} T_{ij}\alpha_{ij}\psi_i \\ \text{subject to} \quad & \gamma_i = \gamma_j \\ & \sum_j \alpha_{ij} = 1 \\ & \alpha_{ij} \geq 0,\ \psi_i \geq 0 \qquad i,j \in \{1,\dots,N\} \end{aligned} \tag{10}$$

where $\gamma_i = \frac{\pi_i}{\lambda_i+\psi_i}$ and π_i satisfies equation (1).

Note that to solve the ORP one would need to explicitly compute the relative throughputs π's using the traffic equation (1). This involves finding the 1-dimensional null space of a $\mathbb{R}^{N^2\times N^2}$ matrix, which becomes computationally expensive as the number of stations become large. Furthermore, the objective function and the constraints $\gamma_i = \gamma_j$ are nonlinear in the optimization variables. In the next section we show how to reduce the dimension of the problem to $\mathbb{R}^N$ and how the ORP can be readily solved as a minimum cost flow problem.

4. Optimal Rebalancing

4.1. Optimal rebalancing

In this section, we show how the ORP can be readily solved as a minimum cost flow problem. The main result of this section is presented in Theorem 4.3. We first present two lemmas that will be key in the proof of the theorem. The first lemma shows how the traffic equations (1) can be written only in terms of the SS nodes.

Lemma 4.1 (Folding of traffic equations)**.** *Consider an autonomous MOD system modeled as a closed Jackson network as described in Section 3.2. Then the relative throughputs π's for the SS nodes can be found by solving the* reduced *traffic equations*

$$\pi_i = \sum_{k\in S} \pi_k \tilde{p}_{ki} \ \textit{ for all } i \in S, \tag{11}$$

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

$$\pi_i = \pi_{\textit{Parent}(i)}\, \tilde{p}_{\textit{Parent}(i)\textit{Child}(i)} \ \textit{ for all } i \in I. \tag{12}$$

Proof. For each node $i \in \mathcal{N}$ equation (1) can be separated into SS nodes and IS nodes as follows

$$\pi_i = \sum_{j\in\mathcal{N}} \pi_j \tilde{r}_{ji} = \sum_{j\in S} \pi_j \tilde{r}_{ji} + \sum_{j\in I} \pi_j \tilde{r}_{ji}.$$

Consider, first, a SS node, i.e., consider i in S. Then one can write,

$$\pi_i = \underbrace{\sum_{j\in S} \pi_j \tilde{r}_{ji}}_{=0} + \sum_{j\in I} \pi_j \tilde{r}_{ji} = \sum_{\substack{j\in I \\ i=\text{Child}(j)}} \pi_j,$$

where $\sum_{j\in S} \pi_j \tilde{r}_{ji} = 0$ since SS nodes are connected exclusively by IS nodes. The last equality follows from the fact that whenever a child node of an IS node j is the SS node i, then $\tilde{r}_{ji} = 1$.

Consider, now, an IS node, i.e., consider i in I. Let Parent$(i) = k$ and Child$(i) = l$. Then one can write,

$$\pi_i = \sum_{j \in S} \pi_j \tilde{r}_{ji} + \underbrace{\sum_{j \in I} \pi_j \tilde{r}_{ji}}_{=0} = \pi_k \tilde{p}_{kl},$$

where $\sum_{j \in I} \pi_j \tilde{r}_{ji} = 0$ since IS nodes are connected *exclusively* to SS nodes, and the second equality follows from the fact that a single SS node feeds into each IS node with probability $\tilde{p}_{kl}$. This proves the second claim.

Collecting the results so far, we obtain, for each i in S,

$$\pi_i = \sum_{\substack{j \in I \\ i = \text{Child}(j)}} \pi_j = \sum_{\substack{j \in I \\ i = \text{Child}(j)}} \pi_{\text{Parent(j)}}\ \tilde{p}_{\text{Parent(j)}\, i} = \sum_{k \in S} \pi_k \tilde{p}_{ki},$$

which proves the first claim. □

Lemma 4.2. *For any rebalancing policy $\{\psi_i\}_i$ and $\{\alpha_{ij}\}_{ij}$, it holds for all $i \in S$*

1. $\gamma_i > 0$,

2. $(\lambda_i + \psi_i)\gamma_i = \sum_{j \in S} \gamma_j(\alpha_{ji}\psi_j + p_{ji}\lambda_j)$.

Proof. Let us prove the first part of the lemma. By assumption, the probabilities $\{p_{ij}\}_{ij}$ constitute an irreducible Markov chain. By equation (9), the probabilities $\{\tilde{p}_{ij}\}_{ij}$ lead to an irreducible Markov chain as well. The π vector satisfying equation (11) is the steady state distribution for the transition probabilities $\{\tilde{p}_{ij}\}_{ij}$ and by the Perron-Frobenius theorem, it is positive (Meyer, 2000, p. 673). In other words, $\pi_i > 0$ for all $i \in S$. By the definition of the relative utilizations γ_i (see equation (3)), we obtain the first part of the claim.

Let us now consider the second part of the lemma. Recall that, by assumption, $p_{ii} = 0$ and $\alpha_{ii} = 0$. By Lemma 4.1, for any $i \in S$, one can write

$$\begin{aligned}
\pi_i &= \sum_{j \in S} \pi_j \tilde{p}_{ji} \\
&= \sum_{j \in S} \pi_j \Big(\alpha_{ji} p_j + p_{ji}(1 - p_j) \Big) \\
&= \sum_{j \in S} \pi_j \Big(\alpha_{ji} \frac{\psi_j}{\lambda_j + \psi_j} + p_{ji} \frac{\lambda_j}{\lambda_j + \psi_j} \Big) \\
&= \sum_{j \in S} \frac{\pi_j}{\lambda_j + \psi_j} (\alpha_{ji}\psi_j + p_{ji}\lambda_j) \\
&= \sum_{j \in S} \gamma_j (\alpha_{ji}\psi_j + p_{ji}\lambda_j),
\end{aligned}$$

where the second equality follows from equation (9), the third equality follows from equation (8), and the last equality follows from equations (3), (6), and (7). This concludes the proof.

□

The next theorem (which represents the main result of this section) shows that we can *always* solve problem ORP by solving a low dimensional linear optimization problem.

Theorem 4.3 (Solution to problem ORP)**.** *Consider the linear optimization problem*

$$\begin{aligned} \underset{\beta_{ij}}{\text{minimize}} \quad & \sum_{i,j} T_{ij}\beta_{ij} \\ \text{subject to} \quad & \sum_{j\neq i}(\beta_{ij}-\beta_{ji}) = -\lambda_i + \sum_{j\neq i} p_{ji}\lambda_j \\ & \beta_{ij} \geq 0 \end{aligned} \tag{13}$$

The optimization problem (13) *is always feasible. Let* $\{\beta^*_{ij}\}_{ij}$ *denote an optimal solution. By setting* $\psi_i = \sum_{j\neq i}\beta^*_{ij}$*,* $\alpha_{ii} = 0$*, and, for* $j \neq i$*,*

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

one obtains an optimal solution to problem ORP.

Proof. First, we note that problem (13) is an uncapacitated minimum cost flow problem and thus is always feasible. Consider an optimal solution to problem (13), $\{\beta^*_{ij}\}_{ij}$, and set $\{\psi_i\}_i$ and $\{\alpha_{ij}\}_{ij}$ as in the statement of the theorem. We want to show that, with this choice, $\{\psi_i\}_i$ and $\{\alpha_{ij}\}_{ij}$ represent an optimal solution to the ORP. Since $\{\beta^*_{ij}\}_{ij}$ is an optimal solution to problem (13), then one easily concludes that $\{\psi_i\}_i$ and $\{\alpha_{ij}\}_{ij}$ are an optimal solution to problem

$$\begin{aligned} \underset{\psi_i,\alpha_{ij}}{\text{minimize}} \quad & \sum_{i,j} T_{ij}\alpha_{ij}\psi_i \\ \text{subject to} \quad & \lambda_i + \psi_i = \sum_j \alpha_{ji}\psi_j + p_{ji}\lambda_j \\ & \sum_j \alpha_{ij} = 1 \\ & \alpha_{ij} \geq 0, \quad \psi_i \geq 0 \end{aligned} \tag{14}$$

The objective is now to show that the constraint

$$\lambda_i + \psi_i = \sum_j \alpha_{ji}\psi_j + p_{ji}\lambda_j \tag{15}$$

is equivalent to the constraint

$$\gamma_i = \gamma_j. \tag{16}$$

Consider, first, the case where the $\{\alpha_{ij}\}_{ij}$ and $\{\psi_i\}_i$ satisfy constraint (15). Then, considering Lemma 4.2, one can write, for all i,

$$\Big(\sum_j \alpha_{ji}\psi_j + p_{ji}\lambda_j\Big)\gamma_i = \sum_{j\in S}\gamma_j(\alpha_{ji}\psi_j + p_{ji}\lambda_j). \tag{17}$$

Let $\varphi_{ij} := \alpha_{ji}\psi_j + p_{ji}\lambda_j$ and $\zeta_{ij} := \varphi_{ij}/\sum_j \varphi_{ij}$. (Note that $\sum_j \varphi_{ij} = \lambda_i + \psi_i > 0$ as $\lambda_i > 0$ by assumption.) Since $\alpha_{ii} = 0$ and $p_{ii} = 0$, one has $\zeta_{ii} = 0$. The variables $\{\zeta_{ij}\}_{ij}$'s can be considered as transition probabilities of an *irreducible* Markov chain (since, by assumption, the probabilities $\{p_{ij}\}_{ij}$ constitute an irreducible Markov chain). Then, one can rewrite equation (17) as $\gamma_i = \sum_j \gamma_j\,\zeta_{ij}$, which can be rewritten in compact form as $Z\,\gamma = \gamma$, where $\gamma = (\gamma_1, \ldots, \gamma_N)^T$ and Z is an irreducible, row stochastic matrix whose ith row is given by $[\zeta_{i1}, \zeta_{i2}, \ldots, \zeta_{i\,i-1}, 0, \zeta_{i\,i+1} \ldots \zeta_{iN}]$, where $i = 1, 2, \ldots, N$. Since Z is an irreducible, row stochastic matrix, by the Perron-Frobenius theorem (Meyer, 2000, p. 673), the eigenspace associated with the eigenvalue equal to 1 is one-dimensional, which implies that the equation $Z\,\gamma = \gamma$ has a unique solution given by $\gamma = (1, \ldots, 1)^T$, up to a scaling factor. This shows that $\gamma_i = \gamma_j$ for all i, j. Conversely, assume that $\{\alpha_{ij}\}_{ij}$ and

$\{\psi_i\}_i$ satisfy constraint (16). Considering Lemma 4.2 (note, in particular, that $\gamma_i > 0$), since $\gamma_i = \gamma_j$ for all i, j, then one immediately obtains that $\{\alpha_{ij}\}_{ij}$ and $\{\psi_i\}_i$ satisfy constraint (15). Hence, we can equivalently restate problem (14) as problem (10), which proves the claim. □

Remarkably, problem (13) has the same form as the linear optimization problem in Pavone et al. (2012) used to find rebalancing policies within a *fluidic* model of an autonomous MOD system. In this respect, the analysis of our paper provides a *theoretical foundation* for the fluidic approximation performed in Pavone et al. (2012), which can be viewed as the limit of a sequence of appropriately scaled queueing models. Note that in the fluidic model, customers are allowed to queue up at stations. We obtain the same linear program because it turns out that the equilibria for the fluidic model correspond to each station having zero customers waiting, which is exactly the situation in our loss model.

The importance of Theorem 4.3 is twofold: it allows us to efficiently find an optimal open-loop, rebalancing promoting policy, and it enables the computation of quality of service metrics (namely, vehicle availability) for autonomous MOD systems as shown next.

4.2. Computation of performance metrics

By leveraging Theorem 4.3, one can readily compute performance metrics (i.e. vehicle availability) for an autonomous MOD system. First, we compute an optimal solution to the ORP using Theorem 4.3, which involves solving a linear optimization problem with N^2 variables. Next, we compute the relative throughputs π's using Lemma 4.1. Finally, we apply a well-known technique called mean value analysis (MVA) (Reiser & Lavenberg, 1980) in order to avoid the explicit computation of the normalization constant in equation (4), which is prohibitively expensive for large numbers of vehicles and stations. The MVA algorithm is an iterative algorithm to calculate the mean waiting times $W_i(n)$ and the mean queue lengths $L_i(n)$ at each node i of a closed separable system of queues, where $n = 1, \ldots, m$ is the numbers of vehicles over which the algorithm iterates. For the Jackson model in Section 3.2, subject to the initial conditions $W_i(0) = L_i(0) = 0$, the equations for MVA read as:

- $W_i(n) = \frac{1}{\mu_i(1)} = T_{\text{Parent}(i)\,\text{Child}(i)}$ for all $i \in I$,
- $W_i(n) = \frac{1}{\mu_i}(1 + L_i(n-1)) = \frac{1}{\tilde{\lambda}_i}(1 + L_i(n-1))$ for all $i \in S$,
- $L_i(n) = \frac{n\pi_i W_i(n)}{\sum_{j\in\mathcal{N}} \pi_j W_j(n)}$ for all $i \in \mathcal{N}$.

Finally, the throughput (or mean arrival rate of vehicles) to each station is given by Little's theorem (Bertsekas et al., 1992, p. 152): $\Lambda_i(m) = L_i(m)/W_i(m)$ for all $i \in S$. Combining equations (3), (2) and (4), one readily obtains the availability at each station as $A_i(m) = \Lambda_i(m)/\tilde{\lambda}_i$.

This procedure scales well to a large number of stations and vehicles, and is applied in Section 5 to real-world settings involving hundreds of stations and thousands of vehicles, to assess the potential performance of an autonomous MOD system in New York City.

The rebalancing promoting policy considered so far, while providing useful insights into the performance and operations of an autonomous MOD system, is ultimately an open-loop policy and hence of limited applicability. In the next section, we use insights gained from the ORP to formulate a *closed-loop* rebalancing policy for the robotic vehicles that appears to perform well in practice.

4.3. Real-time rebalancing policy

In this section, we introduce a practical real-time rebalancing policy that can be implemented on real autonomous MOD systems. In reality, customers arriving at a station would wait in line rather than leave the system immediately (as in the loss model) if a vehicle is not available. In the meantime, information could be collected about the customer's destination

and used in the rebalancing process. Let $v_i^{\text{own}}(t)$ be the number of vehicles "owned" by station i, that is, vehicles that are at station i, on their way to station i, or will be on their way to station i. We can write $v_i^{\text{own}}(t) = v_i(t) + \sum_j v_{ji}(t) + \sum_j \tilde{c}_{ji}(t)$, where $v_i(t)$ is the number of vehicles at station i, $v_{ji}(t)$ is the number of vehicles enroute from station j to i, and $\tilde{c}_{ji}$ is the number of passengers at station j that are about to board an available vehicle to station i. Note that $\sum_i \tilde{c}_{ji}(t) \leq v_j(t)$. Let $v_i^e(t) := v_i^{\text{own}}(t) - c_i(t)$ be the number of excess vehicles there will be at station i, where $c_i(t)$ is the number of customers at station i. The total number of excess vehicles is given by

$$\begin{aligned}\sum_i v_i^e(t) &= \sum_i \Big(v_i(t) + \sum_j v_{ji}(t) + \sum_j \tilde{c}_{ji}(t) - c_i(t) \Big) \\ &= m + \sum_i \min\{v_i(t), c_i(t)\} - \sum_i c_i(t) = m - \sum_i \max\{c_i(t) - v_i(t), 0\}.\end{aligned} \tag{18}$$

The second equality replaces $v_i(t) + \sum_i v_{ji}(t)$ with the total number of vehicles and asserts that in the current time step, either all of the customers or all the vehicles will leave the station. The last equality is obtained by considering both cases when $c_i(t) \geq v_i(t)$ and when $c_i(t) < v_i(t)$.

Through rebalancing, we may wish to distribute these excess vehicles evenly between all the stations, in which case each station will have no less than $v_i^d(t)$ vehicles, given by $v_i^d(t) = \lfloor (m - \sum_i \max\{c_i(t) - v_i(t), 0\})/N \rfloor$. Accordingly, every $t_{\text{hor}} > 0$ time periods, the number of vehicles to rebalance from station i to j, num_{ij}, is computed by solving the linear integer optimization problem

$$\begin{aligned} \underset{\text{num}_{ij}}{\text{minimize}} \quad & \sum_{i,j} T_{ij} \text{num}_{ij} \\ \text{subject to} \quad & v_i^e(t) + \sum_{j \neq i} (\text{num}_{ji} - \text{num}_{ij}) \geq v_i^d(t) \;\; \text{for all } i \in S \\ & \text{num}_{ij} \in \mathbb{N} \;\; \text{for all } i, j \in S, i \neq j \end{aligned} \tag{19}$$

This rebalancing policy takes all the current information known about the system and sets the rebalancing rates (in this case, the number of rebalancing vehicles) so that excess vehicles are distributed evenly to all the stations. This is in part inspired by the optimization problem in Theorem 4.3. It can be shown that the constraint matrix is totally unimodular, and the problem can be solved as a linear program, as the resulting solution will be necessarily integer-valued (Pavone et al., 2012, Section 5). The rebalancing policy presented here is closely related to the one presented in Pavone et al. (2012), the main difference being the inclusion of the current customers in line within the optimization process.

The real-time rebalancing policy will be used in Section 5 to validate the vehicle availability performance criterion.

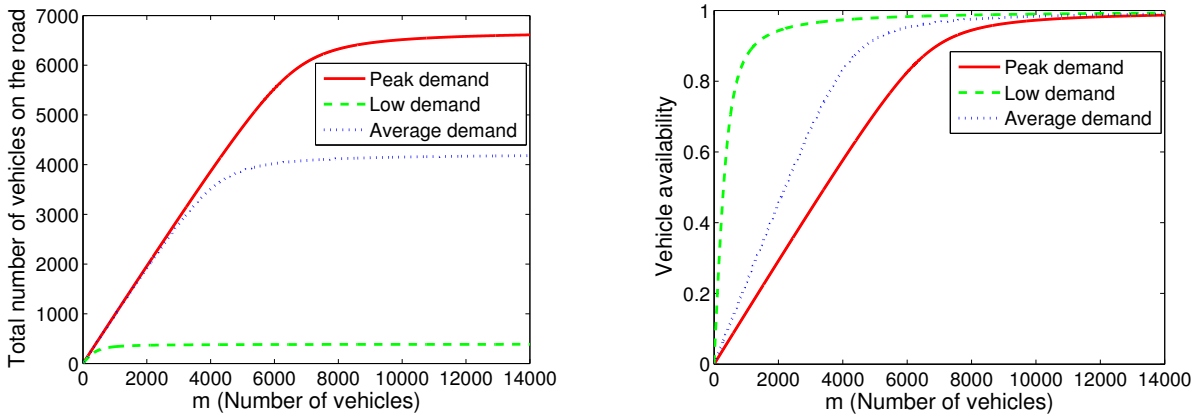
5. Case Study: Autonomous MOD in Manhattan

In this section we apply our availability analysis using the loss model to see how many robotic vehicles in an autonomous MOD system would be required to replace the current fleet of taxis in Manhattan while providing quality service at current customer demand levels. (We recently leveraged the theoretical results in this paper to perform a similar study for Singapore in Spieser et al. (2014).) We then discuss the impact of the placement/number of stations on the case study and ways to further improve the performance of the system.

5.1. Case study overview and results

In 2012, over 13,300 taxis in New York City made over 15 million trips a month or 500,000 trips a day, with around 85% of trips within Manhattan. Our study used taxi trip data collected on March 1, 2012[2] consisting of 439,950 trips within Manhattan. First, trip origins and destinations are clustered into $N = 100$ stations throughout the city using k-means clustering. The resulting locations of the stations are such that a demand is on average less than 300m from the nearest station, or approximately a 3-minute walk. The system parameters λ_i, p_{ij}, and T_{ij} are estimated for each hour of the day using trip data between each pair of stations with Laplace smoothing. Some congestion effects are implicitly taken into account in the computation of T_{ij}, which uses the Manhattan distance and an average speed estimated from the data.

Vehicle availability is calculated for 3 cases - peak demand (29,485 demands/hour, 7-8pm), low demand (1,982 demands/hour, 4-5am), and average demand (16,930 demands/hour, 4-5pm). For each case, vehicle availability is calculated as a function of the fleet size using MVA techniques. In addition to vehicle availability, the average number of vehicles on the road is given by $\sum_{i,j} \Lambda_i \tilde{p}_{ij} T_{ij}$ by Little's theorem (Bertsekas et al., 1992, p. 152). The results are summarized in Figure 2. From Figure 2(a), we see that with 7,000 vehicles, the average number of vehicles traveling is 6,060,



(a) (b)

Fig. 2. 2(a): Average number of vehicles on the road as a function of system size for 100 stations in Manhattan. 2(b): Vehicle availability as a function of system size. Availability and number of vehicles on the road are calculated for peak demand (7-8pm), low demand (4-5am), and average demand (4-5pm).

corresponding to a vehicle utilization rate of 86%. This utilization rate drops off significantly with more than 8,000 vehicles in the system, as the number of vehicles on the road levels off. This suggests that the operating point of the system should be between 7,000 and 8,000 vehicles at peak demand. From Figure 2(b), for high vehicle availability (say, 95%), we would need around 8,000 vehicles ($\sim$70% of the current fleet size operating in Manhattan, which, based on taxi trip data, we approximate as 85% of the total taxi fleet) at peak demand and 6,000 vehicles at average demand. The results suggest that an autonomous MOD system with 8,000 vehicles would be able to meet 95% of the taxi demand in Manhattan, assuming 5% of passengers are impatient and are lost when a vehicle is not immediately available. However, in a real system, passengers would wait in line for the next vehicle rather than leave the system, thus it is important to determine how vehicle availability relates to customer waiting times. We characterize the customer waiting times through simulation, using the real-time rebalancing policy described in Section 4.3. Figure 4 shows a snapshot of the simulation environment with 100 stations and 8,000 vehicles (see Extension 1 for a clip of the simulation). Simulation are performed with discrete time steps of 2 seconds and a simulation time of 24 hours. The time-varying system parameters λ_i, p_{ij}, and average speed are piecewise constant, and change each hour based on values estimated from the taxi data. Travel times T_{ij} are based on average speed

[2] The data is courtesy of the New York City Taxi & Limousine Commission.

and Manhattan distance between i and j, and rebalancing is performed every 15 minutes. Three sets of simulations are performed for 6,000, 7,000, and 8,000 vehicles, and the resulting average waiting times are shown in Figure 3.

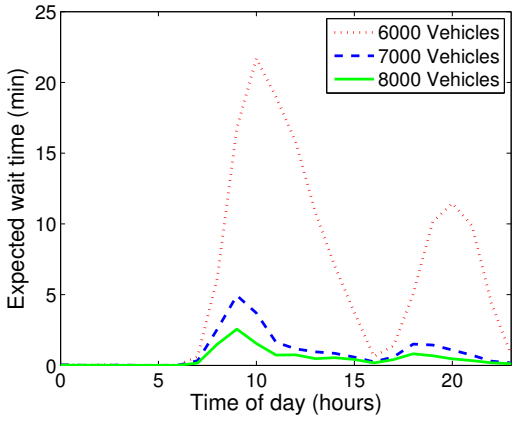


Fig. 3. Average customer wait times over the course of a day, for systems of different sizes.

Figure 3 shows that for a 7,000 vehicle fleet, the peak averaged wait time is less than 5 minutes (9-10am) and for 8,000 vehicles, the average wait time is only 2.5 minutes. The simulation results show that high availability (90-95%) does indeed correspond to low customer waiting time and that an autonomous MOD system with 7,000 to 8,000 vehicles (60-70% of the size of the current taxi fleet) can provide adequate service with current taxi demand levels in Manhattan.

5.2. Placement and number of stations

In the previous analysis, the locations of the stations were determined by a k-means clustering algorithm based on historical demand data. We can evaluate the adequacy of these locations by looking at the waiting time distribution throughout the city. Figure 5(a) shows the distribution of waiting time for 100 stations and 8,000 vehicles at peak demand (between 9 and 10 am). For most stations, the wait times are very low (0 – 2 minutes) but a few stations have wait times much higher than the average. This is in part due to the geographic placement of the stations and in part due to the tuning of the real-time rebalancing policy. For example, in Figure 4, the station just northwest of central park sees high customer demand but is geographically isolated from midtown and lower Manhattan, where most excess vehicles are. During the rebalancing optimization (19), the number of rebalancing vehicles sent to this station takes into account the number of vehicles that will eventually reach the station, but does not take into account the length of time it will take to get there. Due to the longer travel time, there is an effective delay for rebalancing vehicles to reach the station, resulting in longer wait times. This delay can be reduced by tuning the parameters $v_i^d(t)$ to send more rebalancing vehicles to the few isolated stations with high customer demand.

The number of stations, n, may also impact the waiting time distribution. To gain insight into the effects of different n, additional simulations were performed with 8,000 vehicles at peak demand (9–10 am). We studied a range of values for n from 50 to 200. For each n, the stations were placed using k-means, and 10 simulations were performed. Figure 5(b) shows the maximum wait time across all stations for different n values. It is clear that a higher number of stations tends to yield a lower maximum wait time. However, as previously stated, the locations of the stations are just as important as the number of stations, and can significantly impact the maximum wait times. The spike in wait time at 120 stations in Figure 5(b) can be attributed to poor placement of the stations (a local optimum in the k-means algorithm). Overall, this analysis illustrates that the customer wait times can serve as a metric to determine the optimal number of stations and the locations of the stations.

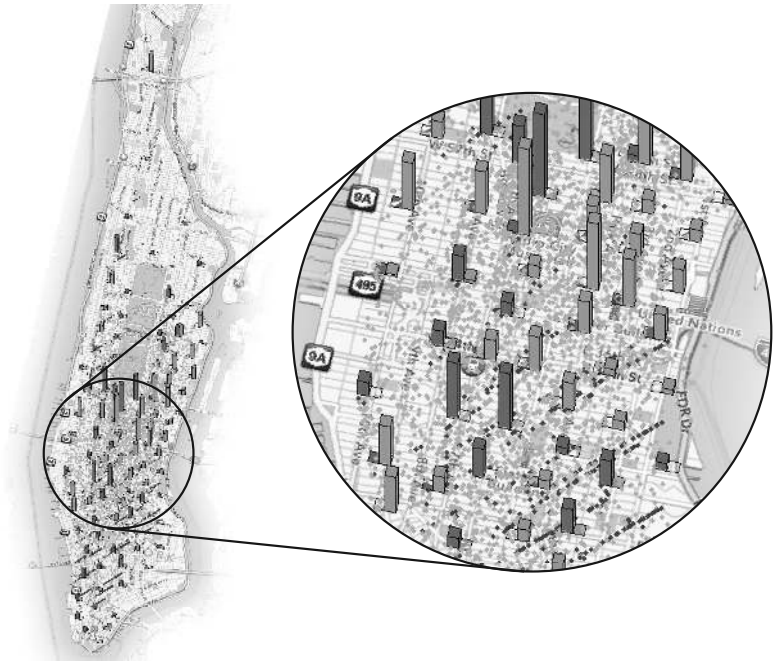


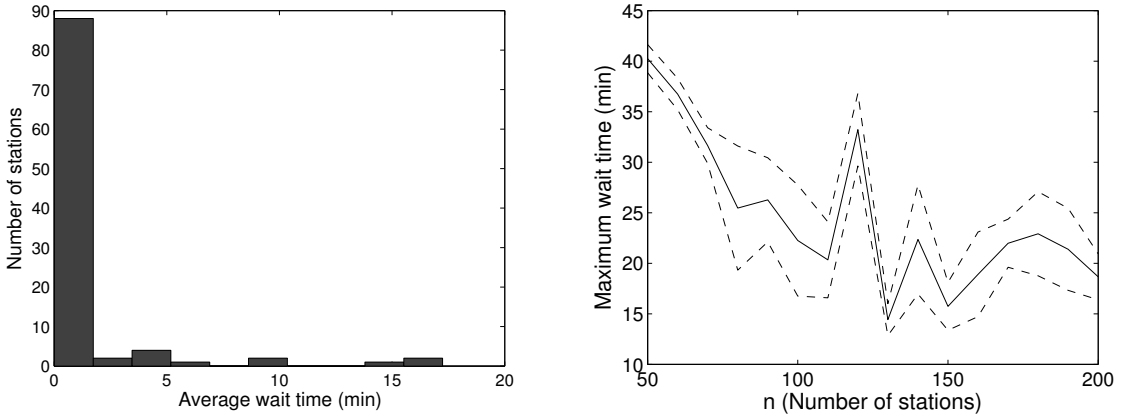
Fig. 4. Simulation environment with 100 stations in Manhattan. Dark bars indicate waiting customers, light bars indicate available vehicles, gray dots are vehicles traveling with passengers and black dots are rebalancing vehicles.

In practice, an autonomous MOD system may not necessarily need fixed infrastructure to serve as stations. Indeed, in a system where customer requests are made through a smartphone app and vehicles provide door-to-door service, the concept of a "station" may simply be a geographical region where vehicles in the vicinity service requests that arrive within the region. Hence, the number of "stations" and their locations may be changed to dynamically adapt to the current demand distribution. Demand prediction can be achieved with Bayesian nonparametric techniques such as Dirichlet process mixture models (Kulis & Jordan, 2012; Campbell et al., 2013). In this way, the system can simultaneously predict demand and route vehicles to provide service, which can be an effective way of handling time-varying customer demands. This is an interesting avenue of future work.

6. Hardware Experiments

To further compare the availability metric from the queueing model with customer wait times in a real system, we developed a small-scale testbed consisting of mobile robots driving in a "mock" city with 4 stations (see Figure 6 and Extension 2). Customers are generated by a central computer according to a Poisson process with rate λ_i and routing distribution p_{ij}, $i, j \in \{1, 2, 3, 4\}$. Once generated, customers form first-come-first-serve queues at each station to wait for vehicles (as opposed to leaving immediately if a vehicle is not at the station). The vehicles are modified Pololu m3pi line-following robots, and communicate with the central computer through a shared WiFi network. When a customer arrives at a station, a vehicle takes the customer to the desired destination by following the black lines along a pre-programmed path. A Vicon motion capture system is used to implement collision avoidance between the vehicles, which (i) allows vehicles to stop behind a stopped vehicle traveling in the same direction, and (ii) allow vehicles to traverse intersections using a first-in-first-out rule. Each vehicle takes around 15 seconds to travel to an adjacent station and 22 seconds to travel to the station on the opposite corner of the city. Eight vehicles were used in the experiment.

The experiments were performed by first randomly generating a demand intensity λ^0 and routing distribution p^0_{ij} that yielded a stable system in terms of customer wait times in simulation under the real-time rebalancing policy. Three runs were performed with demand intensity at $0.5\lambda^0$, λ^0, and $2\lambda^0$, while the routing distribution was held fixed. The real-time



(a) (b)

Fig. 5. 5(a) shows a histogram of the wait times of all 100 stations. 5(b) shows the maximum wait time in the system for systems with different n (number of stations). For 16 values of n ranging from 50 to 200, the simulation (8,000 vehicles, peak demand) was performed 10 times. The solid line denotes the mean of the 10 runs and the dashed lines denote one standard deviation above and below the mean.

rebalancing policy in Section 4.3 was implemented with a time horizon of 2 seconds. Figure 7(a) shows the corresponding availability plots for reference (if the same demands were applied to the loss model) and Figure 7(b) shows the number of waiting customers in each run. The number of idle vehicles is shown in Figure 7(c). In these experiments, the high demand run (corresponding to 57% availability) was unstable, as the customers arrived faster than they can be serviced. The medium demand case (corresponding to 66% availability) had on average one customer waiting at one of the four stations, with an average wait time of 6.4 seconds. The low demand case (corresponding to 69% availability) only had an average of 0.07 waiting customers with an average wait time of less than 1 second. It is interesting to note that the low demand run had very low wait times, but its corresponding availability in the loss model was only 69%. This reveals a limitation of the loss model, where rebalancing is implemented by *randomly* generating virtual customers according to a Poisson process. The real-time rebalancing policy only rebalances vehicles when needed, and hence is able to provide a better quality of service (low wait times). Finally, we see from Figure 7(c) that in the medium demand case, the average number of idle vehicles is only 1.75, which corresponds to a 78% vehicle utilization rate. This high utilization rate suggests that the system is near its stability limit and is comparable to that of the Manhattan analysis (see Figure 2(a)) at peak demand with 8,000 vehicles. Future efforts will focus on scaling up the testbed to more than 20 vehicles so that congestion effects can be visualized and studied.

7. A Mean Value Analysis Approach to the Analysis of Congestion Effects

The queueing model described in Section 3 does not consider congestion effects (roads are modeled as *infinite* server queues, so the travel time for each vehicle is independent of all other vehicles). However, if too many rebalancing vehicles travel on a route that is already congested, they can cause a traffic jam and decrease throughput in the entire system. Hence, in some scenarios, adding robotic vehicles to improve the quality of service might indeed have the opposite effect.

In this section, we propose an approach to study congestion effects that leverages our queueing-theoretical setup. The key idea is to change the infinite-server road queues to queues with a *finite* number of servers, where the number of servers on each road represents the *capacity* of that road. This road congestion model is similar to "vertical queueing" models that have been used in congestion analysis for stop-controlled intersections (Madanat et al., 1994) and for traffic assignment (Huang & Lam, 2002). In traditional traffic flow theory (Lieu, 2003), the flow rate of traffic increases with the density of vehicles up to a critical value at which point the flow decreases, marking the beginning of a traffic jam. By letting the

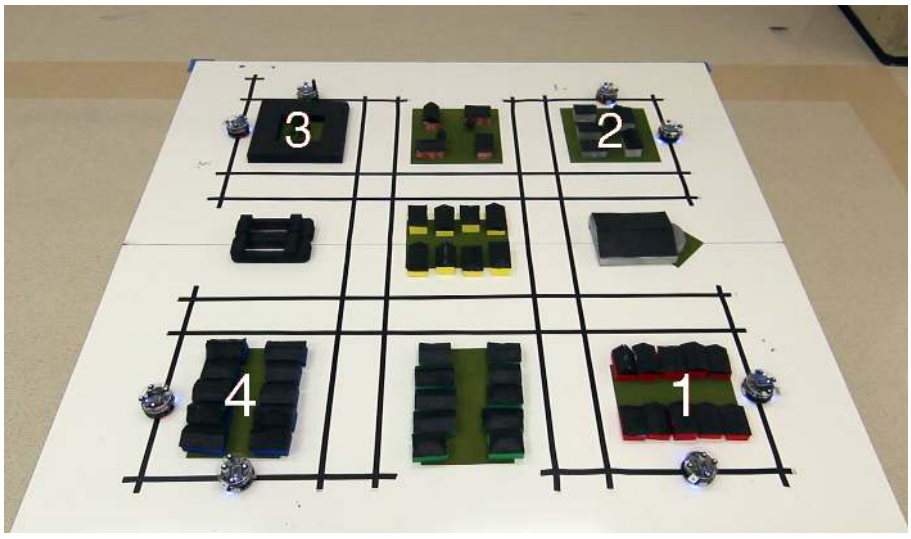


Fig. 6. Hardware testbed with 4 stations and 8 vehicles.

number of servers represent the critical density of the road, the queueing model becomes a good model for traffic flow up to the point of congestion.

Remarkably, the Jackson network model presented in Section 3 can be extended to the case where roads are modeled as finite-server queues; furthermore, the results presented in Section 2 are equally valid. However, the travel times are no longer simply equal to the inverse of the service rates of the road queues, which significantly complicates the formulation of an analogue of problem ORP. While the issue of finding optimal rebalancing policies in the presence of congestion effects is left for future research, in this paper we show how *given* a rebalancing policy one can compute performance metrics such as vehicle availability (for example, one can study the effects of congestion on the performance of the rebalancing policies considered in Section 4).

In our approach, we first model the road network as an abstract queueing network with finite-server road queues, then we apply an extended version of the MVA algorithm for finite-server queues, the details of which are presented in Reiser & Lavenberg (1980).

7.1. Mapping physical roads into finite-server road queues

The main difficulty in mapping the capacities of the road network into the number of servers of the queueing model (or "virtual" capacities, denoted by m_{ij}) is that trips from different origins and destinations may share the *same* physical road.

As a simple example, consider the 3-station network shown in Figure 8. Let q_{ij} represent the maximum number of vehicles that can travel on the road between station i and station j without causing significant congestion. Let m_{ij}, the number of servers between i and j, represent the number of vehicles that can travel between i and j before delays occur due to queueing. In the simple network, to go from station i to station k, one must pass through station j. Hence, one has the following consistency constraints

$$m_{ij} + m_{ik} \leq q_{ij}, \quad m_{jk} + m_{ik} \leq q_{jk}. \tag{20}$$

To maximize the overall road usage, we can define a quadratic objective that seeks to minimize the difference between the real road capacities and the sum of the virtual road capacities:

$$\min_{m_{ij},m_{jk},m_{ik}} (m_{ij} + m_{ik} - q_{ij})^2 + (m_{jk} + m_{ik} - q_{jk})^2$$

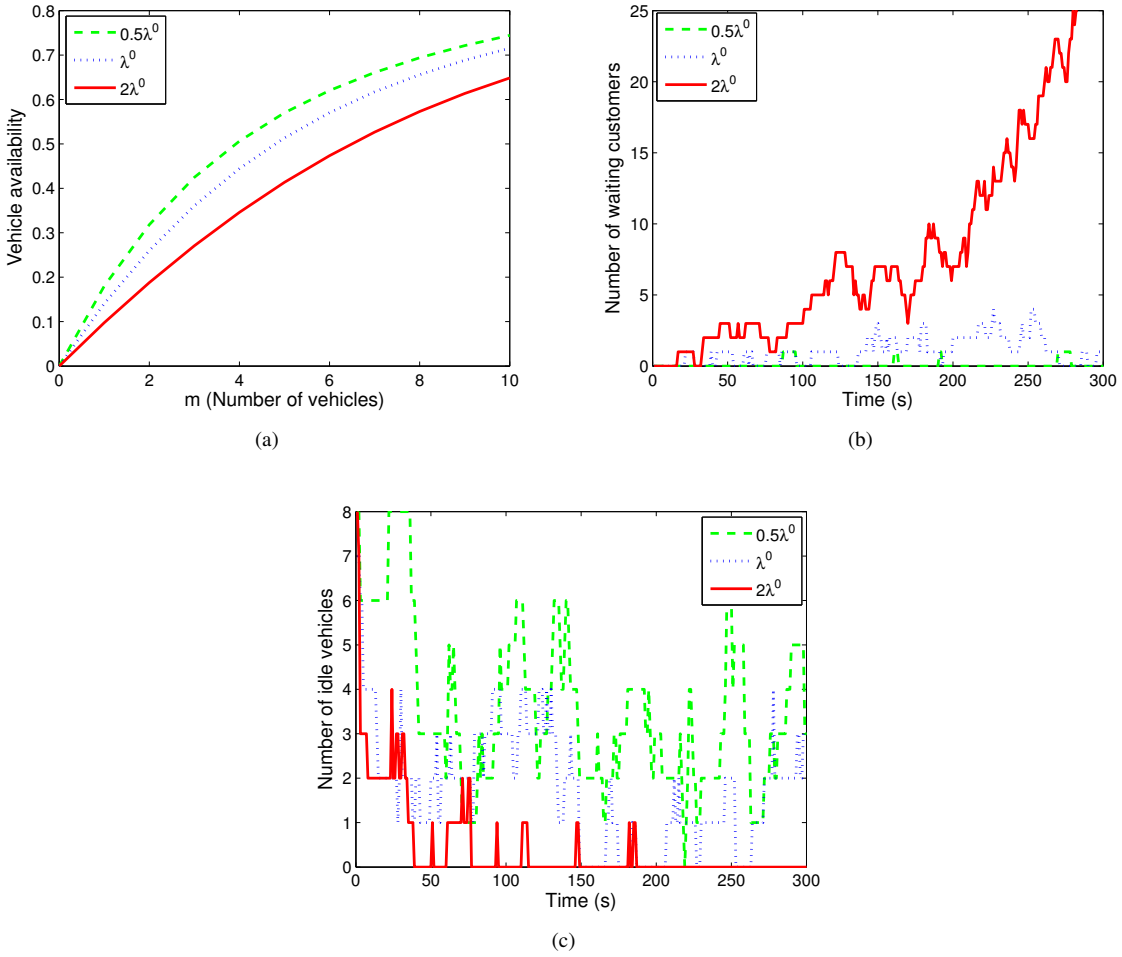


Fig. 7. 7(a) shows the availability curves for the three test cases. 7(b) shows the total number of customers waiting at the four stations for the three test cases. 7(c) shows the total number of idle vehicles at the four stations.

However, this optimization problem, along with the constraints (20), does not yield a unique solution because nothing is assumed about the relative usage rates of the road queues. If relative road usage is known, the m_{ij}'s can be assigned proportional to the amount of traffic between each pair of stations that use the road. Let π_{ij} be the relative throughput of the road queue between station i and j, consistent with the earlier definition. *Heuristically*, the throughputs $\{\pi_{ij}\}_{ij}$ may be obtained from the arrival rates and travel patterns of passengers or from the analysis of a given rebalancing policy assuming no congestion (according to the procedure discussed in Section 4.2). For the simple example, one can write

$$m_{ik} \leq \frac{q_{ij}\pi_{ik}}{\pi_{ik}+\pi_{ij}}, \quad m_{ik} \leq \frac{q_{jk}\pi_{ik}}{\pi_{ik}+\pi_{jk}}. \tag{21}$$

Similar constraints can be written for m_{ij} and m_{jk} so that (20) is satisfied.

For a general road network, let B_{ij} be the set of possible non-cyclic paths from station i to j (assuming no back tracking) and $C_{b_{ij}}$ be the set of road segments along path $b_{ij} \in B_{ij}$. The number of possible paths from i to j is given by $|B_{ij}|$. Let a_{ij}^c denote the fraction of trips from i to j that go through road segment $c = \{\text{origin}, \text{destination}\}$, where $c \in C_{b_{ij}}$. Denote by q_c the capacity of road segment c. For trips going through multiple road segments, the virtual road capacity is

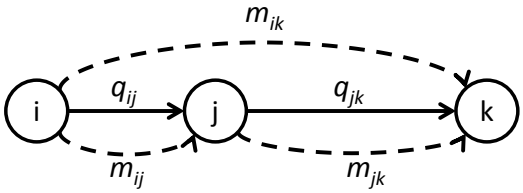


Fig. 8. A simple 3-station example showing the procedure of mapping physical roads into finite-server road queues.

determined by the segment with the lowest capacity. One can then consider as virtual road capacities:

$$m_{ij} = \sum_{b \in B_{ij}} \min_{c \in C_{b_{ij}}} \left\{ \frac{q_c a_{ij}^c \pi_{ij}}{\sum_{k,l,\text{such that } c \in C_{b_{kl}}} a_{kl}^c \pi_{kl}} \right\}.$$

However, this formulation can lead to over-conservative results for example if one road segment along one of the paths fills to capacity while the road segments of other paths are relatively unused. Thus, the routing scheme, represented by a_{ij}^c, is central to the characterization of the mapping from the road network into the queueing network model. One can further optimize the routing scheme using methods similar to dynamic traffic assignment (DTA) (Janson, 1991) to maximize the virtual capacities of the queueing model. The combination of system rebalancing and route optimization to minimize congestion is a very interesting avenue of future research.

An alternative approach, described in the next section, can be very useful for evaluating the impact of rebalancing on congestion. We make the key observation that the throughput of each station (rate of vehicles leaving each station) is the total arrival rate of customers multiplied by the availability, so in systems with a high quality of service, the actual throughput Λ_i can be closely approximated by the arrival rates $\tilde{\lambda}_i$. In the next section we will apply this approach to study congestion effects for autonomous MOD systems on a very simple transportation network.

7.2. Numerical study of congestion effects

In this section we use a simple 9-station road network (shown in Figure 9) to illustrate the impact of rebalancing vehicles on congestion.

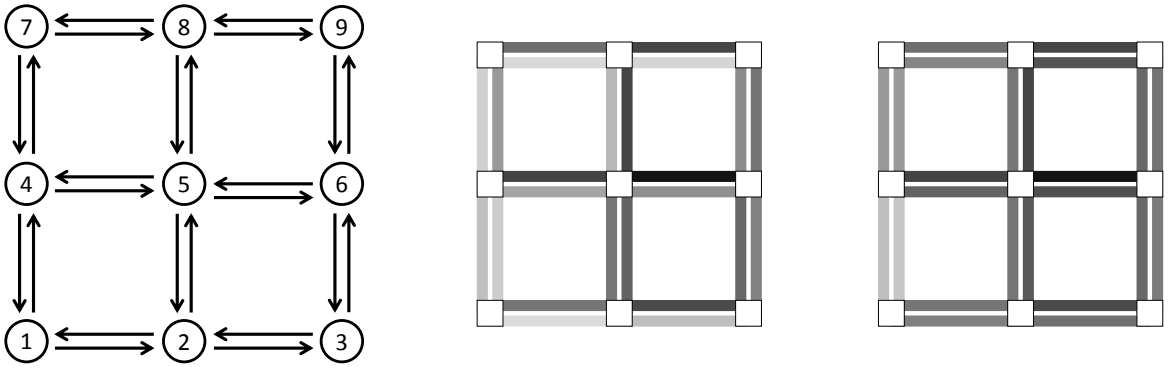


Fig. 9. Left: Layout of the 9-station road network. Each road segment has a capacity of 40 vehicles in each direction. Center: A randomly generated system on the 9-station road network without rebalancing. The shade on each road segment indicates the level of congestion, where white is no congestion, and black is heavy congestion. Right: The same road network with rebalancing vehicles.

The stations are placed on a square grid, and joined by 2-way road segments each of which is 0.5 km long. Each road consists of a single lane, with a critical density of 80 vehicles/km.[3] This means that the capacity of each road segment c is $q_c = 40$ vehicles. Each vehicle travels at 30 km/h (8.33 m/s) in free flow, which means the travel time along each road segment is 1 minute in free flow.

To gain insight into the general system behavior, a variety of systems with different levels of imbalance must be studied. First, arrival rates and routing distributions are randomly generated and rebalancing rates are computed using (13). In steady state, the fraction of vehicles in each road queue ij is given by $\pi_i \tilde{p}_{ij}$ (Lemma 4.1). If we assume 100% availability ($A_i = 1$), the expected rate of vehicles entering each road queue is given by $\Lambda_{ij} = \lambda_i p_{ij}$. Using Little's theorem, the expected number of vehicles on each road queue is given by $L_{ij} = \Lambda_{ij} T_{ij}$. The availability assumption can be justified by the fact that a real system would operate within the regime of high availability and that the number of vehicles on the road gets very close to L_{ij} as availability increases. Similarly, the expected number of rebalancing vehicles on each road queue is given by $L_{ij}^{\text{reb}} = \beta_{ij} T_{ij}$.

To map the queueing network onto the road network, we adopt a similar procedure as the one used to estimate m_{ij} in Section 7.1. Recall that B_{ij} is the set of paths from station i to station j. We adopt the routing strategy that uniformly distributes vehicles from i to j along each path $b_{ij} \in B_{ij}$. The number of vehicles that go through each road segment, L_c^{road}, is then the sum of the number of vehicles from each station to every other station that pass through the road segment, given by

$$L_c^{\text{road}} = \sum_{i,j,\text{such that } c \in C_{b_{ij}}} \frac{L_{ij}}{|B_{ij}|}.$$

Note that for stability, $L_c^{\text{road}} < q_c$. The road utilization is given by $\rho_c^{\text{road}} = L_c^{\text{road}}/q_c$.

Figure 10(a) plots the vehicle and road utilization increases due to rebalancing for 500 randomly generated systems. The x-axis shows the ratio of rebalancing vehicles to passenger vehicles on the road, which represents the inherent imbalance in the system. The dots represent the increase in average road utilization due to rebalancing and the x's represent the utilization increase in the most congested road segment due to rebalancing. It is no surprise that the average road utilization rate is a linear function of the number of rebalancing vehicles. However, remarkably, the maximum congestion increases are much lower than the average, and are in most cases, zero. This means that while rebalancing generally increases the number of vehicles on the road, rebalancing vehicles mostly travel along less congested routes and rarely increase the maximum congestion in the system. This can be seen in Figure 9 right, where rebalancing clearly increases the number of vehicles on many roads but not on the most congested road segment (from station 6 to station 5).

In roughly 10% of the systems simulated, the maximum road utilization in the system increased from rebalancing (Figure 10(a)). This may cause heavy congestion in systems where congestion is already prevalent. Though different routing strategies may help decrease the maximum road utilization, we can also adjust the rebalancing rates by using a "corrected" travel time in the rebalancing optimization (13) that takes into account the congestion in the system. For this example, the travel time is calculated using the Bureau of Public Roads model, a simple relation between road utilization and travel time (Bureau of Public Roads, 1964). The corrected travel time along road segment c is

$$T_c^{\text{cor}} = T_c(1 + 0.15\bar{\rho}_c^4),$$

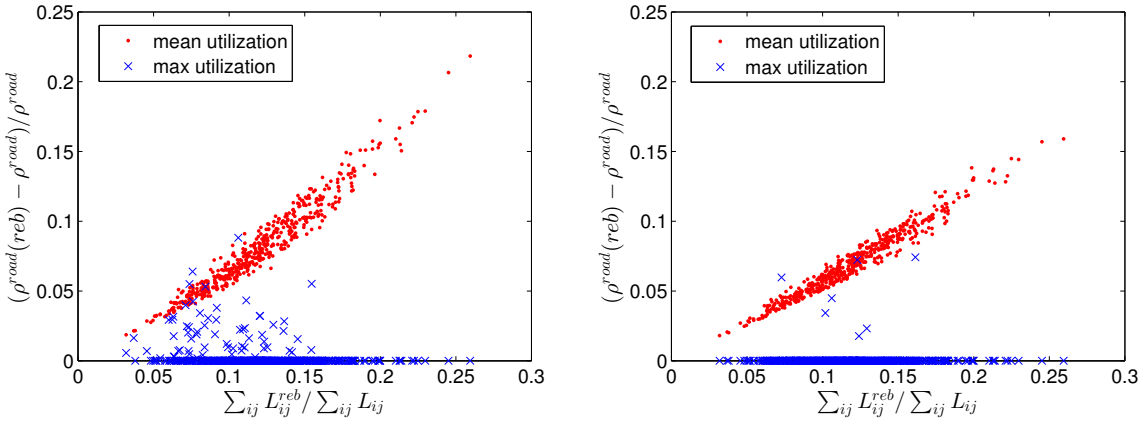
where T_c is the free flow travel time, $\bar{\rho}_c = L_c^{\text{road}}/\bar{L}_c^{\text{road}}$, and $\bar{L}_c^{\text{road}}$ is the mean number of vehicles on each road segment. The intuition here is to penalize rebalancing trips on routes with higher-than-average utilization. The corrected mean travel

[3] If each vehicle is 5 m, this critical density represents a vehicle-to-vehicle separation of 1.5 car-lengths.

time from station i to j is then

$$T_{ij}^{\text{cor}} = \sum_{b_{ij} \in B_{ij}} a_{ij}^{b_{ij}} \sum_{c \in C_{b_{ij}}} T_c,$$

where $a_{ij}^{b_{ij}}$ is the fraction of trips from i to j taking path b_{ij}. In this example, since trips are divided evenly between all paths, $a_{ij}^{b_{ij}} = 1/|B_{ij}|$. The rebalancing rates β_{ij} are then recalculated from (13) using T_{ij}^{cor} instead of T_{ij}.



(a) (b)

Fig. 10. 10(a): The effects of rebalancing on congestion on a 9-station road network illustrated using 500 randomly generated systems with different arrival rates and routing distributions. The x-axis is the ratio of rebalancing vehicles to passenger vehicles on the road. The y-axis is the fractional increase in road utilization due to rebalancing. The dots show the mean increase in road utilization. The crosses show the increase in utilization of the most congested road segment. 10(b): The same 500 systems with rebalancing rates calculated using a "corrected" travel time that takes into account road utilization.

Figure 10(b) shows the impact of rebalancing on road utilization with the rebalancing rates adjusted using the method described above. Comparing with Figure 10(a), we see that the new rebalancing scheme reduced the number of systems with increased maximum road utilization. Of the 500 simulated systems, only 7 (1.4%) saw an increase in the maximum road utilization. Remarkably, the mean road utilization increase due to rebalancing was reduced as well. This shows that by solely adjusting the rebalancing parameters β_{ij} we can almost always prevent additional congestion in the most congested parts of the system.

We further validate these results by considering a larger 7×7 grid road network consisting of 16 stations (one station every 2 blocks). The larger grid permits more routes between stations and is more representative of a real road network than the 3×3 grid 9-station example. The same analysis is carried out for this road network and shown in Figure 11. Rather than focusing on the most congested road segment as in the 9-station example, we look at the 10 most congested road segments in the network, and whether rebalancing increases the congestion on these segments. From Figure 11(c), we see that the increases in utilization on the 10 most congested roads is far less than the average utilization increase (an even more promising result than the 9-station case). In Figure 11(d), the rebalancing correction scheme is applied and the number of systems with increased top 10 road utilization (crosses) is reduced from 143 (28.6%) to 25 (5%). This is achieved at the expense of slightly higher overall rebalancing rates, but with the majority of rebalancing occurring along less utilized routes.

In the few rare cases where maximum road utilization does increase, an intelligent routing strategy similar to Aslam et al. (2012) becomes crucial. While uniform routing along different paths helps distribute vehicles throughout the road network, a better routing strategy would actively route vehicles away from congested roads, limit rebalancing when it may cause further delays, and perhaps even stagger passenger trips to *reduce* congestion. This is related to the simultaneous

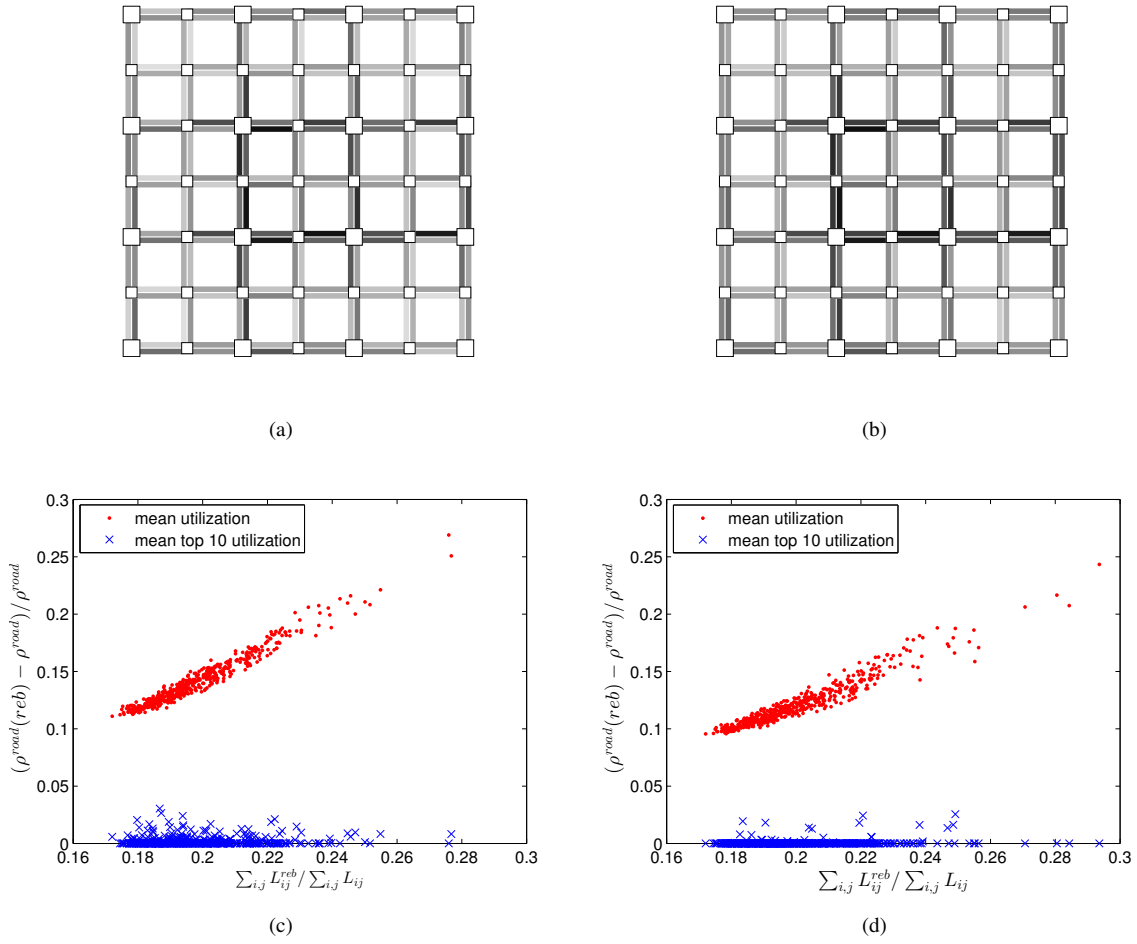


Fig. 11. 11(a): A randomly generated system without rebalancing on a 7×7 square grid road network with 16 stations, shown with large squares. Small squares mark the intersections. White represents low congestion and black represents heavy congestion. 11(b): The same system with rebalancing. 11(c): 500 randomly generated systems with different arrival rates and routing distributions on the 7×7 grid 16-station road network. The x-axis is the ratio of rebalancing vehicles to passenger vehicles on the road. The y-axis is the fractional increase in road utilization due to rebalancing. The dots show the mean increase in road utilization. The crosses show the mean increase in utilization of the top 10 most congested road segments. 11(d): The same 500 systems with adjusted rebalancing rates.

departure and routing problem in Huang & Lam (2002), a class of dynamic traffic assignment problems, and will be the subject of future work.

8. Conclusions

In this paper we presented and analyzed a queueing-theoretical model for autonomous MOD systems. We showed that an optimal open-loop policy can be readily found by solving a linear program. Based on this policy, we developed a closed-loop, real-time rebalancing policy that appears to be quite efficient, and we applied it to a case study of New York City. Finally, we showed that while vehicle rebalancing can potentially have a detrimental impact on traffic congestion in already-congested systems, in most cases, rebalancing vehicles tend to travel along less congested roads and by rerouting these vehicles, the detrimental impact can be effectively avoided.

This paper leaves numerous important extensions open for further research. First, it is of interest to develop rebalancing policies that can both route rebalancing vehicles along less congested roads and limit the number of rebalancing vehicles

when the system is overly congested. Second, we plan to study different performance metrics (e.g., minimization of waiting times) and include a richer set of constraints (e.g., time windows to pick up the customers). Third, the current real-time rebalancing problem is formulated based on a centralized architecture. While this may be reasonable in cities with a widespread availability of cellular networks, it may not be possible in cases where these networks are not sophisticated enough to relay the necessary information in real time. Therefore, we plan to study distributed formulations of the rebalancing problem to further widen the applicability of robotic MOD systems. Fourth, it is of interest to include in the model the provision for mass transit options (e.g., a metro) and develop optimal coordination algorithms for such an intermodal system. Fifth, we plan to extend the theoretical model to more realistic time-varying customer arrival rates and develop rebalancing algorithms by combining demand prediction and routing. Sixth, we plan to consider additional case studies (e.g., from Asia and Europe) and study in more detail the economic and societal benefits of robotic MOD systems. Finally, we plan to extend the small-scale hardware testbed to study congested systems and demonstrate the algorithms on real driverless vehicles providing MOD service in a gated community.

Acknowledgements

The authors would like to acknowledge Christopher Pepper and Brandon Jennings for their contributions to the hardware testbed, as well as Federico Rossi for his work on the multimedia extensions. This research was supported in part by the National Science Foundation under CAREER Award CMMI-1454737 and by the Dr. Cleve B. Moler Stanford Graduate Fellowship.

A. Index to Multimedia Extensions

Extension	Media Type	Description
1	Video	Simulation of case study in Manhattan
2	Video	AMOD Testbed

References

J. Aslam, S. Lim and D. Rus (2012). 'Congestion-aware Traffic Routing System using sensor data'. In *15th International IEEE Conference on Intelligent Transportation Systems*, pp. 1006–1013.

M. Barth, J. Han and M. Todd (2001). 'Performance evaluation of a multi-station shared vehicle system'. In *Proceedings of IEEE Intelligent Transportation Systems*, pp. 1218–1223.

G. Berbeglia, J. F. Cordeau and G. Laporte (2010). 'Dynamic pickup and delivery problems'. *European Journal of Operational Research* **202**(1):8–15.

D. P. Bertsekas, R. G. Gallager and P. Humblet (1992). *Data networks*, vol. 2. Prentice-Hall International.

F. Bullo, E. Frazzoli, M. Pavone, K. Savla and S. L. Smith (2011). 'Dynamic vehicle routing for robotic systems'. *Proceedings of the IEEE* **99**(9):1482–1504.

Bureau of Public Roads (1964). *Traffic Assignment Manual*. US Department of Commerce.

L. Burns, W. Jordan and B. Scarborough (2013). *Transforming personal mobility*. The Earth Institute – Columbia University.

T. Campbell, M. Liu, B. Kulis, J. P. How and L. Carin (2013). 'Dynamic clustering via asymptotics of the Dependent dirichlet process mixture'. In *Advances in Neural Information Processing Systems*, pp. 449–457.

CAR2GO (2011). 'CAR2GO Austin. Car Sharing 2.0: Great Idea for a Great City'. Tech. rep.

M. Dell'Amico, E. Hadjicostantinou, M. Iori and S. Novellani (2014). 'The bike sharing rebalancing problem: Mathematical formulations and benchmark instances'. *Omega* **45**(0):7–19.

L. Di Gaspero, A. Rendl and T. Urli (2013). 'Constraint-Based Approaches for Balancing Bike Sharing Systems'. In *Principles and Practice of Constraint Programming*, vol. 8124 of *Lecture Notes in Computer Science*, pp. 758–773. Springer Berlin Heidelberg.

A. Fisher (2013). *Inside Google's Quest To Popularize Self-Driving Cars*. Popular Science (Online Article).

C. Fricker and N. Gast (2012). 'Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity'. *EURO Journal on Transportation and Logistics* pp. 1–31.

D. K. George and C. H. Xia (2011). 'Fleet-sizing and service availability for a vehicle rental system via closed queueing networks'. *European Journal of Operational Research* **211**(1):198–207.

GM (2011). *EN-Vs Impress Media at Consumer Electronics Show*. Online Article.

H. Huang and W. H. K. Lam (2002). 'Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues'. *Transportation Research Part B: Methodological* **36**(3):253–273.

Induct (2013). *Navia - The 100% Electric Automated Transport*. Online Article.

B. N. Janson (1991). 'Dynamic traffic assignment for urban road networks'. *Transportation Research Part B: Methodological* **25**(2):143–161.

B. Kulis and M. I. Jordan (2012). 'Revisiting k-means: New Algorithms via Bayesian Nonparametrics'. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 513–520.

R. C. Larson and A. R. Odoni (1981). *Urban operations research*. Prentice-Hall.

S. S. Lavenberg (1983). *Computer performance modeling handbook*, vol. 4. Elsevier.

H. Lieu (2003). *Revised monograph on traffic flow theory*. US Department of Transportation Federal Highway Administration.

S. M. Madanat, M. J. Cassidy and M. Wang (1994). 'Probabilistic delay model at stop-controlled intersection'. *Journal of Transportation Engineering* **120**(1):21–36.

C. D. Meyer (2000). *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics.

W. J. Mitchell, C. E. Borroni-Bird and L. D. Burns (2010). *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*. The MIT Press, Cambridge, MA.

M. Pavone, E. Frazzoli and F. Bullo (2011). 'Adaptive and Distributed Algorithms for Vehicle Routing in a Stochastic and Dynamic Environment'. *IEEE Transactions on Automatic Control* **56**(6):1259–1274.

M. Pavone, S. L. Smith, E. Frazzoli and D. Rus (2012). 'Robotic load balancing for mobility-on-demand systems'. *The International Journal of Robotics Research* **31**(7):839–854.

M. Reiser and S. S. Lavenberg (1980). 'Mean-value analysis of closed multichain queuing networks'. *Journal of the ACM* **27**(2):313–322.

R. Serfozo (1999). *Introduction to stochastic networks*, vol. 44. Springer.

S. L. Smith, M. Pavone, E. Schwager, E. Frazzoli and D. Rus (2013). 'Rebalancing the Rebalancers: Optimally Routing Vehicles and Drivers in Mobility-on-Demand Systems'. In *American Control Conference*, pp. 2362–2367.

K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton and M. Pavone (2014). 'Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: a case study in Singapore'. In *Springer Lecture Notes in Mobility*. Springer.

K. Treleaven, M. Pavone and E. Frazzoli (2013). 'Asymptotically Optimal Algorithms for Pickup and Delivery Problems with Application to Large-Scale Transportation Systems'. *IEEE Transactions on Automatic Control* **58**(9):2261–2276.

UN (2011). 'World Urbanization Prospects: The 2011 Revision Population Database'. Tech. rep., United Nations.

A. Waserhole and V. Jost (2013). 'Pricing in Vehicle Sharing Systems: Optimization in queuing networks with product forms'. OSP. At hal.archives-ouvertes.fr <hal-00751744v4>.

R. Zhang and M. Pavone (2014). 'Control of Robotic Mobility-On-Demand Systems: a Queueing-Theoretical Perspective'. In *Robotics: Science and Systems Conference*.