Congestion-Aware Randomized Routing in Autonomous Mobility-on-Demand Systems

Federico Rossi, Ramon Iglesias, Rick Zhang, and Marco Pavone

Abstract— **In this paper we study the routing and rebalancing problem for a fleet of autonomous (i.e., self-driving) vehicles providing on-demand transportation within a capacitated urban road network. We show that finding a *congestion-free* solution to the routing and rebalancing problem is NP-hard. We thus provide a polynomial-time randomized algorithm which finds a *low-congestion* solution that approximately minimizes the travel time of passengers. The algorithm enjoys correctness guarantees in terms of (1) theoretical bounds on the probability of violating the congestion constraints, and (2) approximation factor with respect to the minimum expected passenger travel time that can be achieved with a congestion-free solution for road networks with symmetric edges. We evaluate the performance of the algorithm on a high-fidelity model of the Manhattan road network. We show that the proposed algorithm introduces a very small amount of congestion, while offering travel times that are very close to (and sometimes better than) what is achievable with a congestion-free solution.**

I. Introduction

Transportation networks in dense urban cities are faced with a number of challenges including traffic congestion, pollution, and a shortage of space for additional infrastructure (such as roads and parking structures). These challenges have spurred the development of several key enabling technologies including vehicle sharing, electric vehicles, and autonomous (i.e., self-driving) vehicles. These technologies converge as autonomous mobility-on-demand (AMoD) – a future transportation system whereby a fleet of autonomous vehicles services customers on demand within an urban environment. Aside from potential benefits in terms of safety and transportation costs, AMoD systems have an inherent advantage over traditional taxi or carsharing systems in terms of fleet management and routing. With proper system-level coordination, vehicles in an AMoD system can be routed cooperatively to minimize trip duration and traffic congestion, while providing high quality of service by anticipating future customer demand through *rebalancing* (i.e., redistribution of empty vehicles).

Solving such a large-scale routing and rebalancing problem is computationally challenging. In particular, if an AMoD fleet contributes to a significant fraction of overall traffic, congestion becomes an *endogenous* effect: that is, routing and rebalancing strategies have a significant effect on traffic congestion and, in turn, on travel times. The objective of this paper is to develop an algorithmic framework to efficiently solve large-scale customer routing and vehicle rebalancing problems in the presence of congestion and with guaranteed bounds on solution quality.

Literature review: The problem of vehicle rebalancing has been studied for both carsharing [1], [2], [3] and AMoD systems [4], [5] where the underlying network is a complete graph (point-to-point routing). These approaches (i) seek to only optimize rebalancing routes, and (ii) do not consider congestion effects caused by routing customers and rebalancing vehicles on a road network. The routing and rebalancing problem we study in this paper is similar to the one-to-one pickup and delivery problem on a Euclidean plane [6] or on road networks [7], for which polynomial-time, asymptotically-optimal algorithms exist. However, these formulations do not take into account congestion. In particular, for the case of AMoD, the presence of empty-traveling rebalancing vehicles on the road was believed to have a negative impact on congestion [8], [9]. Recent work, however, suggests that with optimized routing this need not be the case [10]. The work in [10], however, is limited to a *macroscopic* analysis of system behavior, and does not directly allow the computation of individual vehicle routes. In contrast, in this paper we focus on the computation of *individual routes* for the vehicles. In a nutshell, this is achieved by modifying the model in [10] to accommodate integral flows.

In the field of transportation science, our problem is similar to the dynamic traffic assignment (DTA) problem [11]. The two major differences between our problem and DTA approaches is that (i) DTA only optimizes customer routes, not rebalancing routes, and (ii) most DTA methods optimize for user equilibrium, where a decision by any vehicle to change routes would necessarily lead to an increase in travel time. A key advantage of AMoD systems is that vehicles can be centrally coordinated and optimally routed: accordingly, in this paper we seek a *system-optimal* solution, as opposed to a user equilibrium.

Finally, traffic congestion is a well-studied topic in transportation science. Existing congestion models vary in degree of fidelity: from basic models establishing the relationship among speed, density, and flow [12], to simulation-based microscopic car-following models [13]. However, for the most part, the purpose of traffic modeling has been the *analysis* of traffic patterns rather than the *active coordination* and *control* of traffic. In this paper, we leverage simple, yet efective traffic models that are amenable to tractable analysis and control.

Statement of contributions: Our goal is to efficiently find optimal customer-carrying and vehicle rebalancing routes that minimize the overall travel times (or, equivalently, the overall number of vehicles) in the presence of congestion

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

Ramon Iglesias is with the Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305 rdit@stanford.edu.

Rick Zhang is with Zoox Inc., 325 Sharon Park Dr. #909, Menlo Park, CA 94025 rick@zoox.com.

This research was supported by the National Science Foundation under CAREER Award CMMI-1454737.

effects. Our strategy is to design a polynomial-time approximation algorithm to the problem of *congestion-free* routing and rebalancing with minimum number of vehicles (whereby congestion follows a simple threshold model based on road capacities). This latter problem serves as a *proxy* for solving the general problem of minimizing travel times in the presence of congestion effects, and allows us to avoid the use of sophisticated congestion models for the computation of travel times (that would further compound the complexity of the routing problem).

Specifically, our contribution is threefold: First, we propose an approximate randomized algorithm for the solution to the congestion-free routing and rebalancing problem. The algorithm produces solutions that violate the capacity constraints by small amounts – a probabilistic characterization of the degree of constraint violation is analytically derived for symmetric road networks. Second, we provide a semi-analytical characterization of the algorithm's approximation factor for symmetric road networks. The approximation factor represents the ratio between the total customer travel time needed with randomized routing under an empirical congestion model, and the optimal *congestion-free* total customer travel time. Third, we validate the performance of the randomized routing algorithm on a realistic road network with real-world customer demand. The randomized technique provides a solution that compares very favorably with the optimal congestion-free one, and, in general, allows one to route thousands of vehicles with minimal impact on the transportation network.

Organization: The rest of the paper is organized as follows: in Section II we introduce notation, present a network flow model for AMoD, and rigorously define the integer congestion-free routing and rebalancing problem. A randomized approximation algorithm for such a problem is presented in Section III, together with correctness guarantees. Numerical results corroborating our analysis are provided in Section IV, while conclusions and future directions are summarized in Section V.

II. Model Description and Problem Formulation

We define the routing and rebalancing problem as a network flow problem on a capacitated road network. The model we adopt is largely similar to the one presented in [10]. The key difference is our assumption that customer demands, network flows, and road capacities are integral. This assumption is in line with our goal of solving for individual vehicle routes, which can then be used as part of a practical real-time control algorithm for large vehicle fleets.

A. Congestion model

Two congestion models are adopted in this paper. A simpler *synthesis* model is used to compute vehicle routes in Sections III-B and III-C, whereas a higher-fidelity *analysis* model is used to (a) analytically characterize the performance of the approximate routing algorithm presented in Section III-F.2 and (b) numerically evaluate algorithm's performance with real-world customer demand in Section IV.

Both models are consistent with classical traffic flow theory [14], [12]. In classical traffic flow theory, for a given road, vehicle speeds tend to remain relatively constant at low vehicle densities (called the "free flow" speed) [14]. The flow rate (i.e., the number of vehicles traversing a road per unit time) grows with vehicle density up to a critical value (referred to in the literature as the *capacity* of the road), at which point vehicle speeds and flow rate decrease significantly, signaling the onset of congestion. The capacity of the road is reached when the flow rate is maximized.

The synthesis congestion model adopted in this paper is a threshold model. The vehicle density on each road is constrained to be no larger than the critical road density, which corresponds to the road capacity. Every vehicle travels at the free flow speed. This model captures the behavior of traffic up to the onset of congestion: furthermore, any set of vehicle routes that respects the capacity constraints on every road is guaranteed to be *congestion-free*.

The analysis congestion model offers a characterization of the congested behavior of a road. We assume that the travel time $\tilde{t}$ is strictly increasing in the flow rate f traversing the link; we place no further assumptions on its shape. One such congestion model is the widely used Bureau of Public Roads (BPR) link delay model [15], which models the travel time on a link as

$$\tilde{t}(f) = t_0 \left(1 + 0.15\,(f/c)^4\right) \qquad (1)$$

where $\tilde{t}$ is the travel time associated with flow rate f, t_0 is the free-flow travel time and c is the capacity of the road.

B. Network flow model of AMoD system

We model a road network as a capacitated graph $G(\mathcal{V},\mathcal{E})$. The nodes $v \in \mathcal{V}$ represent intersections and locations for customer trip origins and destinations. The edges $(u,v) \in \mathcal{E}$ represent road links. Congestion is modeled as a constraint on the vehicle flow that a road link can accommodate, in accordance with the synthesis congestion model. Specifically, a function $c(u,v) : \mathcal{E} \mapsto N_{\geq 0}$ denotes the capacity of each link (in vehicles per unit time). All vehicles on a link travel at the free-flow speed. The corresponding free-flow travel time across the road link is denoted by $t(u,v) : \mathcal{E} \mapsto \mathbb{R}_{>0}$. Throughout our analysis, we assume that the road network is *symmetric*: $(u,v) \in \mathcal{E} \Leftrightarrow (v,u) \in \mathcal{E}$ and $c(u,v) = c(v,u)$ $\forall (u,v) \in \mathcal{E}$. We will relax this assumption in the numerical experiments presented in Section IV.

Customer requests are denoted by the tuple (s,t,λ), where $s \in \mathcal{V}$ is the origin of the request, $t \in \mathcal{V}$ is the destination of the request, and $\lambda \in \mathbb{N}_{>0}$ is the number of passengers wishing to travel from s to t in one unit of time, henceforth called the intensity of the request. Transportation requests are assumed to be stationary and deterministic, that is, λ is constant in time. The set of transportation requests is denoted as $\mathcal{M} = \{(s_m, t_m, \lambda_m)\}_m$ with $m \in \{1,\ldots,|\mathcal{M}|\}$. In this paper, we restrict our analysis to the case where each customer request has unit intensity. This is without loss of generality: a customer request of intensity λ can be transformed as λ customer requests of unit intensity between the same origin and destination nodes. However, in the interest of making our notation clearer, we will use λ_m whenever we refer to the intensity of a customer request.

A customer route is an ordered list of edges $\{(s_m,u),(u,v),\ldots,(w,t_m)\}$ that forms a path connecting a customer origin s_m with a customer destination t_m. Each customer route is associated with a number λ of customers traveling on the route. Analogously, a rebalancing route is a path connecting a customer destination t_m with a customer origin s_l (for rebalancing paths, the origin and destination may belong to different customers), associated with a number of vehicles traveling on that route. Vehicles follow customer routes to transport customers from their respective origins to their destinations; rebalancing routes realign the vehicle distribution with the distribution of passenger departures by moving empty vehicles to passengers' origins.

We model customer routes and rebalancing routes as *flows* of customers and vehicles on the graph $G(\mathcal{V}, \mathcal{E})$. The concept of network flow is central to our description. For a given origin s, destination t, and intensity λ, a network flow is a function $f(u,v) : \mathcal{E} \mapsto R_{\geq 0}$ that obeys the following equation:

$$\sum_{u \in \mathcal{V}} f(u,v) + 1_{v=s}\lambda = \sum_{w \in \mathcal{V}} f(v,w) + 1_{v=t}\lambda, \ \forall v \in \mathcal{V}, \tag{2}$$

where 1_x denotes the indicator function of the Boolean variable $x = \{\text{true, false}\}$ (1_x equals one if x is true). An *integral* network flow is a network flow that is restricted to take on integer values.

The definition of network flows can be generalized to flows with multiple sources and sinks. Consider a collection of origins $\{s_i\}_i$ with intensities $\{\lambda_i\}_i$, and a collection of destinations $\{t_j\}_j$ with intensities $\{\lambda_j\}_j$. Then a network flow for those origins, destinations, and intensities is a function $f(u,v) : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$ that satisfies

$$\sum_{u \in \mathcal{V}} f(u,v) + \sum_{i} 1_{v=s_i}\lambda_i = \sum_{w \in \mathcal{V}} f(v,w) + \sum_{j} 1_{v=t_j}\lambda_j, \ \forall v \in \mathcal{V}. \tag{3}$$

Note that Equation 3 can be satisfied at all nodes only if $\sum_i \lambda_i = \sum_j \lambda_j$, that is if the sum of the intensities of all origins equals the sum of intensities of all destinations.

For a given origin s, destination t, and intensity λ, we define a *path flow* as a function $f^p(u,v) : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$ that satisfies Equation 2 and only assigns positive flow to edges belonging to a path p going from s to t; in other words, there exists $p = \{(s,u),(u,v),\ldots,(w,t)\}$ such that $f^p(u,v) > 0 \Leftrightarrow (u,v) \in p$. A customer or rebalancing route can be equivalently described as a path flow of intensity $\lambda = 1$ by assigning value $f(u,v) = 1$ to the edges contained in the route and $f(u,v) = 0$ otherwise. Hence, path flows *equivalently* represent customer and rebalancing routes, a fact we will extensively leverage in Section III-B.

C. Integral Congestion-free Routing and Rebalancing

The integral Congestion-free Routing and Rebalancing problem (i-CRRP) is defined as follows,

Definition 2.1 (i-CRRP): Given a network flow model of an AMoD system, compute a set of routes that

(i) transfers customers to their desired destinations (customer-carrying trips),
(ii) rebalances vehicles throughout the network to realign the vehicle fleet with the customers' demands (customer-empty, or rebalancing trips),
(iii) does not cause congestion on any road link, and
(iv) minimizes the overall travel time of customer-carrying vehicles.

The i-CRRP can be cast as a mixed-integer linear program. We represent passenger routes for passengers of class $m \in \mathcal{M}$ with the integral network flow $\{f_m(u,v)\}_{(u,v)}$. Similarly, we represent rebalancing routes with the integral network flow $\{f_{\text{R}}(u,v)\}_{(u,v)}$. Given a capacitated network $G(\mathcal{V}, \mathcal{E})$ and a set of transportation requests $\mathcal{M} = \{(s_m, t_m, \lambda_m)\}$, the i-CRRP entails solving

$$\underset{f_m(\cdot,\cdot), f_{\text{R}}(\cdot,\cdot)}{\text{minimize}} \sum_{m \in \mathcal{M}} \sum_{(u,v) \in \mathcal{E}} t(u,v) f_m(u,v) \tag{4a}$$

$$\text{subject to} \sum_{u \in \mathcal{V}} f_m(u,v) + 1_{v=s_m}\lambda_m$$

$$= \sum_{w \in \mathcal{V}} f_m(v,w) + 1_{v=t_m}\lambda_m \ \ \forall m \in \mathcal{M}, v \in \mathcal{V} \tag{4b}$$

$$\sum_{u \in \mathcal{V}} f_{\text{R}}(u,v) + \sum_{m \in \mathcal{M}} 1_{v=t_m}\lambda_m$$

$$= \sum_{w \in \mathcal{V}} f_{\text{R}}(v,w) + \sum_{m \in \mathcal{M}} 1_{v=s_m}\lambda_m \quad \forall v \in \mathcal{V} \tag{4c}$$

$$f_{\text{R}}(u,v) + \sum_{m \in \mathcal{M}} f_m(u,v) \leq c(u,v) \ \ \forall (u,v) \in \mathcal{E} \tag{4d}$$

$$f_m(u,v) \in \mathbb{N}_{\geq 0} \ \ \forall m \in \mathcal{M}, (u,v) \in \mathcal{E} \tag{4e}$$

$$f_{\text{R}}(u,v) \in \mathbb{N}_{\geq 0} \ \ \forall (u,v) \in \mathcal{E}. \tag{4f}$$

Equation 4a captures the goal of minimizing the number of vehicles on the road. Equations 4b and 4e ensure that each customer-carrying flow $\{f_m(u,v)\}_{(u,v),m}$ is an integral network flow. Equations 4c and 4f ensure that the rebalancing flow $\{f_{\text{R}}(u,v)\}_{(u,v)}$ is an integral network flow. Finally, Equation 4d enforces the capacity constraint for every link. One can easily recognize that the i-CRRP is an instance of an integral minimum-cost multicommodity flow problem (Min-MCF) where customers and rebalancing vehicles are interpreted as commodities. We leverage this observation to characterize its complexity.

Theorem 2.2 (Complexity of the i-CRRP): The decision version of the i-CRRP is NP-complete.

Proof sketch: SAT can be reduced to the i-CRRP: the reduction of SAT to the multi-commodity flow problem in [16] can be adapted to the case of multi-commodity flows with a rebalancing commodity (i.e., the i-CRRP). Therefore, the i-CRRP is NP-hard. A rigorous proof is reported in [**?**]. Any candidate solution to the i-CRRP can be verified in polynomial time: this proves that the i-CRRP is NP-complete.

We re-emphasize that the i-CRRP serves as a proxy for solving the general problem of minimizing travel times in the presence of congestion effects (without the complexity of dealing with sophisticated congestion models).

III. A Randomized Routing algorithm

Theorem 2.2 shows that the i-CRRP can not be solved efficiently for large problem instances unless P=NP. Furthermore, to the best of our knowledge, no polynomial-time approximation schemes are known for the integral Min-MCF problem (of which, as mentioned before, i-CRRP is an instance). Our strategy is then to focus on an approximate version of the i-CRRP, whereby capacity constraints can be slightly violated, that is,

Definition 3.1 (Approximate i-CRRP): Given a network model of an AMoD system, compute a set of routes that

(i) transfers customers to their desired destinations (customer-carrying trips),

(ii) rebalances vehicles throughout the network to re-align the vehicle fleet with the customers' demands (customer-empty, or rebalancing, trips),

(iii) with high probability, violates the capacity constraints on all road links by at most a small value δ,

(iv) has bounded suboptimality (in terms of customer travel times) under the analysis congestion model.

There exist randomized techniques for finding approximately optimal integral solutions for multicommodity flow problems, e.g., randomized routing [17], [18]. However, these require that each commodity has a single source and a single sink. Thus, they cannot be directly applied for the i-CRRP, since the rebalancing flow has multiple sources and sinks.

In [10], the authors show that the problems of routing passenger vehicles and routing rebalancing vehicles in the i-CRRP can be *decoupled* on *capacity*-symmetric networks (symmetric networks considered in this paper fall under the class of capacity-symmetric networks). Specifically, for any given set of feasible customer routes, there exists a feasible set of rebalancing routes. Following this intuition, we exploit randomized techniques to find low-congestion routes for the customer-carrying vehicles. The customer routes may violate some of the congestion constraints: thus, we modify the road network so as to guarantee that a feasible rebalancing solution exists.

The procedure can be summarized as follows. First, a solution to the linear programing (LP) relaxation of the i-CRRP is computed, which produces a set of customers' network flows and a rebalancing network flow (Section III-A). Second, each customer's network flow is "decomposed" into a collection of path flows (Section III-B), which fulfill the capacity constraint (4d) but may violate the other constraints, namely (4b), (4c), (4e), and (4f). Third, a sampling procedure adjusts the path flow intensities to yield customers' path flows that satisfy Equations (4b), (4c), (4e), and (4f) but may (slightly) violate some of the capacity constraints (Section III-C) – such path flows have intensity equal to 1 and are equivalent to customer routes (due to the equivalency between path flows of unit intensity and routes). Fourth, the residual capacity of edges in the road network are adjusted so as to guarantee that a feasible rebalancing flow exists; a rebalancing flow is then found by solving a linear program (Section III-D). Finally, the rebalancing flow is decomposed into a collection of rebalancing flow with unit intensity, which are an equivalent representation for rebalancing routes (Section III-E). In Section (III-F.2) we characterize the probability of violating the capacity constraints and the expected travel time of all customers under an empirical congestion model.

A. Step One: Linear Relaxation of the i-CRRP

The first step entails solving an LP relaxation of the i-CRRP, denoted as the *fractional* CRRP, whereby Equations 4e and 4f are replaced by

$$f_m(u,v) \in \mathbb{R}_{\geq 0} \qquad \forall m \in \mathcal{M}, (u,v) \in \mathcal{E}, \tag{5a}$$

$$f_{\mathrm{R}}(u,v) \in \mathbb{R}_{\geq 0} \qquad \forall (u,v) \in \mathcal{E}. \tag{5b}$$

Additionally, in order to reduce the size of the linear program, we *bundle* customer requests departing from the same origin into a single network flow. Specifically, we replace the set of customer requests leaving from a node s, $\{\{s_m, t_m, \lambda_m\}_m : s_m = s\}$, with a single customer request with origin s (with intensity $\sum_{m:s_m=s} \lambda_m$) and destinations $\{t_m\}_{m:s_m=s}$, each with intensity λ_m.

The resulting fractional CRRP is an instance of the fractional Min-MCF problem, which can be efficiently solved as a linear program of size $|\mathcal{E}|(S+1)$ (where S is the number of distinct origin nodes appearing in the customer requests) or via specialized combinatorial algorithms, e.g., [19].

B. Step Two: Flow Decomposition

The second step decomposes each customer network flow resulting from the relaxed LP into a collection of path flows, which can be later sampled and "adjusted" (in terms of their intensities) to ensure that constraints (4b) and (4e) are satisfied. Specifically, we utilize the decomposition algorithm in [20, Sec. 3.5] . This algorithm decomposes general network flows into a collection of path flows and cycles with complexity $O(|V||\mathcal{E}|)$ [20] – its output is a collection of path flows whose sum equals the input network flow. Each path flow has a single origin and a single destination, corresponding to the origin and one of the destinations in the corresponding network flow. Thus, each path flow corresponds to one of the original customer requests. Importantly, in our case this decomposition step does *not* yield any cycles: if a cycle was present, then removing it would result in a new solution with lower cost.

C. Step Three: Path Sampling

Step two yields, for every customer $m \in \mathcal{M}$, a set $\mathcal{F}_m$ of path flows (the cardinality of each set $\mathcal{F}_m$ is at most $|\mathcal{E}|$). Step three entails sampling path flows from each set $\mathcal{F}_m$ to obtain a set of customer path flows that satisfy constraints (4b), (4c), (4e), and (4f). Specifically, we randomly and independently sample one path flow from each set $\mathcal{F}_m$, by using path flow intensities as probability distribution (the intensities represent a valid probability distribution since for each customer m, $\lambda_m = 1$ by assumption, and hence the path flow intensities must sum to one). We then set the intensity of the sampled path flow, denoted as $\{f^{\mathrm{s}}_m(u,v)\}_{(u,v)}$, equal to one (i.e., $\text{intensity}(\{f^{\mathrm{s}}_m\}) = 1$). By construction, the set of path flows $\{f^{\mathrm{s}}_m\}_m$ satisfy Equations (4b) and (4e). Explicitly, for each customer m, the path flow $\{f^{\mathrm{s}}_m\}$ is a network flow with origin s_m, destination t_m, and intensity 1: thus, it satisfies Equation (4b). Path flows of intensity 1 have integral flow on every edge: thus, both $\{f^{\mathrm{s}}_m\}_m$ and $\{f^{\mathrm{s}}_{\mathrm{R}}\}$ satisfy Equation (4e).

In other words, the third step re-adjusts the intensities of the sampled path flows to ensure satisfaction of continuity constraints (4b) and integrality constraints (4e), at the expense of the capacity constraint (4d). Specifically, the expected (with respect to the randomization procedure) customer network flow crossing every edge $(u,v) \in \mathcal{E}$ is upper-bounded by the capacity of that edge, $c(u,v)$, that is:

$$\begin{aligned}\mathbb{E}\left[\sum_m f^{\mathrm{s}}_m(u,v)\right] &= \sum_m f_m(u,v) \\ &\leq \sum_m f_m(u,v) + f_{\mathrm{R}}(u,v) \leq c(u,v),\end{aligned}$$

where $\{f_m\}_m$ and $\{f_{\mathrm{R}}\}$ are the solutions from the LP relaxation in step one – the equality follows from the fact that path flows and the rebalancing solution are sampled with probability equal to their intensity. However, while the output of the algorithm satisfies the capacity constraints *in expectation*, a given realization may violate them.

D. Step Four: Computing a Rebalancing Flow

For a given collection of customer path flows, we adjust the residual capacity of the road network to ensure feasibility of the rebalancing flow. We define the *augmented capacity* of an edge as

$$\overline{c}(u,v) = \max\left(c(u,v),\, \sum_m f_m^s(u,v),\, \sum_m f_m^s(v,u)\right).$$

We then compute the *residual augmented capacity* of an edge as

$$\overline{c}_{\mathrm{R}}(u,v) = \overline{c}(u,v) - \sum_m f_m^s(u,v).$$

By construction, the sampled rebalancing flows are feasible solutions to the i-CRRP (and, in particular, they satisfy constraint (4d)) for a capacitated road network $\overline{G}(\mathcal{V},\mathcal{E})$ with capacities $\{\overline{c}(u,v)\}_{(u,v)}$. Furthermore, $\overline{c}(u,v) = \overline{c}(v,u)$, so such road network is symmetric. Therefore, in accordance with [10, Theorem 3.5], there exists a feasible rebalancing flow $\{f_{\mathrm{R}}^s(u,v)\}_{(u,v)}$ for the road network $\overline{G}(\mathcal{V},\mathcal{E})$ with capacities $\{\overline{c}(u,v)\}_{(u,v)}$ and customer flows $\{f_m^s(u,v)\}_{m,(u,v)}$ (we use the superscript s to denote the dependency of $\{f_{\mathrm{R}}^s(u,v)\}_{(u,v)}$ on the sampled customer path flows $\{\{f_m^s(u,v)\}_{(u,v)}\}$).

Such a rebalancing flow can be found by solving

$$\underset{f_{\mathrm{R}}^s(\cdot,\cdot)}{\text{minimize}} \quad \sum_{(u,v)\in\mathcal{E}} t(u,v) f_{\mathrm{R}}^s(u,v) \tag{6a}$$

$$\text{subject to} \quad \sum_{u\in\mathcal{V}} f_{\mathrm{R}}^s(u,v) + \sum_{m\in\mathcal{M}} 1_{v=t_m}\lambda_m$$

$$= \sum_{w\in\mathcal{V}} f_{\mathrm{R}}^s(v,w) + \sum_{m\in\mathcal{M}} 1_{v=s_m}\lambda_m \quad \forall v\in\mathcal{V} \tag{6b}$$

$$f_{\mathrm{R}}^s(u,v) \le \overline{c}_{\mathrm{R}}(u,v) \quad \forall (u,v)\in\mathcal{E} \tag{6c}$$

$$f_{\mathrm{R}}^s(u,v) \in \mathbb{N}_{\ge 0} \quad \forall (u,v)\in\mathcal{E}. \tag{6d}$$

Problem (6) is an instance of a single-commodity flow problem and all the source flows, sink flows and edge capacities are integral – as a result, the problem enjoys a totally unimodular structure and can be exactly and efficiently solved as a linear program by replacing constraint (6d) with

$$f_{\mathrm{R}}^s(u,v) \in \mathbb{R}_{\ge 0}, \quad \forall (u,v)\in\mathcal{E}. \tag{7}$$

E. Step Five: Flow Decomposition of the Rebalancing Network Flow

In the fifth step, we decompose the rebalancing network flow into a collection of path flows of unit intensity. In analogy with the customer path flow decomposition in Section III-B, we decompose the rebalancing network flow using the flow decomposition algorithm. The output of the flow decomposition algorithm is a collection of path flows connecting every rebalancing origin (i.e., every customer destination) with a rebalancing destination (i.e., customer origin). Since the rebalancing network flow is integral, the flow decomposition algorithm returns *integral* path flows, i.e., path flows with integral intensity. As before, the decomposition of the rebalancing network flow does not yield any cycles.

F. Randomized Routing: Complexity and Performance

In this section, we characterize the complexity and performance of the randomized routing algorithm in terms of probabability of capacity constraint violations and approximation factor. Algorithm 1 summarizes the procedure detailed in the previous subsections for finding a solution to the approximate i-CRRP (function FlowDecomposition represents the flow decomposition algorithm [20, Sec. 3.5]).

Algorithm 1 Randomized routing algorithm

```
1: Input: An instance of the i-CRRP
2: Output: customer and rebalancing path flows, $\{f_m^{\mathrm{s}}\}$ and $\{f_{\mathrm{RS}}^{\mathrm{s}}\}$
3: $\{f_m\}_m, \{f_{\mathrm{R}}\} \leftarrow$ solve LP relaxation of the i-CRRP
4: for all $m \in \mathcal{M}$ do
5:     $\{\mathcal{F}_m\} \leftarrow$ FlowDecomposition($\{f_m\}$)
6: $\{f_m^s\} \leftarrow$SampleCustomerPaths($\{\mathcal{F}_m\}_m$)
7: $\overline{c}_{\mathrm{R}} =$ ResidualAugmentedCapacity($\{f_m^s\}$)
8: $\{f_{\mathrm{R}}^s\} \leftarrow$ solve the LP relaxation of Problem 6
9: $\{f_{\mathrm{RS}}^s\} \leftarrow$FlowDecomposition($\{f_{\mathrm{R}}^s\}$)
```

1) Complexity: Computational complexity is given by,

Theorem 3.2 (Complexity of randomized routing): The computational complexity of Algorithm 1 is polynomial in the size of the i-CRRP.

Proof: Any linear program can be solved in time polynomial in the size of its inputs [21]. In particular, the LP relaxation of the i-CRRP in line 3 and the linear relaxation of Problem 6 in line 8 can be solved in polynomial time [19]. The computational complexity of the flow decomposition algorithm is $O(|V||\mathcal{E}|)$ [20]: the decomposition algorithm is called S times in lines 4-5 and once in line 9. The sampling procedure on line 6 is carried out $|\mathcal{M}|$ times and only involves trivial network flow manipulations. Finally, the residual augmented capacity of the network is computed with $|M||E|$ operations. Hence, the overall complexity is polynomial in the size of the i-CRRP. ■

2) Performance: In this section we characterize the performance of the randomized routing algorithm (Algorithm 1). We first study the probability that a capacity constraint is violated (recall that Algorithm 1 ensures satisfaction of the capacity constraints only in *expectation*).

Theorem 3.3 (Violation of capacity constraints): Assume $\max_{(u,v)\in\mathcal{E}} \sqrt{3\,\log(|\mathcal{E}|)\,/c(u,v)} \le 1$ and let $\underline{\alpha}$ be the unique $\alpha\in(0,\,1]$ such that

$$\max_{(u,v)\in\mathcal{E}} \sqrt{3\,\log(|\mathcal{E}|/\alpha)\,/c(u,v)} = 1.$$

Then, for any $\alpha\in[\underline{\alpha},\,1]$, with probability $1-\alpha$, Algorithm 1 finds a solution to the approximate i-CRRP such that the capacity constraint for each link (u,v) is violated at most by a multiplicative factor $(1+\delta_{u,v})$, where

$$\delta_{u,v} := \sqrt{3\,\log(|\mathcal{E}|/\alpha)\,/c(u,v)}, \quad \text{for all } (u,v).$$

Proof sketch: The proof is similar to that of Theorem 3.2 in [17]. It uses a Chernoff bound to characterize the probability that each capacity constraint is violated by the sampled customer path flows, and then Boole's inequality to bound the probability that no capacity constraint is violated by more than $(1+\delta_{u,v})$. By construction, if the sampled customer path flows do not violate the capacity constraints by more than a multiplicative factor $(1+\delta_{u,v})$ on any road link, the rebalancing network flow satisfies the same property: this establishes the claimed result. A rigorous proof is reported in [**?**].

A few comments are in order. Clarify, Theorem 3.3 relies on the assumption that $\max_{(u,v)\in\mathcal{E}} \sqrt{3\,\log(|\mathcal{E}|)\,/c(u,v)} \leq 1$. This assumption is quite mild: for typical routing maps, $|\mathcal{E}| \sim 10{,}000$ and $c(u,v) \geq 100$, which leads to $\underline{\alpha} \sim 10^{-10}$. Second, when Theorem 3.3's assumption is met, the violation of a capacity constraint is at most 100% (since $\delta_{u,v} \leq 1$, always) with probability $1-\underline{\alpha}$. Third, Theorem 3.3 relies on a Boole's inequality argument, which may lead to significant conservatism. A numerical characterization of the capacity constraint violations is performed in Section IV.

We now turn our attention to the approximation factor of the randomized routing algorithm. The Chernoff bound guarantees that any sampled solution yields a sampled customer traffic flow (and therefore a level of congestion) very close to the average with high probability. The rebalancing flow may add congestion: however, the overall vehicle flow on a link (u,v) is upper-bounded by the maximum between the capacity of the link, the customer traffic flow on link (u,v), and the customer traffic flow on link (v,u). As a result, the distribution of the customer travel time remains close to the distribution of the travel time experienced by customers on a road with infinite capacity.

To formalize this intuition, we first provide two lemmas characterizing the distribution of the overall travel time and the per-vehicle travel time of customer-carrying vehicles on a link. We then provide a lemma characterizing the effect of the rebalancing flow on the overall travel time of customer-carrying vehicles. Finally, in Theorem 3.7 we provide a bound on the ratio of expected customer travel time between Algorithm 1 and the fractional CRRP solution.

Let $\{f^{\mathrm{s}}_{\mathrm{C}}(u,v)\}_{(u,v)}$ be the customer network flow induced by the output of Algorithm 1:

$$\{f^{\mathrm{s}}_{\mathrm{C}}(u,v)\}_{(u,v)} = \left\{\sum_m f^{\mathrm{s}}_m(u,v)\right\}_{(u,v)}. \tag{8}$$

We denote the expectation of $f^{\mathrm{s}}_{\mathrm{C}}(u,v)$ as $f_{\mathrm{C}}(u,v) := \mathbb{E}\left[f^{\mathrm{s}}_{\mathrm{C}}(u,v)\right]$. Note that $f_{\mathrm{C}}(u,v)$, is equal to the total flow $\sum_m f_m(u,v)$ of customer-carrying vehicles on link (u,v) induced by the solution to the LP relaxation of the i-CRRP (as discussed in Section III-C).

Lemma 3.4 (Per-link approximation factor, no rebalancing): Consider a link $(u,v) \in \mathcal{E}$. Let $\tilde{t}_{(u,v)}(\cdot) \in \mathcal{T}$ be an analysis congestion model for link (u,v) and assume $\mathbb{E}\left[f^{\mathrm{s}}_{\mathrm{C}}(u,v)\right] > 0$. Then, there exists a function $U : \mathcal{T} \times \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$, which can be numerically evaluated, such that

$$\frac{\mathbb{E}\left[\tilde{t}_{(u,v)}(f^{\mathrm{s}}_{\mathrm{C}}(u,v)) \cdot f^{\mathrm{s}}_{\mathrm{C}}(u,v)\right]}{\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v)) \cdot f_{\mathrm{C}}(u,v)} \leq U\Big(\tilde{t}_{(u,v)}(\cdot),\, f_{\mathrm{C}}(u,v)\Big).$$

Proof sketch: Define the random variable

$$R := \frac{\tilde{t}_{(u,v)}(f^{\mathrm{s}}_{\mathrm{C}}(u,v)) \cdot f^{\mathrm{s}}_{\mathrm{C}}(u,v)}{\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v)) \cdot f_{\mathrm{C}}(u,v)}.$$

The goal is to compute an upper bound on $\mathbb{E}[R]$. The first step entails computing a Chernoff bound for the random variable $f^{\mathrm{s}}_{\mathrm{C}}(u,v)$. We then define the bijective function $y \mapsto q(y)$:

$$q(y) = \frac{\tilde{t}_{(u,v)}(y) \cdot y}{\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v)) \cdot f_{\mathrm{C}}(u,v)},$$

whose inverse is denoted as $q^{-1}(\cdot)$. Since $q(\cdot)$ is bijective, the Chernoff bound for $f^{\mathrm{s}}_{\mathrm{C}}(u,v)$ can be transformed into a lower bound for the cumulative distribution function of R. Define the function $x \mapsto g(x)$:

$$g(x) = (x/f_{\mathrm{C}}(u,v) - 1),$$

and the auxiliary random variable H with complementary cumulative distribution function:

$$\mathbb{P}\left(H \leq h\right) = 1 - \left(\frac{e^{g(q^{-1}(h))}}{(1+g(q^{-1}(h)))^{(1+g(q^{-1}(h)))}}\right)^{f_{\mathrm{C}}(u,v)}.$$

Note that $\mathbb{E}[H]$ depends only on the analysis congestion model $\tilde{t}_{(u,v)}(\cdot)$ and the expected total flow. The proof is completed by showing that $\mathbb{E}(H)$ is indeed an upper bound on $\mathbb{E}[R]$ (thus, $U\Big(\tilde{t}_{(u,v)}(\cdot),\, f_{\mathrm{C}}(u,v)\Big) = \mathbb{E}[H]$). A rigorous proof is reported in [**?**].

A very similar procedure can be used to provide an upper bound on the travel time on a link.

Lemma 3.5 (Per-link travel time, no rebalancing): Consider a link $(u,v) \in \mathcal{E}$. Let $\tilde{t}_{(u,v)}(\cdot) \in \mathcal{T}$ be an analysis congestion model for link (u,v) and assume $\mathbb{E}\left[f^{\mathrm{s}}_{\mathrm{C}}(u,v)\right] > 0$. Then, there exists a function $V : \mathcal{T} \times \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$, which can be numerically evaluated, such that

$$\frac{\mathbb{E}\left[\tilde{t}_{(u,v)}(f^{\mathrm{s}}_{\mathrm{C}}(u,v))\right]}{\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v))} \leq V\Big(\tilde{t}_{(u,v)}(\cdot),\, f_{\mathrm{C}}(u,v)\Big).$$

Proof sketch: The proof of Lemma 3.5 is identical to the proof of Lemma 3.4.

We are now in a position to characterize the effect of the rebalancing flow on the overall travel time of customer-carrying vehicles.

Lemma 3.6 (Per-link approximation factor): Consider a link $(u,v) \in \mathcal{E}$. Let $\tilde{t}_{(u,v)}(\cdot) \in \mathcal{T}$ be an analysis congestion model for link (u,v) and assume $\mathbb{E}\left[f^{\mathrm{s}}_{\mathrm{C}}(u,v)\right] > 0$. Then, there exists a function $B : \mathcal{T} \times \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$, which can be numerically evaluated, such that

$$\frac{\mathbb{E}\left[\tilde{t}_{(u,v)}(f^{\mathrm{s}}_{\mathrm{C}}(u,v) + f^{s}_{\mathrm{R}}(u,v)) \cdot f^{\mathrm{s}}_{\mathrm{C}}(u,v)\right]}{\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v)) \cdot f_{\mathrm{C}}(u,v)} \leq B\Big(\tilde{t}_{(u,v)}(\cdot),\, f_{\mathrm{C}}(u,v)\Big).$$

Proof sketch: Three cases are possible. If both $f^{s}_{\mathrm{C}}(u,v)$ and $f^{s}_{\mathrm{C}}(v,u)$ satisfy the congestion threshold, then the travel time may increase by up to $\Delta t_{\mathrm{R},(u,v)} := \tilde{t}_{(u,v)}(c(u,v))/\tilde{t}_{(u,v)}(0)$. If $f^{s}_{\mathrm{C}}(u,v) > c(u,v)$ (that is, $f^{s}_{\mathrm{C}}(u,v)$ does not satisfy the congestion constraint) and $f^{s}_{\mathrm{C}}(u,v) \geq f^{s}_{\mathrm{C}}(v,u)$, then the augmented residual capacity $\overline{c}(u,v)$ is zero: the rebalancing flow on the link is also zero and the travel time is unchanged. Finally, if $f^{s}_{\mathrm{C}}(u,v) < f^{s}_{\mathrm{C}}(v,u), f^{s}_{\mathrm{C}}(v,u) \geq c(v,u)$, the rebalancing flow may increase the travel time on link (u,v) up to $\tilde{t}(f^{s}_{\mathrm{C}}(v,u))$. One can show that $f^{s}_{\mathrm{C}}(u,v)$ and $f^{s}_{\mathrm{C}}(v,u)$ are independent: as a result, the travel time on link (u,v) admits an upper bound of $V(\tilde{t}_{(v,u)}(\cdot), f_{\mathrm{C}}(v,u))\,\Delta t_{\mathrm{R},(u,v)}$. In conclusion,

$$B(\tilde{t}_{(u,v)}, f_{\mathrm{C}}(u,v)) := \max\big(\Delta t_{\mathrm{R},(u,v)} U(\tilde{t}_{(u,v)}, f_{\mathrm{C}}(u,v)),\; V(\tilde{t}_{(v,u)}(\cdot), f_{\mathrm{C}}(v,u))\Delta t_{\mathrm{R},(u,v)}\big).$$

A rigorous proof is reported in [**?**].

Lemma 3.6 shows that the ratio between the expected overall travel time of customer-carrying vehicles on link (u,v)

under Algorithm 1 (i.e., $\mathbb{E}\left[\tilde{t}_{(u,v)}(f^{s}_{\mathrm{C}}(u,v))\cdot f^{s}_{\mathrm{C}}(u,v)\right]$) and the overall travel time on link (u,v) under the LP relaxation of the i-CRRP (i.e., $\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v))\cdot f_{\mathrm{C}}(u,v)$) is upper bounded by a function that depends only on the analysis congestion model $\tilde{t}_{(u,v)}(\cdot)$ and the expected total flow. Such a function can be numerically evaluated by computing $\mathbb{E}[H]$ (as defined in the proof sketch of Lemma 3.4), given $\tilde{t}_{(u,v)}(\cdot)$ and $f_{\mathrm{C}}(u,v)$. As an example, consider the Bureau of Public Roads (BPR) delay model presented in Section II-A. Figure 1 shows the bound B as a function of the capacity parameter (i.e., c in Equation (1)) and the total traffic flow. The bound is quite tight, especially when the total traffic flow is close to link capacity – thus the routes computed with Algorithm 1 appear to induce travel times on the road relatively close to the optimal congestion-free number.

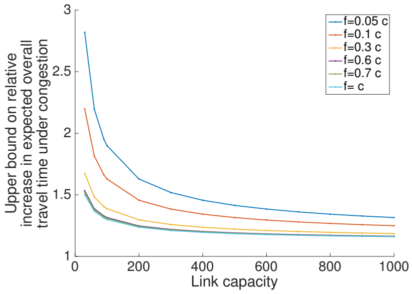

Fig. 1. Upper bound B on the fractional increase in expected value of the overall travel time of customer-carrying vehicles on a link as a function of link flow and link capacity. The BPR link delay model is used.

The per-link bound in Lemma 3.6 can be easily extended to a system-wide bound as follows.

Theorem 3.7 (System-wide approximation factor): Let $\tilde{t}_{(u,v)}(\cdot)\in\mathcal{T}$ be an analysis congestion model for each link $(u,v)\in\mathcal{E}$. Then, the ratio between the expected number of vehicles on the road under Algorithm 1 and the number of vehicles on the road under the LP relaxation of the i-CRRP is upper bounded by:

$$\frac{\sum_{(u,v)\in\mathcal{E}} B\Big(\tilde{t}_{(u,v)}(\cdot),\, f_{\mathrm{C}}(u,v)\Big)\,\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v))\cdot f_{\mathrm{C}}(u,v)}{\sum_{(u,v)\in\mathcal{E}}\tilde{t}_{(u,v)}(f_{\mathrm{C}}(u,v))\cdot f_{\mathrm{C}}(u,v)}. \tag{9}$$

Proof: The proof of this statement follows trivially from the linearity of expectation and Lemma 3.6. ■

IV. Numerical Experiments

A. Performance of the randomized routing and rebalancing algorithm

We explore the performance of the randomized routing procedure described in Algorithm 1 on a real-world, asymmetric road network with real customer demands. We consider a road network of Manhattan with 3137 roads and 1351 intersections, derived from OpenStreetMap data. Customer requests are derived from 55412 actual taxi rides in Manhattan on March 1, 2012 from 6 to 8 p.m [1].

We adjust the capacities of the roads such that, on average, the flows induced by these trips are close to the onset of congestion. In order to guarantee feasibility of the LP relaxation of the i-CRRP, we relax the congestion constraints by introducing slack variables, each associated with a large cost. Since the road network is not symmetric (and, in particular, some roads may be one-way streets), it is not possible to compute augmented road capacities. To circumvent this, we associate slack variables to the congestion constraints in the LP relaxation of Problem 6: the cost associated with each slack variable is proportional to the effect that an increase in congestion would have on the overall customer travel time. Intuitively, the algorithm is allowed to select the *minimal* relaxation of the congestion constraints that guarantees feasibility of the rebalancing problem. We solve the i-CRRP with Algorithm 1 and compare the overall travel time of customer-carrying vehicles of the solution (computed with the BPR link delay function) with the *optimal* congestion-free overall travel time of customer-carrying vehicles (computed with the CPLEX MILP solver). In the problem instance considered in this experiment, the optimal travel time of customer-carrying vehicles coincide with the optimal travel time under the LP relaxation. However, we stress that the linear relaxation does not yield integral routes: thus, it is not suitable for real-time control of an AMoD system.

We consider 100 realizations of the randomized rounding algorithm. Table I summarizes our results. Figure 2 shows the distribution of the ratio between the required number of vehicles of the sampled solution and the required number of vehicles of the LP solution.

TABLE I

Results of the numerical simulations

	LP	Rand. routing
Avg. cust. travel time [s]	91.552	91.716
Avg. num. congested edges	233	333.7
Avg. cong. (as a fraction of link capacity) on congested edges	36.5%	25.6%

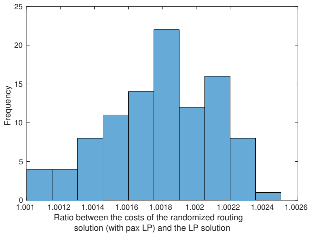

Fig. 2. Distribution of the ratio between the overall customer travel time of the randomized routing solution and the overall travel time of the LP.

Interestingly, for some simulations with low congestion (not shown for brevity), we observed that the overall customer travel time required by the sampled solution is sometimes smaller than the customer travel time in the LP relaxation. This counterintuitive result is due to the fact that the LP relaxation computes a congestion-free solution, even if this results in longer paths for the customers. The randomized routing algorithm, on the other hand, sometimes samples shorter paths that induce a small amount of congestion but, overall, result in smaller travel times.

The average execution time of Algorithm 1 on a commodity PC is 655 s; this time is dominated by the time required to solve Problem 4. In contrast, the execution time of the MILP solver is 950 s. Thus, the algorithm finds a high-quality solution 45% faster than than the exact solver.

Algorithm 1 is not, in general, guaranteed to be faster than the MILP solver (indeed we observed that, for espe-

[1] Courtesy of the New York Taxi and Limousine Commission.

cially simple instances of the i-CRRP, performance of the MILP solver is comparable to our algorithm). Nevertheless, for large-scale problems, the randomized routing algorithm guarantees that the solution time will *always* be polynomial in the problem size (an especially relevant concern for real-time control) and the value of the solution will be close to the optimal one.

B. Performance of a receding-horizon implementation

Algorithm 1 holds promise as a real-time algorithm for the control of fleets of AMoD vehicles. While a receding-horizon implementation is beyond the scope of this paper, we compare the performance of Algorithm 1 with the performance of a state-of-the-art receding-horizon rebalancing algorithm [10] for a single problem instance. This instance can be seen as proxy for a single step of a receding-horizon algorithm. We consider the same New York City road network discussed in Section IV-A; we randomly sample 1000 customer arrivals (approximately corresponding to the number of customers arriving in two minutes in the original data set) and scale road capacity accordingly. The receding-horizon algorithm in [10] only computes rebalancing routes; customer-carrying vehicles are greedily routed along the fastest path.

We considered fifty realizations of the customer requests: on average, the overall travel time of customer-carrying vehicles was $5.28 \pm 0.72\%$ lower with Algorithm 1. Thus, the a receding-horizon implementation of the algorithm holds promise for real-time control of city-scale AMoD systems.

V. Conclusions and Future Work

In this paper we presented a randomized algorithm for simultaneously computing customer routes and rebalancing routes on a capacitated road network in an autonomous mobility-on-demand system. Our goal is to find a set of routes that minimize the total travel time of customer-carrying vehicles: since the problem is intractable for generic congestion models, we adopt a simple threshold congestion model that yields congestion-free routes. We formulate the routing and rebalancing problem as an integral Minimum-Cost Multi-Commodity Flow problem and, after showing that even this simple problem is NP-hard, we develop a sampling technique that extends known randomized routing algorithms to flows with multiple origins and destinations. The sampling technique may violate some of the congestion constraints: nevertheless we prove that the expected overall travel time of customer-carrying vehicles produced by the randomized algorithm under a realistic congestion model remains very close to the optimal congestion-free travel time (a proxy for the optimal travel time). Numerical results on a realistic Manhattan road network show that the travel time of customers required by our algorithm is very close to the optimal travel time required by a congestion-free solution. Preliminary experiments show that a receding-horizon implementation of the algorithm yields promising performance when compared with state-of-the-art rebalancing algorithms.

This work paves the way for the development of large-scale congestion-aware routing and rebalancing algorithms for AMoD systems. Our randomized algorithm is easily scalable and can offer real-time performance. Future work will explore a receding-horizon, closed-loop implementation of the algorithm and integration with state-of-the-art traffic simulators such as MATSIM and SUMO and characterize its performance in presence of stochastic fluctuations in the demand, travel times, and routing distribution. Additionally we would like to study other ways of reducing congestion such as staggering demands, ride-sharing, and integration with public transportation.

References

[1] M. Barth and M. Todd, “Simulation model performance analysis of a multiple station shared vehicle system,” *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 4, pp. 237–259, 1999.

[2] 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*, 2013.

[3] R. Zhang and M. Pavone, “A queueing network approach to the analysis and control of Mobility-on-Demand systems,” in *American Control Conference*, 2015.

[4] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, “Load balancing for Mobility-on-Demand systems,” in *Robotics: Science and Systems*, 2011.

[5] R. Zhang and M. Pavone, “Control of robotic Mobility-on-Demand systems: A queueing-theoretical perspective,” *Int. Journal of Robotics Research*, vol. 35, no. 1-3, pp. 186–203, 2016.

[6] 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.

[7] ——, “Models and efficient algorithms for pickup and delivery problems on roadmaps,” in *Proc. IEEE Conf. on Decision and Control*, 2012.

[8] B. Templeton. (2010) Traffic congestion & capacity. Available at http://www.templetons.com/brad/robocars/congestion.html.

[9] M. W. Levin, T. Li, S. D. Boyles, and K. M. Kockelman, “A general framework for modeling shared autonomous vehicles,” in *95th Annual Meeting of the Transportation Research Board*, 2016.

[10] R. Zhang, F. Rossi, and M. Pavone, “Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms,” in *Robotics: Science and Systems*, 2016.

[11] B. N. Janson, “Dynamic traffic assignment for urban road networks,” *Transportation Research Part B: Methodological*, vol. 25, no. 2–3, pp. 143–161, 1991.

[12] M. J. Lighthill and G. B. Whitham, “On kinematic waves. II. a theory of traffic flow on long crowded roads,” *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[13] M. Treiber, A. Hennecke, and D. Helbing, “Microscopic simulation of congested traffic,” in *Traffic and Granular Flow ’99*. Springer Berlin Heidelberg, 2000, pp. 365–376.

[14] J. G. Wardrop, “Some theoretical aspects of road traffic research,” *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, 1952.

[15] Bureau of Public Roads, “Traffic assignment manual,” U.S. Department of Commerce, Urban Planning Division, Tech. Rep., 1964.

[16] S. Even, A. Itai, and A. Shamir, “On the complexity of timetable and multicommodity flow problems,” *SIAM Journal on Computing*, vol. 5, no. 4, pp. 691–703, 1976.

[17] A. Srinivasan, “A survey of the role of multicommodity flow and randomization in network design and routing,” in *Randomization Methods in Algorithm Design*, vol. 43, 1999, pp. 271–302.

[18] P. Raghavan and C. D. Tompson, “Randomized rounding: A technique for provably good algorithms and algorithmic proofs,” *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[19] A. Goldberg, J. Oldham, S. Plotkin, and C. Stein, “An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow,” in *Int. Conf. on Integer Programming and Combinatorial Optimization*, 1998, pp. 338–352.

[20] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[21] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.

[22] R. M. Karp, “On the computational complexity of combinatorial problems,” *Networks*, vol. 5, no. 1, pp. 45–68, 1975.

[23] D. P. Dubhashi and D. Ranjan, “Balls and bins: A study in negative dependence,” *BRICS Report Series*, vol. 3, no. 25, pp. 1–27, 1996.

[24] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.