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

Federico Rossi, Rick Zhang, and Marco Pavone

Abstract— **In this paper we study the routing and rebalancing problem for a fleet of autonomous vehicles providing on-demand transportation within a congested urban road network. We leverage a network flow model with integral flows, where unit flows represent individual vehicle routes. We show that the integral *congestion-free* routing and rebalancing problem is NP-hard and provide a randomized algorithm which finds a *low-congestion* solution to the routing and rebalancing problem in polynomial time. We provide theoretical bounds on the probability of violating the congestion constraints; we also characterize the expected travel times of the solution with a commonly-used empirical congestion model. Numerical experiments on a realistic road network with real-world customer demands show that our algorithm introduces very small amounts of congestion. The performance of our algorithm in terms of travel times (taking into account congestion) is very close to (and sometimes better than) the optimal 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). Motivation to address these challenges have spurred the development of several key enabling technologies including vehicle sharing, electric vehicles, and autonomous vehicles. These technologies converge as autonomous mobility-on-demand (AMoD) – a future transportation system whereby a fleet of autonomous, self-driving cars service customers within a dense urban environment. Aside from potential benefits in safety and cost, 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 the AMoD system can be routed cooperatively to minimize trip duration and traffic congestion while providing high quality of service by anticipating future customer demands through *rebalancing* (redistribution of empty vehicles).

Solving this cooperative routing and rebalancing problem for large scale systems on an individual vehicle level is challenging. It can be shown that the static problem of routing only customer trips (without rebalancing) in a capacity-constrained road network is NP-hard [1]. The objective of this paper is to develop a randomized routing framework to solve both the customer routing problem and the rebalancing problem on a large scale in the presence of network congestion with guaranteed bounds on the quality of the solution.

Literature review: The rebalancing problem has been studied for carsharing systems [2], [3], [4] and AMoD systems [5], [6] where the underlying network is a complete graph (point-to-point routing). These approaches (i) seek to only optimize empty vehicle routes (rebalancing routes), and (ii) do not consider congestion effects caused by routing customers and empty vehicles on a road network. The routing and rebalancing problem we study in this paper is similar to one-to-one pickup and delivery problems on a Euclidean plane [7] or on a road network [8] for which combinatorial asymptotically optimal algorithms exist. However, these formulations do not take into account congestion. The presence of empty, rebalancing vehicles on the road was believed to have a negative impact on congestion in a transportation network [9], [10], but recent work suggests that with intelligent routing, this need not be the case [11]. The routing and rebalancing problem on a congested network was studied in [11] using a fluidic model, which, while providing insights on macroscopic system behavior, does not directly allow the computation of individual vehicle routes. In this paper we focus on the computation of individual paths for vehicles: to this end, we adapt the model in [11] to accommodate integral flows.

In the field of transportation science, our problem is similar to dynamic traffic assignment (DTA) problems [12]. 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 only lead to an increase in its travel time. A key advantage of AMoD systems is cooperative behavior between the vehicles: 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. Congestion modeling has been studied at many different degrees of fidelity from basic models establishing the relationship between speed, density, and flow [13], to simulation-based microscopic car-following models [14]. 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. Hence, we leverage simple traffic models that are amenable to tractable analysis and control.

Statement of contributions: Our ultimate goal is to find optimal vehicle routes that minimize the overall travel times of all vehicles in a congested road network. However, evaluating travel times in a congested network requires congestion models which are in general complicated and not amenable to control synthesis. Thus, we apply a simple congestion model (a threshold model) to compute *congestion-free* vehicle routes as a *proxy* for solving the general problem of minimizing travel times. We develop a scalable polynomial

This research was supported by National Science Foundation under CAREER Award CMMI-1454737 and by the Dr. Cleve B. Moler Stanford Graduate Fellowship.

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

time randomized algorithm based on the congestion-free solution and evaluate its performance against the congestion-free solution in terms of overall travel times using a common empirical congestion model.

Specifically, our contributions are threefold: first, we propose a randomized routing technique for integral multi-commodity flows that produces solutions that violate the capacity constraints by small amounts, and we provide a probabilistic characterization of the degree of constraint violation. The technique extends well-known results in combinatorial optimization [15] to the case where the flows have multiple origins and destinations and is applicable to general routing problems where small violations of the capacity constraints are acceptable. Second, we provide a semi-analytical characterization of the expected travel time produced by our randomized technique under a widely adopted empirical congestion model. We then provide an upper bound on the ratio between the travel time produced by our algorithm and the optimal *congestion-free* travel time. Third, we validate the performance of the randomized routing algorithm on a realistic road network with real customer demands. The performance of the randomized technique compares very favorably with the optimal congestion-free solution: indeed, the randomized routing algorithm occasionally yields *better* travel times than the best congestion-free solution, since it selects shorter (if slightly congested) routes.

Organization: The rest of this paper is organized as follows: in Section II we describe the network flow model of an AMoD system and define the integer congestion-free routing and rebalancing problem. In Section III we describe a randomized algorithm to solve the routing and rebalancing problem and provide guaranteed bounds on the expectation of the cost of the algorithm (in terms of travel times) with respect to the linear relaxation of the original problem. Numerical results are provided in Section IV validating the performance of the randomized algorithm, and conclusions and future directions are summarized in Section V.

II. Model Description

We represent 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 [11], the key difference being 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 paths, which can then be used as part of a practical real-time control algorithm for large vehicle fleets.

A. Congestion model

In this section we present two congestion models: a simpler *synthesis model*, used in the definition of our problem, and a more general *analysis model* that offers a better representation of congested traffic behavior.

Both models are consistent with classical traffic flow theory [13]. In classical traffic flow theory, for a given road link, vehicle speeds tend to remain relatively constant at low vehicle densities (called the “free flow” speed). The flow rate in this regime is given by the product of the speed and the vehicle density (when the speed is constant, the flow rate is directly proportional to density). The flow rate grows with density up to a critical value (referred to in the literature as the *capacity* of the road link), at which point vehicle speeds decrease significantly, signaling the onset of congestion. The capacity of the road link is reached when the flow rate is maximized.

Our synthesis congestion model is a threshold model. The vehicle density on each link is constrained to be no larger than the critical link density, which corresponds to the link 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 congestion constraints on every link is guaranteed to be *congestion-free*.

Our analysis congestion model offers a characterization of the congested behavior of a road link. If the density of vehicles on a road is smaller than the road capacity, all vehicles travel at the free flow speed, as in the synthesis model. If the road capacity is exceeded, the speed of each vehicle is a monotonically decreasing function of the vehicle density. Specific congestion functions will be presented and considered in Section III-C.

The synthesis model is used to optimize route assignments for customers and rebalancing vehicles that are, in expectation, congestion-free. The analysis model allows us to bound the expected travel times on these routes. In our numerical experiments (Section IV), we evaluate the congested travel times, calculated using a well-known congestion function, along routes computed using the synthesis model.

B. Network flow model of AMoD system

We model the road network as a capacitated graph $G(\mathcal{V}, \mathcal{E})$. The nodes $v \in \mathcal{V}$ represent intersections and locations of customer trip origins and destinations. The edges $(u, v) \in \mathcal{E}$ represent road links. Congestion is modeled as a constraint on the number of vehicles that a road link can accommodate, in accordance with the synthesis congestion model: a function $c(u, v) : \mathcal{E} \mapsto N_{\geq 0}$ denotes the capacity[1] of each link.

We assume that the road network is *capacity-symmetric*: for any node $v \in \mathcal{V}$, the overall capacity of the edges entering v equals the overall capacity of all edges exiting v, that is

$$\sum_{u \in \mathcal{V}:(u,v) \in \mathcal{E}} c(u, v) = \sum_{w \in \mathcal{V}:(v,w) \in \mathcal{E}} c(v, w)$$

It is easy to verify this definition is equivalent to the notion of capacity-symmetry presented in [11].

All vehicles on a link travel at the free-flow speed. The corresponding free-flow travel time across the road link is denoted by $t(u, v) : \mathcal{E} \mapsto \mathbb{R}_{>0}$.

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. Transportation requests are assumed to be stationary and deterministic: λ is constant in time. The set of transportation requests is denoted as $\mathcal{M} = \{(s_m, t_m, \lambda_m)\}_m$.

[1] We use capacity to refer to the number of vehicles a link can accommodate. This is different from the normal usage of “capacity” found in the literature, which refers to maximum flow rate. However, due to the constant speed assumption, the two usages can be viewed as equivalent.

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 the 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 passenger origins.

We model customer routes and rebalancing routes as *flows* of customers and vehicles on the graph. The concept of network flows 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 equations:

$$\sum_{u \in \mathcal{V}:(u,s) \in \mathcal{E}} f(u, s) + \lambda = \sum_{w \in \mathcal{V}:(s,w) \in \mathcal{E}} f(s, w) \tag{1}$$

$$\sum_{u \in \mathcal{V}:(u,t) \in \mathcal{E}} f(u, t) = \lambda + \sum_{w \in \mathcal{V}:(t,w) \in \mathcal{E}} f(t, w) \tag{2}$$

$$\sum_{u \in \mathcal{V}:(u,v) \in \mathcal{E}} f(u, v) = \sum_{w \in \mathcal{V}:(v,w) \in \mathcal{E}} f(v, w) \quad \forall v \in \{\mathcal{V} \setminus \{s, t\}\} \tag{3}$$

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

$$\sum_{u \in \mathcal{V}} f(u, v) + \sum_i 1_{v=s_i} \lambda_i = \sum_{w \in \mathcal{V}} f(v, w) + \sum_j 1_{v=t_j} \lambda_j \quad \forall v \in \mathcal{V} \tag{4}$$

Note that Equation 4 can be satisfied at all nodes only if $\sum_i \lambda_i = \sum_j \lambda_j$, i.e. if the sum of the intensities of all origins equals the sum of intensities of all destinations.

Path flows form the link between network flows and customer routes. 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 Equations 1, 2 and 3 and only assigns positive flow to edges belonging to a path p going from s to t: $\exists p = \{(s, u), (u, v), \ldots, (w, t)\} : f^p(u, v) > 0 \Leftrightarrow (u, v) \in p$. Note that, if the path contains no repeated edges, each edge is assigned the same value in a path flow: $f^p(u, v) = \lambda_p \forall (u, v) \in p$. If an edge is repeated k times in the path, its path flow is $k\lambda_p$. For this reason, the pair $\{p, \lambda_p\}$ is an equivalent representation of a given path flow. A customer route can be equivalently described as a path flow of intensity 1 by assigning value $f(u, v) = 1$ to the edges contained in the route and $f(u, v) = 0$ otherwise.

Any network flow with multiple sources and multiple sinks can be decomposed into a collection of path flows connecting the sources and the sinks [16]. In general, the path flows resulting from such a decomposition have fractional intensity. However, if the network flow is integral (that is, if $f(u, v) \in \mathbb{N}_{\geq 0}$ for every edge, the resulting path flows are also integral and therefore represent a collection of routes.

C. The integral Congestion-free Routing and Rebalancing problem (i-CRRP)

The goal is to compute a set of routes for the autonomous vehicles that (i) transfer customers to their desired destinations (customer-carrying trips) and (ii) rebalance vehicles throughout the network to realign the vehicle fleet with the customers' demands (customer-empty trips), while minimizing the overall duration of the customer-carrying trips and customer-empty trips. We describe the passenger routes for passenger $m \in \mathcal{M}$ with the integral network flow $\{f_m(u, v)\}_{(u,v)}$ and rebalancing routes with the integral network flow $\{f_R(u, v)\}_{(u,v)}$. As mentioned in Section II-B, integral network flows are an equivalent description for routes: once a (customer or rebalancing) network flow is found, it can be decomposed into a collection of (customer or rebalancing) routes.

We define the integral Congestion-free Routing and Rebalancing problem (i-CRRP) as follows. Given a capacitated network $G(\mathcal{V}, \mathcal{E})$ and a set of transportation requests $\mathcal{M} = \{(s_m, t_m, \lambda_m)\}$, solve

$$\underset{f_m(\cdot,\cdot), f_R(\cdot,\cdot)}{\text{minimize}} \sum_{m \in \mathcal{M}} \sum_{(u,v) \in \mathcal{E}} t(u, v) f_m(u, v) + \sum_{(u,v) \in \mathcal{E}} t(u, v) f_R(u, v) \tag{5}$$

$$\text{subject to} \sum_{u \in \mathcal{V}} f_m(u, s_m) + \lambda_m = \sum_{w \in \mathcal{V}} f_m(s_m, w) \;\; \forall m \in \mathcal{M} \tag{6}$$

$$\sum_{u \in \mathcal{V}} f_m(u, t_m) = \lambda_m + \sum_{w \in \mathcal{V}} f_m(t_m, w) \;\; \forall m \in \mathcal{M} \tag{7}$$

$$\sum_{u \in \mathcal{V}} f_m(u, v) = \sum_{w \in \mathcal{V}} f_m(v, w) \quad \forall m \in \mathcal{M}, v \in \mathcal{V} \setminus \{s_m, t_m\} \tag{8}$$

$$\sum_{u \in \mathcal{V}} f_R(u, v) + \sum_{m \in \mathcal{M}} 1_{v=t_m} \lambda_m = \sum_{w \in \mathcal{V}} f_R(v, w) + \sum_{m \in \mathcal{M}} 1_{v=s_m} \lambda_m \quad \forall v \in \mathcal{V} \tag{9}$$

$$f_R(u, v) + \sum_{m \in \mathcal{M}} f_m(u, v) \leq c(u, v) \;\; \forall (u, v) \in \mathcal{E} \tag{10}$$

$$f_m(u, v) \in \mathbb{N}_{\geq 0} \;\; \forall m \in \mathcal{M}, (u, v) \in \mathcal{E} \tag{11}$$

$$f_R(u, v) \in \mathbb{N}_{\geq 0} \;\; \forall (u, v) \in \mathcal{E} \tag{12}$$

Equations 6, 7, 8 and 11 ensure that each customer-carrying flow $\{f_m(u, v)\}_{(u,v),m}$ is an integral network flow. Equations 9 and 12 ensure that the rebalancing flow $\{f_R(u, v)\}_{(u,v)}$ is an integral network flow. Finally, Equation 10 enforces the capacity constraint of every link.

D. Complexity of the i-CRRP

The i-CRRP can be cast as an instance of the integral Minimum-Cost Multi-Commodity Flow (Min-MCF) problem. Given a capacitated graph and a set of integral commodity requests, the decision version of the integral Min-MCF seeks a feasible set of commodity flows or a certificate of infeasibility. The problem is known to be NP-complete [17], [1] in the general case; therefore the Min-MCF problem is NP-hard. It follows that the i-CRRP is NP-hard, and even determining whether an instance of the i-CRRP is feasible is NP-hard.

III. A Randomized Routing algorithm

While polynomial-time exact algorithms are known for variants of the integral MCF on specific network topologies [18], these techniques do not extend to the Min-MCF problem. Polynomial-time approximation schemes are known for variants of the integral multi-commodity flow problem such as Max-MCF [19]; however, to the best of our knowledge, no such approximation schemes are known for Min-MCF.

If all commodities have a single origin and destination, randomized routing techniques can be employed to find integral solutions to the Min-MCF that violate the congestion constraints with some small probability and have optimal expected cost [20], [15]. However, in the integral version of the CRRP, the rebalancing flow has multiple origins and destinations: naive randomized routing cannot guarantee that the integral rebalancing flow will satisfy continuity constraints (Equation 9) at every origin and destination.

In this paper, we present a randomized routing technique that yields integral passenger and rebalancing flows (hence, passenger and rebalancing routes) that satisfy the continuity equations at every origin and every destination; the flows may violate the congestion constraints with some (small) probability. The technique extends existing randomized routing techniques [15] to the case where flows have multiple origins and destinations and can be applied to general Min-MCF problems.

The procedure, described in detail in the next two sections, can be summarized as follows: (i) we first compute a solution to the LP relaxation of the i-CRRP and decompose the solution into fractional customer routes and fractional rebalancing solutions (Section III-A), (ii) we sample the fractional customer routes and rebalancing solutions to obtain a solution to the i-CRRP that may violate some of the congestion constraints (Section III-B). In Section III-C we characterize the degree of violation of the congestion constraints and the implications on travel times if a more general congestion model is used.

Without loss of generality, we restrict our analysis to the case where each customer flow has unit intensity: a customer flow of intensity ϕ is modeled as ϕ customer flows of unit intensity between the same origin and destination nodes.

A. Linear relaxation of the CRRP and flow decomposition

The first step in our randomized routing technique is to solve a linear relaxation of the i-CRRP, which we denote as the *fractional* CRRP. To obtain the fractional CRRP, we replace Equations 11 and 12 with

$$f_m(u,v) \in \mathbb{R}_{\geq 0} \qquad \forall m \in \mathcal{M}, (u,v) \in \mathcal{E} \tag{13}$$

$$f_R(u,v) \in \mathbb{R}_{\geq 0} \qquad \forall (u,v) \in \mathcal{E} \tag{14}$$

The fractional CRRP can be reduced to an instance of the fractional Min-MCF problem, which can be efficiently solved directly as a linear program (of size $|\mathcal{E}|(M+1)$) or via specialized combinatorial algorithms, e.g. [21].

The solution of the fractional CRRP produces a network flow for each customer and a rebalancing network flow.

Next, we decompose each customer's network flow into a collection of path flows. Ford-Fulkerson's flow decomposition algorithm (reported in [16] and also in the Appendix as Algorithm 4) performs this decomposition in $O(|\mathcal{E}|)$ time.

Computing a suitable decomposition of the rebalancing flow is more challenging. Without loss of generality, we restrict our attention to rebalancing network flows where each origin and each destination has unit intensity. If a node s is an origin with intensity $\phi > 1$, we (i) replace it with a "dummy" node, $\tilde{s}$, with zero intensity and (ii) generate ϕ origin nodes, each with unit intensity, connected to $\tilde{s}$ with edges of infinite capacity and zero travel time. Destination nodes are treated identically. If a node is both an origin and a destination of rebalancing flow, with origin flow ϕ_o and destination flow ϕ_d, we transform it into an origin node with flow $\phi_o - \phi_d$ if $\phi_o > \phi_d$, a destination node with flow $\phi_d - \phi_o$ if $\phi_d > \phi_o$, or a regular node that is neither an origin nor a destination if $\phi_o = \phi_d$.

Ford-Fulkerson's flow decomposition algorithm can be used to decompose the rebalancing flow $\{f_R(u,v)\}_{(u,v)}$ into a collection of path flows, each connecting a rebalancing origin with a rebalancing destination. However it is not possible to *independently* sample the rebalancing path flows and guarantee that each rebalancing origin and each rebalancing destination has exactly one incident path. To overcome this, we decompose the rebalancing flow into a set of *fractional rebalancing solutions*, which we define in the following.

Intuitively, a fractional rebalancing solution r is a collection of paths that rebalances every origin and every destination, associated with an intensity λ_r (which may be interpreted as a probability). More rigorously, consider a collection $\mathcal{P}$ of path flows $\{f_p(u,v)\}_{(u,v),p}$, each of intensity λ_r, such that, for every origin node $o \in \mathcal{V}$,

$$\sum_{p \in \mathcal{P}} \sum_{u:(u,v)\in\mathcal{E}} f_p(u,v) + \lambda_r = \sum_{p \in \mathcal{P}} \sum_{w:(v,w)\in\mathcal{E}} f_p(v,w)$$

and for every destination node $d \in \mathcal{V}$

$$\sum_{p \in \mathcal{P}} \sum_{u:(u,v)\in\mathcal{E}} f_p(u,v) = \sum_{p \in \mathcal{P}} \sum_{w:(v,w)\in\mathcal{E}} f_p(v,w) + \lambda_r.$$

Then the sum of these path flows $\{f_{re}(u,v)\}_{u,v}$, with

$$f_{re}(u,v) = \sum_{p \in \mathcal{P}} f_p(u,v) \quad \forall (u,v) \in \mathcal{E}$$

is a fractional rebalancing solution with intensity λ_r.

Algorithm 1 computes such a decomposition for any rebalancing flow $\{f_R(u,v)\}_{(u,v)}$.

Algorithm 1 Rebalancing flow decomposition in fractional rebalancing solutions.

Input: a rebalancing flow $\{f_R(u,v)\}_{(u,v)}$ with sources and sinks of equal intensity ϕ
Output: A collection of fractional rebalancing solutions $\mathcal{S}$.
Initialize $\mathcal{S} = \emptyset$.
$\mathcal{S} \leftarrow$ RebDecomposition($f_R, \mathcal{S}$)
procedure RebDecomposition($f, \mathcal{S}$)
    $\mathcal{P} \leftarrow$FlowDecomposition(f)
    Find a fractional rebalancing solution f_{re} embedded in $\mathcal{P}$ that contains at least one path $p \in \mathcal{P}$
    $\mathcal{S} \leftarrow$ RebDecomposition($f_R - f_{re}, \mathcal{S} \cup f_{re}$)
    return $\mathcal{S}$

The two following results prove the correctness of Algorithm 1.

Lemma 3.1: Consider a graph $G(\mathcal{V}, \mathcal{E})$ and a fractional rebalancing flow $\{f_R(u,v)\}_{(u,v)}$ such that each origin and each destination has the same intensity $\phi \in \mathbb{R}_{\geq 0}$. Also consider a flow decomposition $\mathcal{F}$ of $\{f_R(u,v)\}_{(u,v)}$, i.e. a collection of path flows $p \in \mathcal{F}$ such that, for any edge $(u,v) \in \mathcal{E}$, $\sum_{p\in\mathcal{F}} f_p(u,v) = f_R(u,v)$. Then there exists a rebalancing solution f_{rs} of intensity $\sigma \leq \phi$ such that (i) for every edge $(u,v) \in \mathcal{E}$, $f_{rs}(u,v) \leq f_R(u,v)$ and (ii) f_{rs} contains at least one path flow in $\mathcal{F}$, that is, $\exists p \in \mathcal{F}$: $f_{rs}(u,v) \geq f_p(u,v) \, \forall (u,v) \in p$.

Lemma 3.2: For any rebalancing flow $\{f_R(u,v)\}_{(u,v)}$, Algorithm 1 decomposes the flow in a collection of rebalancing solutions $\{f^i_{rs}(u,v)\}_{(u,v),i}$ such that, for every edge $(u,v) \in \mathcal{E}$, $\sum_i f^i_{rs}(u,v) = f_R(u,v)$. The algorithm terminates in at most $\|\mathcal{E}\|$ iterations.

The proofs of Lemmas 3.1 and 3.2 are reported in the Appendix.

In order to find a fractional rebalancing solution $\{f_{re}\}_{(u,v)}$ embedded in the path decomposition $\mathcal{P}$ of the rebalancing flow $\{f_R\}_{(u,v)}$ that contains at least one path $p \in \mathcal{P}$, we solve the following linear program. We set σ as the smallest intensity among the path flows $p \in \mathcal{P}$. We define $\mathcal{O}$ as the set of origin nodes and $\mathcal{D}$ as the set of destination nodes. We then solve

$$\text{Find} \quad f_{re}(u,v) \;\; \forall (u,v) \in \mathcal{E} \tag{15}$$

$$\text{subject to} \sum_{s\in\mathcal{V}} f_{re}(u,v) + \sum_{o\in\mathcal{O}} 1_{v=o}\sigma = \sum_{w\in\mathcal{V}} f_{re}(v,w) + \sum_{d\in\mathcal{D}} 1_{v=d}\sigma \quad \forall v \in \mathcal{V} \tag{16}$$

$$f_{re}(u,v) \leq f_R(u,v) \;\; \forall (u,v) \in \mathcal{E} \tag{17}$$

$$f_{re}(u,v) \in \mathbb{R}_{\geq 0} \;\; \forall (u,v) \in \mathcal{E} \tag{18}$$

Lemma 3.1 shows that the linear program above is always feasible. One can show that the LP can be transformed in an instance of the single-commodity flow problem with unit flows and integral link capacities, and therefore enjoys a totally unimodular constraint matrix: thus, the problem admits a solution $\{f_{re}\}_{(u,v)}$ where all link flows are either zero or multiples of σ. It is then easy to show that the resulting flow is a fractional rebalancing solution of weight σ. We refer the reader to the proof of Lemma 3.1 in the Appendix for a more thorough discussion.

B. Sampling passenger and rebalancing routes

Ford-Fulkerson's flow decomposition algorithm produces a flow decomposition of each customer flow $\{f_m(u,v)\}_{(u,v)}$: for each customer $m \in \mathcal{M}$, it generates a collection $\mathcal{F}_m$ of path flows $p \in \mathcal{F}_m$. The output of Algorithm 1 is a collection of fractional rebalancing solutions $r \in \mathcal{S}$.

We are now in a position to sample $\mathcal{F}_m$ and $\mathcal{S}$ to produce a collection of integral customer and rebalancing flows. For each customer $m \in \mathcal{M}$, we independently sample exactly one path flow from $\mathcal{F}_m$ with probability equal to the intensity of the path flow (note that, since we restrict our analysis to origins and destinations with unit intensity, the sum of the intensities of the path flows in $\mathcal{F}_m$ is indeed 1). We assign the sampled path to the customer and change its intensity to 1. We call the corresponding path flow $\{f^s_m(u,v)\}_{(u,v),m}$.

We also independently sample one fractional rebalancing solution from $\mathcal{S}$ with probability equal to the intensity of the rebalancing solution. Again, since we restrict our analysis to the case where rebalancing origins and rebalancing destinations all have unit intensity, the sum of the intensities of all rebalancing solutions is 1. We set the intensity of the sampled rebalancing solution to one; we call the resulting rebalancing flow $\{f^s_R(u,v)\}_{(u,v)}$.

The procedure is summarized in Algorithm 2.

Algorithm 2 Sample customer path flows and the fractional rebalancing solution to obtain integral routes.

Input: A collection of path flows for each customer$\{\mathcal{P}_m\}_m$
A collection of fractional rebalancing solutions $\mathcal{S}$
Output: A path flow p_m of unit intensity for each customer.
An integral rebalancing flow
procedure SAMPLEPATHS($\{\mathcal{P}_m\}_m, \mathcal{S}$)
  for $m \in \mathcal{M}$ **do**
    Sample one path flow $\{p_m, \lambda_m\}$ from $\mathcal{P}_m$ w.p. λ_m
    Set the intensity of the sampled path flow to $\lambda_m = 1$
  Sample one fractional rebalancing solution $\{f_R(u,v)\}_{(u,v)} \in \mathcal{S}$ with probability λ_r
  Set the intensity of the rebalancing solution to $\lambda_r = 1$.
  return $\{p_m\}_m, \{f_R(u,v)\}_{(u,v)}$

Algorithm 3 summarizes the full algorithm for finding an integral solution to the i-CRRP in polynomial time, using Algorithms 1 and 2 and Ford-Fulkerson's flow decomposition algorithm as subroutines.

Algorithm 3 Randomized routing computation of a low-congestion solution to the i-CRRP

Input: An instance of the i-CRRP
Output: A route r_m for each customer m.
A collection $\{r_{reb}\}$ of rebalancing routes.
$\{f_m\}_{(u,v),m}, \{f_R\}_{(u,v)} \leftarrow$ solve the linear relaxation of the i-CRRP
for all $m \in \mathcal{M}$ **do**
  $\{\mathcal{F}_m\} \leftarrow$ FLOWDECOMPOSITION($\{f_m\}$)
$\mathcal{S}_R \leftarrow$ REBDECOMPOSITION($f_R, \emptyset$)
$\{r_m\}_m, \{f_R\}_{(u,v)} \leftarrow$SAMPLEPATHS($\{\mathcal{F}_m\}_m, \mathcal{S}_R$)
$r_{reb} \leftarrow$FLOWDECOMPOSITION($\{f_R\}_{(u,v)}$)

Theorem 3.3 (Runtime of Algorithm 3): The runtime of Algorithm 3 is polynomial in the size of the i-CRRP.

The proof of Theorem 3.3 is omitted in the interest of brevity and can be found in the Appendix.

By construction, each customer is assigned a path flow of intensity 1 and the rebalancing solution is a network flow for the rebalancing origins and destinations. Furthermore, $\{f^s_m(u,v)\}_{(u,v)}$ and $\{f^s_R(u,v)\}_{(u,v)}$ only assume integer values. Therefore the flows satisfy Equations 6, 7, 8, 9, 11 and 12. The expected number of (customer and rebalancing) paths crossing every edge $(u,v) \in \mathcal{E}$ is upper-bounded by the capacity of the edge $c(u,v)$:

$$\mathbb{E}\left[\sum_m f^s_m(u,v) + f^s_R(u,v)\right] = \sum_m f_m(u,v) + f_R(u,v) \le c(u,v)$$

since each path flow, as well as the rebalancing solution, is sampled with probability equal to its intensity. On the other hand, while the output of the algorithm satisfies the capacity constraints *in expectation*, a given realization may violate them.

C. Analysis

In this section we analyze the performance of the randomized routing procedure, Algorithm 3. First, we characterize the probability that a congestion constraint is violated. Next, we quantify the effect of the violation of the congestion constraints on the travel time under the analysis congestion model: specifically, we prove a bound on the ratio between the expected overall travel time and the travel time of the fractional, congestion-free solution.

The output of Algorithm 3 may violate the capacity constraints. However, the degree to which these constraints may be violated can be characterized exactly.

Theorem 3.4 (Performance of Algorithm 3): Consider a fractional solution to the CRRP. With probability $1-\alpha$, Algorithm 3 finds an integral solution of the i-CRRP such that no congestion constraint for edges $(u,v)\in\mathcal{E}$ is violated by more than a multiplicative factor $\delta_{u,v} = \sqrt{3/c(u,v)\log(|\mathcal{E}|/\alpha)}$, if $\delta_{u,v}\le 1$ for all (u,v).

Proof: The proof relies on a Chernoff bound for every congestion constraint and a union bound.

Congestion on a single edge: We define the random variable $X(u,v)$ as the number of paths selected by Algorithm 3 that crosses any one edge (u,v). $X(u,v)$ has a Poisson binomial distribution and its mean is upper-bounded by the capacity of the edge $c(u,v)$. Therefore a Chernoff bound holds [23]:

$$\mathbb{P}\left(X(u,v) \ge (1+\delta)c(u,v)\right) \le \exp(-c(u,v)\delta^2/3) \quad \text{if } 0<\delta\le 1$$

Congestion on all edges: Select $\delta_{u,v}$ on every edge such that $c(u,v)\delta^2_{u,v} = c(w,t)\delta^2_{w,t}$ for every pair of edges$(u,v),(w,t)\in\mathcal{E}$. Then, by a union bound, the probability that at least one edge violates the congestion constraint is upper-bounded by

$$\mathbb{P}\left(\exists (u,v)\in\mathcal{E} : X(u,v) \ge (1+\delta_{u,v})c(u,v)\right) \le |\mathcal{E}|\exp(-c(u,v)\delta^2_{u,v}/3)$$

Let us call $\alpha = |\mathcal{E}|\exp(-c(u,v)\delta^2_{u,v}/3)$. Solving for $\delta_{u,v}$ proves our claim. ■

Remark A similar result can be found in the general case where $\delta_{u,v}\in\mathbb{R}_{>0}$ by exploiting a more general version of the Chernoff bound.

Theorem 3.4 gives a very conservative characterization of the performance of the randomized routing algorithm. In practical applications, a small violation of the congestion constraints merely result in an (often small) increase in the vehicles' travel times: if the relationship between the number of vehicles and the travel time is known in the congested regime, this increase can be quantified.

In our synthesis congestion model, we assume that the travel time on a link is a piecewise constant function of the number of vehicles on a link that assumes the value ∞ if the link is congested. Let us call $f(u,v) = \sum_m f_m(u,v) + f_R(u,v)$. Then the travel time on a link is:

$$t(u,v,f(u,v)) = \begin{cases} t(u,v) \text{ if } f(u,v)\le c(u,v) \\ \infty \text{ otherwise}\end{cases}$$

The analysis model provides a more accurate (if less actionable) description of the relation between congestion and travel time:

$$\tilde{t}(u,v,f(u,v)) = \begin{cases} t(u,v) \text{ if } f(u,v)\le c(u,v) \\ \tau^C(u,v,f(u,v)) \text{ otherwise}\end{cases}$$

where $\tau^C(u,v,f(u,v))$ is some function characterizing the congested travel time on a link. We assume that $\tau^C(u,v,f(u,v))$ is monotonically increasing in the overall number of vehicles $f(u,v)$ crossing the link.

Under this description of travel times, the expected travel time obtained by Algorithm 3 is

$$\sum_{(u,v)\in\mathcal{E}} \mathbb{E}\left[\tilde{t}(u,v,f^s(u,v))f^s(u,v)\right]$$

with $f^s(u,v) = \sum_m f^s_m(u,v) + f^s_R(u,v)$.

Analytical computation of the expected values in the expression above for general congested travel time functions $\tilde{t}$ is generally not feasible.

However, numerical computation for specific fractional solutions and congested travel time functions can be carried out. Furthermore, and critically, the Chernoff bound guarantees that any sampled solution will yield a number of vehicles (and therefore a level of congestion) value very close to the average with high probability. Figure 1 gives a graphical depiction of this intuition. We select $t(u,v)=1$, $c(u,v)=50$, 150 flows each crossing (u,v) with probability 1/3. We use the Bureau of Public Roads (BPR) link delay model [24], a widely adopted empirical congestion model. In this example, the difference between the optimal fractional cost and the expected cost of a sampled solution is 1.53%.

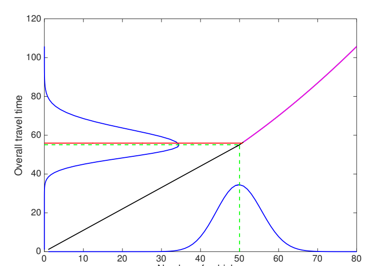

Fig. 1. Cumulative travel time on a link: fractional solution (dashed green) and expected value of a sampled solution (red). While some sampled solutions do violate the congestion constraint, the expected cost of the sampled solution is only 1.53% higher than the cost of the fractional solution. The BPR link delay model is used.

To formalize this intuition, we first provide two lemmas characterizing the distribution of travel times on a link, then in Theorem 3.7 we provide a bound on the ratio of expected overall travel times between Algorithm 3 and the fractional CRRP solution.

Lemma 3.5 (Expected overall travel time on a link): Consider a link $(u,v)\in\mathcal{E}$. Call X the number of vehicles traversing the link: $X = \sum_m f^s_m(u,v) + f^s_R(u,v)$. Then the expected overall travel time on link (u,v) under the analysis congestion model, $\tilde{t}(u,v,X)$ (i.e. $\mathbb{E}\left[\tilde{t}(u,v,X)X\right]$), admits an upper bound $U_{ott}(u,v,\mathbb{E}[X])$ that only depends on the congestion model for link (u,v) and on the expected flow $\mathbb{E}[X] = \sum_m f_m(u,v) + f_R(u,v)$ of the link.

The proof of Lemma 3.5 is omitted for brevity and is reported in the Appendix.

Lemma 3.5 can be used to characterize the ratio between the travel time of the solution produced by Algorithm 3 and the optimal travel time.

Lemma 3.6: (Expected approximation factor of the overall travel time on a link): Consider a link $(u,v) \in \mathcal{E}$. Call X the number of vehicles traversing the link. Then, if $\mathbb{E}[X] > 0$, the ratio between $\mathbb{E}\left[\tilde{t}(u,v,X)X\right]$ and $\tilde{t}(u,v,\mathbb{E}[X])\mathbb{E}[X]$, admits an upper bound $UR_{ott}(u,v,\mathbb{E}[X])$ that only depends on the congestion model for link (u,v) (and therefore on the capacity of the link) and on the expected flow $\mathbb{E}[X]$ on the link.

Proof: [Proof sketch] The proof is identical to the proof of Lemma 3.5: for a given link and a given expected number of vehicles $\mathbb{E}[X]$, the random variable $TT/\tilde{t}(\mathbb{E}[X])\mathbb{E}[X]$ only differs from the random variable TT by a constant multiplicative factor. ∎

The result in Lemma 3.6 can be used to characterize the ratio between the expected overall travel time on a link numerically. We use the Bureau of Public Roads (BPR) delay model to characterize the relationship between travel time and vehicle flow in the congested regime.

Figure 2 shows the bound UR_{ott} as a function of the link capacity and the link flow.

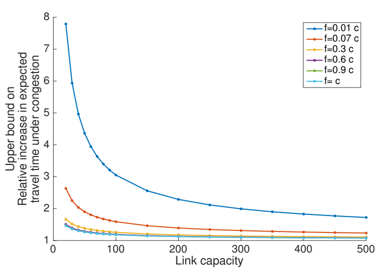

Fig. 2. Upper bound on the fractional increase in expected value of the overall travel time on a link as a function of link flow and link capacity. The BPR link delay model is used.

Our bound performs very poorly when the flow on a link is small with respect to the link capacity. However, crucially, when links experience significant traffic, the increase in overall travel time is very well controlled: the expected travel time on a congested link with capacity $c(u,v) = 50$ is within 30% of the optimum, and the ratio drops below 20% for links of capacity 100 and above.

These bounds on the expected overall travel time on a single link can be used to characterize the system-wide performance of our sampling technique. The following theorem formalizes this intuition.

Theorem 3.7 (Expected travel time): Consider an instance of the i-CRRP and an analysis link delay function $\tilde{t}(u,v,f)$. Also consider a fractional solution to the linear relaxation of the i-CRRP $\{f_m(u,v)\}_{(u,v),m}$, $\{f_R(u,v)\}_{u,v}$ and an integral solution IS obtained with Algorithm 3. Call $\{f(u,v)\}_{(u,v)} = \{\sum_m f_m(u,v) + f_R(u,v))\}_{(u,v)}$.

Then, for every link $(u,v) \in \mathcal{E}$, it is possible to compute an upper bound $UR_{ott}(u,v)$ on the ratio between the expected overall travel time of the integral solution IS under the analysis congestion function $\tilde{t}(u,v,f)$ and the travel time of the fractional solution. In addition, the ratio between the expected overall travel time of IS and the overall travel time of the uncongested solution is upper-bounded by

$$UB_{tt} = \frac{\sum_{(u,v)\in\mathcal{E}} UR_{ott}(u,v)\tilde{t}(u,v,f(u,v))f(u,v)}{\sum_{(u,v)\in\mathcal{E}} \tilde{t}(u,v,f(u,v))f(u,v)} \quad (19)$$

Proof: Lemma 3.6 proves that an upper bound on the ratio between the expected overall travel time and the travel time of the fractional solution can be found. The theorem then follows from linearity of expectation. ∎

Remark The bound UB_{tt} on the overall travel time (19) in Theorem 3.7 is a weighed average of the bounds on the individual links, with higher weight assigned to links with large flows (and therefore large capacity). The bound $UR_{ott}(c)$, shown in Figure 2, is strongly decreasing in the link flow and the link capacity: therefore, the bound in Theorem 3.7 becomes tight for congested networks with large edge capacity.

IV. Numerical Experiments

In this section we explore the performance of the randomized routing procedure described in Algorithm 3 on a real-world road network with realistic customer demands. We consider the road network shown in Figure 3(a), a model of Manhattan with 1005 roads and 357 intersections. Customer requests are sampled from actual taxi rides in New York City on March 1, 2012 from 6 to 8 p.m[2]. We consider twenty realizations of the customer requests: in each case, we randomly select approximately 30% of the taxi trips (on average 14531 trips) and 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 for all realizations, we relax the congestion constraints by introducing slack variables, each associated with a large cost. We then solve the i-CRRP with Algorithm 3 and compare the overall travel time of the solution (computed with the BPR link delay function) with the travel time of the linear relaxation, which is a lower bound on the travel time of the optimal uncongested solution.

Table I summarizes our results. Figure 3 shows the distribution of the ratio between the overall travel time of the sampled solution and the overall travel time of the LP solution.

TABLE I
Results of the numerical simulations

	LP	Rand. routing
Avg. overall travel time	54342014	54339669
Avg. number of congested edges	7.2	50
Avg. congestion on cong. edges	261.4	380.1

Interestingly, the overall travel time of the sampled solution is sometimes smaller than the travel time of the LP relaxation; indeed, on average, the sampled solution's travel time is 99.996% of the travel time of the LP relaxation. This counterintuitive result is due to the fact that the LP relaxation computes a congestion-free solution, even if this

[2]courtesy of the New York Taxi and Limousine Commission

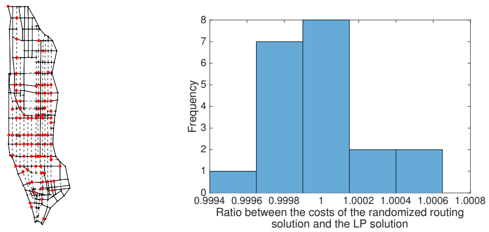

Fig. 3. Left: model of the Manhattan road network. Right: Distribution of the ratio between the overall travel time of the randomized routing solution and the overall travel time of the LP. In some cases, the randomized routing solution has a lower overall travel time than the LP, since it selects shorter (but congested) paths

results in longer paths for the customers and the rebalancing vehicles. 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 standard deviation of the ratio between the cost of the randomized routing solution and the cost of the LP solution is 0.027%: the randomized routing algorithm tracks the cost of the LP solution very closely.

V. Conclusions and Future Work

In this paper we presented a randomized algorithm for solving simultaneously customer routes and rebalancing routes on a capacitated road network in an autonomous mobility-on-demand system. Our goal was to find a set of routes that minimize the overall travel time: since the problem is intractable for generic congestion models, we adopted a simple threshold model for the synthesis algorithm that yields congestion-free routes. We formulated the routing and rebalancing problem as an integral Minimum-Cost Multi-Commodity Flow problem and developed a sampling technique that extends known randomized routing algorithms to flows with multiple sources and sinks. The sampling technique may violate some of the congestion constraints: we argued that, even if congestion constraints are violated, the expected total travel time of 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 showed that the total travel time computed by the randomized algorithm was within 0.027% of the optimal congestion-free travel time.

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. Stochastic fluctuations in the demand, travel times, and routing distribution can be taken into account by repeatedly running the algorithm in a receding-horizon fashion. We would also like to apply high fidelity congestion models (such as microscopic simulation models) to further validate the performance of the routing algorithm. Finally, we would like to study other ways of reducing congestion such as staggering demands, ride-sharing, and integration with public transportation.

References

[1] 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.

[2] 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.

[3] S. L. Smith, M. Pavone, M. Schwager, E. Frazzoli, and D. Rus, "Rebalancing the Rebalancers: Optimally Routing Vehicles and Drivers in Mobility-On-Demand Systems," in *American Control Conference*, Washington, DC, Jun. 2013, pp. 2362–2367.

[4] R. Zhang and M. Pavone, "A Queueing Network Approach to the Analysis and Control of Mobility-On-Demand Systems," in *American Control Conference*, Chicago, IL, Jul. 2015, pp. 4702–4709.

[5] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Load Balancing for Mobility-On-Demand Systems," in *Robotics: Science and Systems*, Los Angeles, CA, Jun. 2011, pp. 1–8.

[6] Rick Zhang and Marco Pavone, "Control of Robotic Mobility-On-Demand Systems: A Queueing-Theoretical Perspective," *International Journal of Robotics Research*, 2015, in press.

[7] 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, Sep. 2013.

[8] ——, "Models and Efficient Algorithms for Pickup and Delivery Problems on Roadmaps," in *Proc. IEEE Conf. on Decision and Control*, Maui, HI, Dec. 2012, pp. 5691–5698.

[9] B. Templeton, "Traffic Congestion & Capacity," 2015, available at http://www.templetons.com/brad/robocars/congestion.html.

[10] 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.

[11] F. Rossi, R. Zhang, and M. Pavone, "Autonomous vehicle routing in congested road networks," Mar. 2016, available at http://arxiv.org/abs/1603.00939.

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

[13] M. J. Lighthill and G. B. Whitham, "On kinematic waves. I. Flood movement in long rivers," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 229, no. 1178. The Royal Society, 1955, pp. 281–316.

[14] M. Treiber, A. Hennecke, and D. Helbing, "Microscopic simulation of congested traffic," in *Traffic and Granular Flow 99*. Springer, 2000, pp. 365–376.

[15] A. Srinivasan, "A survey of the role of multicommodity flow and randomization in network design and routing," *American Mathematical Society, Series in Discrete Mathematics and Theoretical Computer Science*, vol. 43, pp. 271–302, 1999.

[16] Lester Randolph Ford and Delbert Ray Fulkerson, *Flows in Networks*. Princeton University Press, 1962.

[17] R. M. Karp, "On the computational complexity of combinatorial problems," in *Networks, Networks (USA), (Proceedings of the Symposium on Large-Scale Networks, Evanston, IL, USA, 18-19 April 1974.)*, vol. 5, no. 1, Jan. 1975, pp. 45–68.

[18] H. Okamura, "Multicommodity flows in graphs," *Discrete Applied Mathematics*, vol. 6, no. 1, pp. 55 – 62, 1983.

[19] N. Garg, V. V. Vazirani, and M. Yannakakis, "Primal-dual approximation algorithms for integral flow and multicut in trees," *Algorithmica*, vol. 18, no. 1, pp. 3–20, 1997.

[20] 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.

[21] A. V. Goldberg, J. D. Oldham, S. Plotkin, and C. Stein, "An Implementation of a Combinatorial Approximation Algorithm for Minimum-Cost Multicommodity Flow," in *Integer Programming and Combinatorial Optimization*, ser. Lecture Notes in Computer Science, R. Bixby, E. Boyd, and R. Ros-Mercado, Eds. Springer Berlin Heidelberg, 1998, vol. 1412, pp. 338–352.

[22] F. Rossi, R. Zhang, and M. Pavone, "Congestion-aware randomized routing in autonomous mobility-on-demand systems (extended version)," Mar. 2016, available at http://web.stanford.edu/~pavone/papers/Rossi.Zhang.Pavone.CDC16.pdf.

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

[24] Bureau of Public Roads, "Traffic Assignment Manual," U.S. Department of Commerce, Urban Planning Division, Washington, D.C (1964), Tech. Rep., 1964.

APPENDIX: PROOFS OF TECHNICAL RESULTS

Proof: [Proof of Lemma 3.1] The proof is constructive and consists of six parts. First, we build a bipartite graph G_B that essentially captures the flow decomposition of the rebalancing flow. Second, we remove from the bipartite graph the two nodes incident on the edge of smallest weight (this is akin to removing the nodes belonging to the path flow of smallest intensity from the path decomposition). Third, we show that a suitable network flow can be found in the reduced bipartite graph. Fourth, we show that this network flow is a collection of path flows. Fifth, we construct a collection of path flows in the original bipartite graph G_B. Sixth, we map this collection of path flows to the original graph G.

A bipartite graph representation of the flow decomposition.: We consider a bipartite capacitated graph $G_B(\mathcal{V}_B, \mathcal{E}_B)$, with $\mathcal{V}_B = \{\mathcal{O}_B \cup \mathcal{D}_B\}$ where nodes $v \in \mathcal{O}_B$ correspond to rebalancing origins in the i-CRRP and nodes $v \in \mathcal{D}_B$ correspond to rebalancing destinations. Two nodes $s \in \mathcal{O}_B, t \in \mathcal{D}_B$ are connected by an edge $(u, v) \in \mathcal{E}_B$ only if the flow decomposition of $\{f_R(u, v)\}_{(u,v)}$ contains a path from s to t. Each edge $(s, t) \in \mathcal{E}_B$ is assigned a capacity $c_B(s, t)$ and a flow $f_B(s, t)$, both equal to the intensity of the path flow from s to t in G. The graph G_B is a simplified representation of the flow decomposition of $\{f_R(u, v)\}_{(u,v)}$: it encodes the origin, destination and intensity of each path flow but not the actual path. Edges only connect origin nodes to destination nodes: thus G_B is bipartite.

Removing the smallest path flow: Let us now consider the edge (s_m, t_m) of minimum capacity in G_B, corresponding to the smallest path flow in $\mathcal{F}$. We define the flow crossing the edge as $\sigma = c(s_m, t_m) = f_B(s_m, t_m)$. We remove nodes s_m and t_m and all edges incident on either s_m or t_m and we call the resulting graph $\tilde{G}_B(\tilde{\mathcal{V}}_B, \tilde{\mathcal{E}}_B)$, with $\tilde{\mathcal{V}}_B = \{\tilde{\mathcal{O}}_B \cup \tilde{\mathcal{D}}_B\}$, $\tilde{\mathcal{O}}_B = \{\mathcal{O}_B \setminus s_m\}$, $\tilde{\mathcal{D}}_B = \{\mathcal{D}_B \setminus t_m\}$.

The overall flow exiting node s_m is $\sum_{v \in \mathcal{D}_B} f_B(s_m, t) = \phi$. Also, $f_B(s_m, t_m) = \sigma$. Therefore $\sum_{t \in \tilde{\mathcal{D}}_B} f_B(s_m, t) = \phi - \sigma$ and, for any edge (s_m, t) with $t \in \tilde{\mathcal{D}}_B$, $f_B(s_m, t) \leq \phi - \sigma$. An identical reasoning shows that, for any edge (s, t_m) with $s \in \tilde{\mathcal{O}}_B$, $f_B(s, t_m) \leq \phi - \sigma$.

Finding a network flow: We now wish to show that there exists a network flow $\{\tilde{f}_B^{rs}(s, t)\}_{(s,t)}$ in $\tilde{G}_B$ where each node in $\tilde{\mathcal{O}}_B$ is an origin with intensity σ and each node in $\tilde{\mathcal{D}}_B$ is a destination with intensity σ and that satisfies the capacity constraints.

To this end, we prove the existence of an auxiliary network flow that acts as an upper bound on $\{\tilde{f}_B^{rs}(s, t)\}_{(s,t)}$. We assign intensity $\phi - f_B(s, t_m)$ to every origin node $u \in \tilde{\mathcal{O}}_B$. Note that $\phi - f_B(s, t_m) \geq \phi - (\phi - \sigma) \geq \sigma$. We also assign intensity $\phi - f_R(s_m, t) \geq \sigma$ to every destination node $v \in \tilde{\mathcal{D}}$.

Consider the network flow $\tilde{f}_B(s, t) = f_B(s, t) \, \forall (s, t) \in \tilde{\mathcal{E}}$ (i.e. the reduction of $\{f_B(s, t)\}_{(s,t)}$ to the edges contained in $\tilde{\mathcal{E}}_B$). It is easy to verify that $\tilde{f}_B(s, t)$ is a valid network flow in $\tilde{G}_B$ for the origin and destination intensities we assigned above. Also, by construction, capacity constraints are satisfied.

The origins and destinations of $\{\tilde{f}_B(s, t)\}$ all have intensity no smaller than σ, and the network flow satisfies the capacity constraints: then a network flow $\{\tilde{f}_B^{rs}(s, t)\}_{(s,t)}$ with origin and destination intensities σ that satisfies the congestion constraints also exists.

The network flow is a collection of path flows: Next, we show that the there exists a network flow $\{\tilde{f}_B^{rs}(s, t)\}_{(s,t)}$ that is a collection of path flows. The flow $\{\tilde{f}_B^{rs}(s, t)\}_{(s,t)}$ is generated by origins and destinations of intensity σ, and every edge $(s, t) \in \tilde{\mathcal{E}}$ connects an origin to a destination. Therefore, for every edge $(s, t) \in \tilde{\mathcal{E}}$, $\tilde{f}_B^{rs}(s, t) \leq \sigma$. Then, modifying the capacity of each edge $(s, t) \in \tilde{\mathcal{E}}$ so that $c_B(s, t) = \sigma$ does not affect the feasibility of $\{\tilde{f}_B^{rs}(s, t)\}_{(s,t)}$.

The problem of finding a network flow when all origins and destinations have unit intensity and all edges have unit capacity is known to be totally unimodular [**?**]: the LP relaxation of the problem admits a solution where all flows are integral. Then, the problem

$$
\begin{aligned}
\text{Find} \quad & \tilde{f}_B^{rs}(s, t) \;\; \forall (s, t) \in \tilde{\mathcal{E}} && (20)\\
\text{subject to} \quad & \sum_{s \in \tilde{\mathcal{V}}_B} \tilde{f}_B^{rs}(s, t) + \sum_{o \in \tilde{\mathcal{O}}_B} 1_{t=o}\sigma \\
& = \sum_{w \in \tilde{\mathcal{V}}_B} \tilde{f}_B^{rs}(t, w) + \sum_{d \in \tilde{\mathcal{D}}_B} 1_{t=d}\sigma \quad \forall t \in \tilde{\mathcal{V}}_B && (21)\\
& \tilde{f}_B^{rs}(s, t) \leq c_B(s, t) \;\; \forall (s, t) \in \tilde{\mathcal{E}}_B && (22)\\
& \tilde{f}_B^{rs}(s, t) \in \mathbb{R}_{\geq 0} \;\; \forall (s, t) \in \tilde{\mathcal{E}}_B && (23)
\end{aligned}
$$

has at least one solution where flows $\tilde{f}_B^{rs}(s, t)$ are either σ or 0 on every edge. That is, there exists a $\tilde{f}_B^{rs}(s, t)$ that is a collection of path flows of intensity σ, each connecting a node in $\tilde{\mathcal{O}}_B$ to a node in $\tilde{\mathcal{D}}_B$.

Remark The collection of path flows in $\tilde{f}_B^{rs}(s, t)$ induces a perfect matching on $\tilde{G}$: each node $o \in \tilde{\mathcal{O}}$ is the origin of a path flow and each node $d \in \tilde{D}$ is the destination of a path flow. Thus, $\tilde{f}_B^{rs}(s, t)$ can be found by computing a perfect matching in $\tilde{G}$.

A rebalancing solution in G_B: We are now in a position to build a rebalancing solution in G_B. Consider the flow $f_B^{rs}(s, t)$ in G_B with

$$
f_B^{rs}(s, t) = \begin{cases} \tilde{f}_B^{rs}(s, t) & \text{if } (s, t) \in \tilde{\mathcal{E}}_B \\ \sigma & \text{if } s = s_m, t = t_m \\ 0 & \text{otherwise} \end{cases}
$$

The flow is a rebalancing solution: (i) each edge $(s, t) \in \mathcal{E}_B$ with $f_B^{rs}(s, t) > 0$ is a path flow of intensity σ connecting an origin to a destination and (ii) each origin and each destination has an incident path flow.

A rebalancing solution in G: We can now map the rebalancing solution $\{f_B^{rs}(s, t)\}_{(s,t)}$ to a rebalancing solution $\{f_{rs}(u, v)\}_{u,v}$ in the original graph $G(\mathcal{V}, \mathcal{E})$. We start with the null network flow $f(u, v) = 0 \forall (u, v) \in \mathcal{E}$. Then, for every edge $(s, t) \in \mathcal{E}_B$, we consider the corresponding path flow in $G(\mathcal{V}, \mathcal{E})$. If $f_B^{rs}(s, t) = \sigma$, we add flow σ to all edges belonging to the corresponding path flow in G; otherwise, we add no flow to the edges belonging to the corresponding path flow.

By construction, the resulting flow $\{f_{rs}(u, v)\}_{u,v}$ is a collection of path flows of intensity σ. Furthermore, every origin and every destination in G has an incident path flow, since every node in $\mathcal{O}_B$ and $\mathcal{D}_B$ (corresponding to the origin and destination nodes in G respectively) has one incident edge with flow σ in $\{f_B^{rs}(s, t)\}_{(s,t)}$. Therefore $\{f_{rs}(u, v)\}_{u,v}$

is a fractional rebalancing solution.

For every edge, $f_{rs}(u,v) \le f_R(u,v)$. The flow $f_R(u,v)$ is a collection of path flows of intensity no smaller than σ: in $f_{rs}(u,v)$, the same path flows have intensity σ or 0.

Finally, the path flow $p_{s_m t_m} \in \mathcal{F}$ connecting s_m and t_m has intensity σ both in $f_{rs}(u,v)$ and in $f_R(u,v)$. By construction, for any edge $(u,v) \in p_{s_m t_m}$, $f_{rs}(u,v) \ge \sigma$. Therefore the rebalancing solution contains the path flow $p_{s_m t_m}$.

This concludes our proof. ∎

Proof: [Proof of Lemma 3.2] Correctness of the routine `RebDecomposition` is proved in Lemma 3.1. The paper [16] proves that the path decomposition of any network flow on a graph $G(\mathcal{V},\mathcal{E})$ contains at most $|\mathcal{E}|$ paths: therefore, $|\mathcal{P}| \le \mathcal{E}$. Every fractional rebalancing solution found by `RebDecomposition` contains one path $p \in \mathcal{P}$: thus, after at most $|\mathcal{E}|$ iterations, `RebDecomposition` returns an empty flow. Therefore Algorithm 1 terminates in at most $|\mathcal{E}|$ iterations. ∎

Proof: [Proof of Theorem 3.3] Any linear program can be solved in polynomial time [**?**]. In particular, the linear relaxation of the i-CRRP can be solved efficiently [21]. The runtime of the flow decomposition algorithm is $O(|\mathcal{E}|)$ [16]. The runtime of Algorithm 1 is also $O(|\mathcal{E}|)$, by Lemma 3.2. Finally, the sampling procedure in Algorithm 2 can be carried out in linear time in the number of passenger paths and fractional rebalancing solution. ∎

Proof: [Proof of Lemma 3.5] For ease of notation, we omit the indication of the edge (u,v) in functions $f_m(u,v)$, $f_R(u,v)$, $c(u,v)$, $\tilde{t}(u,v,f)$ and $U_{ott}(u,v,c(u,v))$. The expected value of X is $\mathbb{E}[X] = \sum_m f_m^s + f_R^s \le c$. Since paths are sampled independently, the following Chernoff bound holds:

$$\mathbb{P}\left(X \le (1+\delta)\mathbb{E}[X]\right) \ge 1 - \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}[X]}$$

The bound can be rewritten in a more familiar form as a bound on the cumulative distribution function (CDF) of X. Define $\delta(x) = x/\mathbb{E}[X] - 1$. Then we have that

$$\mathbb{P}\left(X \le x\right) \ge 1 - \left(\frac{e^{\delta(x)}}{(1+\delta(x))^{(1+\delta(x))}}\right)^{\mathbb{E}[X]}$$

Since the overall travel time $ott(x) = \tilde{t}(x)x$ is strictly monotonic in x by assumption (we recall that $\tilde{t}(x)$ depends on the link capacity c), we have (i) the mapping $x \mapsto ott(x) : \mathbb{R}_{\ge 0} \mapsto \mathbb{R}_{\ge 0}$ is bijective and admits an inverse, which we denote as $x(ott)$, and (ii)

$$X \le x \quad \Leftrightarrow \quad ott(X) \le ott(x).$$

Then the following bound on the CDF of the overall travel times holds:

$$\mathbb{P}\left(ott(X) \le ott(x)\right) \ge 1 - \left(\frac{e^{\delta(x)}}{(1+\delta(x))^{(1+\delta(x))}}\right)^{\mathbb{E}[X]}. \tag{24}$$

The distribution of X induces a distribution on the travel times $ott(X)$ along a link. We denote the resulting random variable as TT. The overall expected travel time on an edge can be expressed as a function of the cumulative distribution function $CDF_{TT}(t)$:

$$\mathbb{E}[TT] = \int_{t=0}^{\infty} \left(1 - CDF_{TT}(t)\right) dt \tag{25}$$

The CDF of TT admits a lower bound:

$$\mathbb{P}\left(TT \le ott\right) \ge 1 - \left(\frac{e^{\delta(x(ott))}}{(1+\delta(x(ott)))^{(1+\delta(x(ott)))}}\right)^{\mathbb{E}[X]} \tag{26}$$

Let us now define a random variable H whose CDF obeys

$$\mathbb{P}\left(H \le h\right) = 1 - \left(\frac{e^{\delta(x(h))}}{(1+\delta(x(h)))^{(1+\delta(x(h)))}}\right)^{\mathbb{E}[X]}$$

The CDF of H is a lower bound on the CDF of TT by Equation 26. Therefore, the expected value of H is larger than the expected value of TT, according to Equation 25.

We denote $U_{ott} = \mathbb{E}[H]$. The distribution of H only depends on $\mathbb{E}[X]$ and $x(ott)$, which is fully determined by $\tilde{t}(f)$. In addition, $U_{ott} = \mathbb{E}[H] \ge \mathbb{E}[TT]$. This concludes our proof. ∎

Algorithm 4 Flow decomposition algorithm [16].

Input: A network flow $\{f_m(u,v)\}_{(u,v)}$
  A set $\mathcal{O}$ of origin nodes $s \in \mathcal{O}$
  A set $\mathcal{D}$ of destination nodes $t \in \mathcal{D}$
Output: A list of paths $\{Path_i\}_i$. Each entry $Path_i$ is a collection of consecutive edges $\{(s,u),(u,v),\dots(w,t)\}$.
  A list of path intensities $\{\lambda_i\}_i$
procedure FlowDecomposition($\{f_m(u,v)\}_{(u,v)}$)
  i=1;
  while $\sum_{s\in\mathcal{O}}\sum_{v\in V} f_m(s,v) > 0$ **do**
    $Path_i \leftarrow$ a path from some $s \in \mathcal{O}$ to some $t \in \mathcal{D}$ containing only edges (u,v) with $f_m(u,v) > 0$
    $\lambda_i = \min_{(u,v)\in Path_i} f_m(u,v)$
    for all $(u,v) \in Path_i$ **do**
      $f_m(u,v) = f_m(u,v) - \lambda_i$
    $i = i + 1$
  return the path flows $\{\{Path_i\},\{\lambda_i\}\}_i$