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 (that is, a road network where traffic speed depends on vehicle density). We show that the 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 that approximately minimizes the number of vehicles on the road in polynomial time. We provide theoretical bounds on the probability of violating the congestion constraints; we also characterize the expected number of vehicles required by the solution with a commonly-used empirical congestion model and provide a bound on the approximation factor of the algorithm. 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 and required number of vehicles 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 has 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. 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 a framework to efficiently 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 [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 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 [6] or on a road network [7] 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 [8], [9], but recent work suggests that with intelligent routing, this need not be the case [10].

The routing and rebalancing problem on a congested network was studied in [10] 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 [10] to accommodate integral flows.

In the field of transportation science, our problem is similar to dynamic traffic assignment (DTA) problems [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 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 [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. Hence, we leverage simple traffic models that are amenable to tractable analysis and control.

Statement of contributions: Our ultimate goal is to efficiently find optimal vehicle routes that service passenger requests and minimize the overall number of vehicles on the road (and therefore the operating cost) 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 (i.e. a threshold model where each road has a maximum capacity) to compute congestion-free vehicle routes as a proxy for solving the general problem of minimizing travel times. We develop a scalable polynomial time randomized algorithm based on the congestion-free solution and evaluate its performance in terms of overall travel times using a more accurate 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 violation. The technique extends well-known results in combinatorial optimization [14] to the case where the flows have multiple origins and destinations and is of independent interest. Second, we provide a semi-analytical characterization of the expected travel time (a proxy for the number of vehicles on the road) produced by the randomized technique under a widely adopted empirical congestion model. We also provide an upper bound on the the approximation factor of the algorithm, i.e. the ratio between the number of vehicles produced by our algorithm and the optimal congestion-free number of vehicles. 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 (and the same number of vehicles on the road as) 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-A we describe a randomized algorithm to solve the routing and rebalancing problem and provide guaranteed bounds on the approximation factor of the cost of the solution provided by the algorithm (in terms of required vehicles) and on the approximation factor with respect to the optimal congestion-free cost. 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 AND PROBLEM FORMULATION

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 [10], 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

Two congestion models are adopted in this paper. A simpler synthesis model is used in control synthesis in Sections III-A and III-B whereas a more general analysis model that offers a better representation of congested traffic behavior is used to (a) analytically characterize the performance of the routing and rebalancing algorithm in Section III-C and (b) simulate the algorithm’s behavior with real customer demands in Section IV.

Both models are consistent with classical traffic flow theory [15], [12]. 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) [15]. The flow rate (i.e. the number of vehicles traversing a link per unit time) grows with vehicle density up to a critical value (referred to in the literature as the capacity of the road link), at which point vehicle speeds and flow rate decrease significantly, signaling the onset of congestion. The capacity of the road link 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 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 capacity constraints on every link is guaranteed to be congestion-free.

The 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 a speed close to 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.

B. Network flow model of AMoD system

We model the road network as a capacitated graph \( G(V,E) \). The nodes \( v \in V \) represent intersections and locations of customer trip origins and destinations. The edges \((u,v) \in E\) represent road links. Congestion is modeled as a constraint on the flow of a vehicle on a road link that can accommodate, in accordance with the analysis congestion model: a function \( c(u,v) : E \rightarrow \mathbb{N}_{\geq 0} \) 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 \( t(u,v) : E \rightarrow \mathbb{R}_{>0} \).

Customer requests are denoted by the tuple \((s,t,\lambda)\), where \( s \in V \) is the origin of the request, \( t \in V \) 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: \( \lambda \) is constant in time. The set of transportation requests is denoted as \( M = \{(s_m,t_m,\lambda_m)\}_{m} \) with \( m \in \{1,\ldots,|M|\} \). In this paper, we restrict our analysis to the case where each customer request has unit intensity: a customer request of intensity \( \lambda \) is modeled as \( \lambda \) customer requests of unit intensity between the same origin and destination nodes.

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 \( s_i \) (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 \( f(u,v) \) 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 \( \lambda \), a network flow is a function \( f(u,v) : \mathcal{E} \rightarrow \mathbb{R}_{\geq 0} \) that obeys the following equation:

\[
\sum_{u \in \mathcal{V}} f(u,v) + \sum_{i} 1_{v = s_i} \lambda_i = \sum_{u \in \mathcal{V}} f(v,u) + \sum_{j} 1_{v = t_j} \lambda_j \quad \forall v \in \mathcal{V} \quad (1)
\]

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

\[
\sum_{u \in \mathcal{V}} f(u,v) + \sum_{i} 1_{v = s_i} \lambda_i = \sum_{u \in \mathcal{V}} f(v,u) + \sum_{j} 1_{v = t_j} \lambda_j \quad \forall v \in \mathcal{V} \quad (2)
\]

Note that Equation (2) 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 \( \lambda \), we define a path flow as a function \( f^p(u,v) : \mathcal{E} \rightarrow \mathbb{R}_{\geq 0} \) that satisfies Equation (1) and only assigns positive flow to edges belonging to a path \( p \) going from \( s \) to \( t \): \( p = \{(s,u),(u,v), \ldots, (w,t)\} : f^p(u,v) > 0 \iff (u,v) \in p \). 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 decomposition may also contain cycles. In Section III-A we show that this is not the case for the network flows considered in this paper). 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)

We are now in a position to define the integral Congestion-free Routing and Rebalancing problem.

Integral Congestion-free Routing and Rebalancing problem (i-CRRP): Compute a set of routes that

(i) transfer customers to their desired destinations (customer-carrying trips),
(ii) rebalance vehicles throughout the network to realign the vehicle fleet with the customers’ demands (customer-empty, or rebalancing, trips),
(iii) do not cause congestion on any road link and
(iv) minimize the overall number of vehicles on the road.

The i-CRRP can be cast as a mixed-integer linear program. 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 \( \{f_R(u,v)\}_{(u,v)} \).

The number of vehicles needed to implement these flows can be expressed as [17]

\[
V_{\text{min}} = \left[ \sum_{m \in \mathcal{M}} \sum_{(u,v) \in \mathcal{E}} t(u,v) (f_m(u,v) + f_R(u,v)) \right].
\]

Given a capacitated network \( G(V,E) \) and a set of transportation requests \( \mathcal{M} = \{ (s_m,t_m,\lambda_m) \} \), we solve

\[
\begin{align*}
\text{minimize} & \quad \sum_{m \in \mathcal{M}} \sum_{(u,v) \in \mathcal{E}} t(u,v) f_m(u,v) + \sum_{(u,v) \in \mathcal{E}} t(u,v) f_R(u,v) \\
\text{subject to} & \quad \sum_{u \in \mathcal{V}} f_m(u,v) + 1_{v = s_m} \lambda_m \\
& \quad = \sum_{u \in \mathcal{V}} f_m(v,u) + 1_{v = t_m} \lambda_m \quad \forall m \in \mathcal{M}, v \in \mathcal{V} \quad (3b) \\
& \quad \sum_{u \in \mathcal{V}} f_R(u,v) + \sum_{m \in \mathcal{M}} 1_{v = s_m} \lambda_m \\
& \quad = \sum_{u \in \mathcal{V}} f_R(v,u) + \sum_{m \in \mathcal{M}} 1_{v = t_m} \lambda_m \quad \forall v \in \mathcal{V} \quad (3c) \\
& \quad f_R(u,v) + \sum_{m \in \mathcal{M}} f_m(u,v) \leq c(u,v) \quad \forall (u,v) \in \mathcal{E} \quad (3d) \\
& \quad f_m(u,v) \in \mathbb{N}_{\geq 0} \quad \forall m \in \mathcal{M}, (u,v) \in \mathcal{E} \quad (3e) \\
& \quad f_R(u,v) \in \mathbb{N}_{\geq 0} \quad \forall (u,v) \in \mathcal{E} \quad (3f)
\end{align*}
\]

Equation (3a) captures the goal of minimizing the number of vehicles on the road. We remark that this goal is aligned with the goal of minimizing overall travel times for customer-carrying and rebalancing routes. Equations (3b) and (3c) ensure that each customer-carrying flow \( \{f_m(u,v)\}_{(u,v),m} \) is an integral network flow. Equations (3d) and (3f) ensure that the rebalancing flow \( \{f_R(u,v)\}_{(u,v)} \) is an integral network flow. Finally, Equation (3e) enforces the capacity constraint of every link.

D. Complexity of the i-CRRP

Theorem 2.1 (Complexity of the i-CRRP): The decision version of the i-CRRP is NP-complete.

Proof: We show that any instance of SAT can be reduced to an instance of two-customer i-CRRP (the extension to \( k \geq 3 \) customers is trivial).

The i-CRRP is an instance of the integral Minimum-Cost Multi-Commodity Flow (Min-MCF) problem. However, in the i-CRRP, one of the commodities is always a rebalancing commodity, i.e. a commodity whose origin nodes and corresponding intensities coincide with the set of destination
nodes and intensities of the other commodities, and vice versa.

The decision version of the integral Min-MCF is known to be NP-complete [18], [19] in the general case. Theorem 3 in [19] can be modified to also hold for the special case of MCF problems with a rebalancing commodity. In [19], the authors show that any instance of $k$-SAT (with $k \geq 3$) can be cast as an instance of integral two-commodity flow: the first commodity (with intensity 1) ensures that every variable is either true or false but not both, whereas the second commodity (with intensity $k$) models clause satisfaction. Note that, in the construction in [19], no directed path exists from the destination of commodities 1 and 2 to their origins. We modify the construction by introducing a directed edge with capacity 1 from the destination of commodity 1 to its origin and a directed edge with capacity $k$ from the destination of commodity 2 to its origin. In order for a rebalancing flow to exist, both edges must be saturated; the rest of the graph is identical to the graph in [19]. Thus, any instance of SAT can be reduced to an instance of (the decision version of) two-customer i-CRRP. It follows that the decision version of i-CRRP is NP-hard.

Theorem 2.1 shows that the i-CRRP can not be solved efficiently for large numbers of passengers or realistic road networks unless $P=NP$. Furthermore, while polynomial-time approximation schemes are known for variants of the integral multi-commodity flow problem such as Max-MCF [20], to the best of our knowledge no such approximation schemes exist for Min-MCF.

However, in classical traffic flow theory, small violations of the congestion constrains produce proportionally small decreases in traffic speed [15]. Thus, we turn our attention to efficiently solving an approximate version of the i-CRRP that admits small violations of the congestion constraints [3d].

### III. A Randomized Routing Algorithm

Theorem 2.1 shows that the i-CRRP can not be solved efficiently for large numbers of passengers or realistic road networks unless $P=NP$. Furthermore, while polynomial-time approximation schemes are known for variants of the integral multi-commodity flow problem such as Max-MCF [20], to the best of our knowledge no such approximation schemes exist for Min-MCF.

However, in classical traffic flow theory, small violations of the congestion constrains produce proportionally small decreases in traffic speed [15]. Thus, we turn our attention to efficiently solving an approximate version of the i-CRRP that admits small violations of the congestion constraints [3d].

#### Approximate Integral Congestion-free Routing and Rebalancing problem:

Compute a set of routes that

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

(ii) rebalance vehicles throughout the network to realign the vehicle fleet with the customers’ demands (customer-empty, or rebalancing, trips),

(iii) violate the congestion constraints on all road links by at most a small value $\delta$ with high probability,

(iv) has bounded suboptimality (in terms of required number of vehicles) under the analysis congestion model.

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 small probability and have optimal expected cost [21], [14]. However, in the i-CRRP, the rebalancing flow has multiple origins and destinations: naive randomized routing cannot guarantee that the integral rebalancing flow will satisfy continuity constraints (Equation 3c) 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 [14] to the case where flows have multiple origins and destinations and can be applied to general Min-MCF problems where small violations of the capacity constraints are acceptable.

The procedure, described in detail in the next two sections, can be summarized as follows: (i) first, a solution to the LP relaxation of the i-CRRP is computed, (ii) then, the solution is decomposed into fractional customer routes and fractional rebalancing solutions (Section III-A) and (iii) finally, the fractional customer routes and rebalancing solutions are sampled 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.

#### A. Linear relaxation of the CRRP and flow decomposition

The first step in the randomized routing technique is to solve a linear relaxation of the i-CRRP, denoted as the fractional CRRP. To obtain the fractional CRRP, we replace Equations 3c and 3f with

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

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

(4a)

(4b)

The resulting fractional CRRP is an instance of the fractional Min-MCF problem, which can be efficiently solved directly as a linear program (of size $|\mathcal{E}|(|\mathcal{M}| + 1)$) or via specialized combinatorial algorithms, e.g. [22].

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. The flow decomposition algorithm (reported in [16, Sec. 3.5] and also in the Appendix as Algorithm 4) decomposes a network flow in a collection of path flows and cycles in $O(|V||E|)$ [16]. The flow decomposition of a general network flow may contain cycles. However, the flow decomposition of a solution to the linear relaxation of Problem 3 contains no cycles due to the minimization objective: if a cycle was present, then removing it would result in a new solution with lower cost, contradicting (3a).

Computing a suitable decomposition of the rebalancing flow is more challenging. The 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 (the decomposition of the rebalancing flow contains no cycles, analogously to the passenger flows, due to the form of the objective function 3a). 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.
Intuitively, a fractional rebalancing solution \( r \) is a collection of paths that collectively rebalances every origin and every destination (that is, such that every customer origin has an inbound path and every customer destination has an outbound path), associated with an intensity \( \lambda_r \) (which may be interpreted as a probability). More rigorously, consider a collection \( \mathcal{P} \) of path flows \( \{f_p(u, v)\}_{u, v} \). Then the sum of these path flows is a fractional rebalancing solution with intensity \( \lambda_r \) is a fractional rebalancing solution with intensity \( \lambda_r \).

Lemma 3.1: Consider a graph \( G(V, E) \) 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 \( (u, v) \in E \), \( \sum_{p \in \mathcal{F}} f_p(u, v) = f_R(u, v) \). Then there exists a rebalancing solution \( f_{rs} \) of intensity \( \sigma \leq \phi \) such that (i) for every edge \( (u, v) \in E \), \( f_{rs}(u, v) \leq f_R(u, v) \) and (ii) \( f_{rs} \) contains at least one path flow in \( \mathcal{F} \), that is, \( \exists p \in \mathcal{F} : f_{rs}(u, v) \geq f_p(u, v) \forall (u, v) \in p \).

Proof sketch: The proof is constructive. First, a flow decomposition of \( \{f_R(u, v)\}_{u, v} \) is performed: the intensity of the smallest path flow \( p \) is denoted as \( \sigma \). The capacities of all edges \( (u, v) \in E \) are then reduced to coincide with \( \sigma \). Total unimodularity is then used to show that there exists a rebalancing flow where all link flows are either zero or multiples of \( \sigma \) (i.e., a fractional rebalancing solution of intensity \( \sigma \)) embedded in \( \{f_R(u, v)\}_{u, v} \) and that this solution contains the path flow \( p \).

A rigorous proof of Lemma 3.1 is reported in the Appendix.

The difference between a rebalancing flow with intensity \( \phi \) and a fractional rebalancing solution with intensity \( \sigma \) is also a rebalancing flow (with intensity \( \phi - \sigma \)). Thus, any rebalancing flow can be decomposed in rebalancing solutions. Lemma 3.2 formalizes this intuition.

Lemma 3.2: Any rebalancing flow \( \{f_R(u, v)\}_{u, v} \) can be decomposed in a collection of rebalancing solutions \( \{f_{rs_i}(u, v)\}_{i(u, v)} \) such that, for every edge \((u, v) \in E\), \( \sum_{i} f_{rs_i}(u, v) = f_R(u, v) \). Furthermore, the decomposition contains at most \(|E| \) fractional rebalancing solutions.

Proof: The lemma follows from recursive application of Lemma 3.1 and from the observation that the flow decomposition \( \mathcal{F} \) of \( \{f_R(u, v)\}_{u, v} \) contains at most \(|E| \) path flows.

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

Find \( f_{re}(u, v) \forall (u, v) \in \mathcal{E} \)

subject to

\[
\begin{align*}
\sum_{u \in V} f_{re}(u, v) + \sum_{o \in O} 1_{v=o} \sigma &= \sum_{w \in V} f_{re}(v, w) + \sum_{d \in D} 1_{d=v} \sigma \quad \forall v \in V \\
f_{re}(u, v) \leq f_R(u, v) \forall (u, v) \in \mathcal{E} \\
f_{re}(u, v) \in \mathbb{R}_{\geq 0} \forall (u, v) \in \mathcal{E}
\end{align*}
\]

Lemma 3.1 shows that Problem 5 is always feasible. The linear program 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 an optimal solution \( \{f_{re}(u, v)\}_{u, v} \) where all link flows are either zero or multiples of \( \sigma \). It is then easy to show that the resulting flow is a fractional rebalancing solution of weight \( \sigma \). We refer the reader to the proof of Lemma 3.1 in the Appendix for a more thorough discussion.

Problem 5 can then be solved recursively to decompose a rebalancing flow in a collection of fractional rebalancing solutions. This procedure is formalized in Algorithm 1.

Algorithm 1 Rebalancing flow decomposition in fractional rebalancing solutions.

\[
\begin{align*}
S &\leftarrow \text{REBDECOMPOSITION}(\{f_R(u, v)\}_{u, v}, \emptyset) \\
\text{procedure} \quad \text{REBDECOMPOSITION}(f, S) \\
&\quad f_{re} \leftarrow \text{an integral solution to Problem 5} \\
&\quad S \leftarrow \text{REBDECOMPOSITION}(f_{re} - f_{rs}, S \cup f_{re}) \\
&\quad \text{return } S
\end{align*}
\]

Remark: Lemma 3.2 shows that the rebalancing decomposition of any flow contains at most \(|M| \) fractional rebalancing solutions. Thus, Algorithm 1 performs at most \(|M| \) recursive calls.

B. Sampling passenger and rebalancing routes

The flow decomposition algorithm produces a flow decomposition of each customer flow \( \{f_m(u, v)\}_{u, v} \): for each customer \( m \in M \), 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 \( r \in S \).

We are now in a position to sample \( \mathcal{F}_m \) and \( S \) to produce a collection of integral customer and rebalancing flows. For each customer \( m \in M \), we independently sample exactly one path flow from \( \mathcal{F}_m \) with probability equal to the intensity of the path flow (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 \( \{f_m(u, v)\}_{u, v} \).

We also independently sample one fractional rebalancing solution from \( S \) 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 \( \{ f_R(u, v) \}_{u,v} \).

The procedure is summarized in Algorithm 2.

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

```
procedure SAMPLEPATHS(\( \mathcal{F}_m \), \( S \))
    for \( m \in \mathcal{M} \) do
        Sample one path flow \( \{ p_{m}, \lambda_{m} \} \) from \( \mathcal{F}_m \) w.p. \( \lambda_{m} \).
        Set the intensity of \( p_m \) to \( \lambda_m = 1 \).
        Sample one fractional rebalancing solution \( \{ f_R(u, v) \}_{u,v} \) with probability \( \lambda_r \).
        Set the intensity of the rebalancing solution to \( \lambda_r = 1 \).
    return \( \{ p_m \}, \{ f_R(u, v) \}_{(u,v)} \)
```

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

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

1: **Input:** An instance of the i-CRRP
2: **Output:** A route \( r_m \) for each customer \( m \).
3: A collection \( \{ r_{reb} \} \) of rebalancing routes.
4: \( \{ f_m \}_{(u,v)}, \{ f_R(u,v) \} \leftarrow \text{solve the linear relaxation of the i-CRRP} \)
5: for all \( m \in \mathcal{M} \) do
6: \( \{ f_m \} \leftarrow \text{FLOWDECOMPOSITION(} \{ f_m \} \) \)
7: \( S_R \leftarrow \text{REBDECOMPOSITION(} f_R, 0 \) \)
8: \( \{ r_m \}, \{ f_{R}(u,v) \} \leftarrow \text{SAMPLEPATHS(} \{ F_m \}, \{ S_R \} \) \)
9: \( r_{reb} \leftarrow \text{FLOWDECOMPOSITION(} \{ f_R(u,v) \} \) \)

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

**Proof:** Any linear program can be solved in polynomial time [23]. In particular, the linear relaxation of the i-CRRP on line 4 and Problem 5 in Algorithm 1 can be solved efficiently [22]. The runtime of the flow decomposition algorithm is \( O(|V| |E|) \) [16]: the algorithm is called \( |\mathcal{M}| \) times on lines 5 to 8 and once on line 9. The rebalancing decomposition procedure on line 7 performs at most \( |E| \) recursions, according to Lemma 3.2; each recursion solves an instance of Problem 5. Finally, the sampling procedure on line 8 is carried out in linear time in the number of passenger paths and fractional rebalancing solution.

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, \( \{ f^m(u, v) \}_{(u,v),m} \) and \( \{ f^R(u, v) \}_{(u,v)} \) only assume integer values. Therefore the flows satisfy Equations 4 and 5. The expected traffic flow crossing every edge \( (v, u) \in E \) is upper-bounded by the capacity of the edge \( c(u, v) \):

\[
\mathbb{E} \left[ \sum_{m} f^m(u, v) + f^R_R(u, v) \right] = \sum_{m} f^m(u, v) + f^R_R(u, v) \leq c(u, v)
\]

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 degree to which the output of Algorithm 3 may violate the capacity constraints 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 approximate i-CRRP such that no congestion constraint for edges \( (u, v) \in \mathcal{E} \) is violated by more than a multiplicative factor \( \delta_{u,v} = \sqrt{3/2} \alpha \log(|\mathcal{E}|/\alpha) \), if \( \delta_{u,v} \leq 1 \) for all \( (u, v) \).

**Proof:** The proof relies on a Chernoff bound for every congestion constraint and on Boole’s inequality.

**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, since it is the sum of independent Bernoulli trials, and its mean is upper-bounded by the capacity of the edge \( c(u, v) \). Therefore a Chernoff bound holds [24]:

\[
P(X(u, v) \geq (1 + \delta) c(u, v)) \leq \exp(-c(u, v)\delta^2/3) 
\]

if \( 0 < \delta \leq 1 \).

**Conegion on all edges:** Select \( \delta_{u,v} \) on every edge such that \( c(u, v)\delta_{u,v}^2 \) for every pair of edges \( (u, v), (w, t) \in \mathcal{E} \). Then, by Boole’s inequality, the probability that at least one edge violates the congestion constraint is upper-bounded by

\[
P(\exists(u, v) \in \mathcal{E} : X(u, v) \geq (1 + \delta_{u,v}) c(u, v)) \leq |\mathcal{E}| \exp(-c(u, v)\delta_{u,v}^2/3) 
\]

Let us call \( \alpha = |\mathcal{E}| \exp(-c(u, v)\delta_{u,v}^2/3) \). 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 (and therefore in the number of required vehicles); if the relationship between traffic flow and
traffic speed (i.e. the analysis congestion model) is known in the congested regime, this increase can be quantified.

Let us consider an analysis congestion model \( \hat{t}(u, v, f(u, v)) \) describing the travel time on link \((u, v) \in \mathcal{E}\) when the traffic flow traversing the link is \( f(u, v) = \sum_m f_m(u, v) + f_R(u, v) \). We assume that \( \hat{t}(u, v, f(u, v)) \) is nondecreasing in \( f(u, v) \); we place no further assumptions on its shape. One such congestion model is the widely used Bureau of Public Roads (BPR) link delay model [25], which models the travel time on a link as

\[
\hat{t}(u, v, f(u, v)) = t(u, v) \left( 1 + 0.15 \left( \frac{f(u, v)}{c(u, v)} \right)^4 \right)
\]

Under this description of travel times, the expected number of vehicles obtained by Algorithm \( \mathcal{A} \) is

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

with \( f^*(u, v) = \sum_m f_m^*(u, v) + f_R^*(u, v) \).

Analytical computation of the expected values in the expression above for general congested travel time functions \( \hat{t} \) 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 traffic flow (and therefore a level of congestion) 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 BPR link delay 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 and required number of vehicles 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.](image)

To formalize this intuition, we first provide two lemmas characterizing the distribution of travel times on a link, then in Theorem 5.7 we provide a bound on the ratio of expected number of required vehicles between Algorithm \( \mathcal{A} \) 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 traffic flow traversing the link: \( X = \sum_m f_m(u, v) + f_R(u, v) \). Then the expected required number of vehicles on link \((u, v)\) under the analysis congestion model \( \mathcal{A} \) (i.e. \( \mathbb{E} \left[ \hat{t}(u, v, X) \right] \)), admits an upper bound \( \mathcal{U}_v(u, v, \mathbb{E}[X]) \) 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.

Proof: For ease of notation, we omit the indication of the edge \((u, v)\) in functions \( f_m(u, v) \), \( f_R(u, v) \), \( c(u, v) \), \( \hat{t}(u, v, f) \) and \( \mathcal{U}_v(u, v, c(u, v)) \). The expected value of \( X \) is \( \mathbb{E}[X] = \sum_m f_m^* + f_R^* \leq c \). Since paths are sampled independently, the following Chernoff bound holds:

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

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

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

Since the required number of vehicles \( v(x) = \hat{t}(x) x \) is strictly monotonic in \( x \) by assumption (we recall that \( \hat{t}(x) \) depends on the link capacity \( c \)), (i) the mapping \( x \mapsto v(x) : \mathbb{R}_{>0} \mapsto \mathbb{R}_{>0} \) is bijective and admits an inverse, which we denote as \( x(v) \), and (ii) \( X \leq x \iff v(X) \leq v(x) \).

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

\[
\mathbb{P} (v(X) \leq v(x)) \geq 1 - \left( \frac{e^{\delta(x)}}{(1 + \delta(x))^{1+\delta(x)}} \right)^{\mathbb{E}[X]}
\]

The distribution of \( X \) induces a distribution on the required number of vehicles \( v(X) \) along a link. We denote the resulting random variable as \( \mathcal{V} \). The overall expected required number of vehicles on an edge can be expressed as a function of the cumulative distribution function \( CDF_{\mathcal{V}}(t) \):

\[
\mathbb{E} [\mathcal{V}] = \int_{t=0}^{\infty} (1 - CDF_{\mathcal{V}}(t)) \, dt
\]

The CDF of \( \mathcal{V} \) admits a lower bound:

\[
\mathbb{P} (\mathcal{V} \leq v) \geq 1 - \left( \frac{e^{\delta(x)}}{(1 + \delta(x))^{1+\delta(x)}} \right)^{\mathbb{E}[X]}
\]

Let us now define a random variable \( \mathcal{H} \) whose CDF obeys

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

The CDF of \( \mathcal{H} \) is a lower bound on the CDF of \( \mathcal{V} \) by Equation 8. Therefore, the expected value of \( \mathcal{H} \) is larger than the expected value of \( \mathcal{V} \), according to Equation 7.

We denote \( \mathcal{U}_v = \mathbb{E} [\mathcal{H}] \). The distribution of \( \mathcal{H} \) only depends on \( \mathbb{E} [X] \) and \( x(v) \), which is fully determined by \( \hat{t}(f) \). In addition, \( \mathcal{U}_v = \mathbb{E} [\mathcal{H}] \geq \mathbb{E} [\mathcal{V}] \). This concludes our proof.

Lemma 3.5 can be used to characterize the ratio between the travel time of the solution produced by Algorithm \( \mathcal{A} \) and the optimal travel time.

Lemma 3.6: (Expected approximation factor of the required number of vehicles on a link): Consider a link \((u, v) \in \mathcal{E}\). Call \( X \) the traffic flow traversing the link. Then, if \( \mathbb{E} [X] > 0 \), the ratio between \( \mathbb{E} \left[ \hat{t}(u, v, X) \right] \) and \( \mathbb{E} [X] \), admits an upper bound \( \mathcal{R}_v(u, v, \mathbb{E}[X]) \) 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 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 \( \mathcal{V}/(\mathbb{E}[X])/\mathbb{E}[X] \)
only differs from the random variable $V$ by a constant multiplicative factor.

The result in Lemma 3.6 can be used to numerically characterize the ratio between the expected overall travel time on a link and vehicle flow in the congested regime. 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 $R_v$ 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.](image)

The 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 the number of required vehicles is very well controlled: the expected required number of vehicles 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 required number of vehicles on a single link can be used to characterize the systemwide performance of our sampling technique. The following theorem formalizes this intuition.

**Theorem 3.7 (Expected required number of vehicles):**
Consider an instance of the i-CRRP and an analysis link delay function $l(u, v, f)$. 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 $I$ 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 $(u, v) \in E$, it is possible to compute an upper bound $R_u(u, v)$ on the ratio between the expected required number of vehicles in the integral solution $I$ under the analysis congestion function $l(u, v, f)$ and the required number of vehicles in the fractional solution. In addition, the ratio between the expected overall required number of vehicles in $I$ and the overall required number of vehicles of the fractional solution is upper-bounded by

$$\mathcal{U}_v = \frac{\sum_{(u,v) \in E} R_v(u,v)l(u,v,f(u,v))f(u,v)}{\sum_{(u,v) \in E} l(u,v,f(u,v))f(u,v)}$$

**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 $U_v$ on the overall travel time 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 $R_v(c)$, shown in Figure 2, is strongly decreasing in both 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. 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 required number of vehicles of the solution (computed with the BPR link delay function) with the required number of vehicles of the linear relaxation. We remark that the linear relaxation does not yield integral routes: thus, it is not suitable for real-time control of an AMoD system. However, the linear relaxation offers a lower bound on the travel time of the optimal integral uncongested solution.

Table I summarizes our results. Figure 3 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. overall travel time            | 34.52/14 | 34.52/14 |
| Avg. number of vehicles on the road | 7548 | 7548 |
| Avg. number of congested edges      | 7.2 | 50 |
| Avg. congestion on congested edges  | 261.4 | 380.1 |

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.

Interestingly, the overall travel time required by the sampled solution is sometimes smaller than the number of vehicles in the LP relaxation; indeed, on average, the sampled solution's cost is 99.996% of the cost of the LP relaxation.

1 courtesy of the New York Taxi and Limousine Commission
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 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 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 number of vehicles on the road: 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 sources and sinks. The sampling technique may violate some of the congestion constraints; nevertheless we prove that the expected total travel time of the randomized algorithm (and thus the total number of vehicles on the road) 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 number of vehicles on the road required by our algorithm matches the optimal number of vehicles required by a congestion-free solution.

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. Finally, 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, Washington, DC, Jun. 2013, pp. 2362–2367.

[3] 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.

[4] 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.

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

[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. 12, pp. 2261–2276, Sep. 2013.

[7] ——, “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.

[8] B. Templeton, “Traffic Congestion & Capacity,” 2015, available at http://www.templetons.com/road/robotcars/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,” in Transportation Research Part B: Methodological, vol. 25, no. 2, pp. 143–161, 1991.

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

[12] M. Treiber, A. Hennecke, and D. Helbing, “Microscopic simulation of congested traffic,” in Traffic and Granular Flow 99. Springer, 2000, pp. 356–375.

[13] 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.

[14] J. G. Wardrop, “Some Theoretical Aspects of Road Traffic Research,” in ICE Proceedings: Engineering Divisions, vol. 1, no. 3. Thomas Telford, 1952, pp. 325–362.

[15] Rajivendra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, Network Flows: Theory, Algorithms and Applications. Upper Saddle River, New Jersey 07458: Prentice Hall, 1993.

[16] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, Network Flows: Theory, Algorithms and Applications. Upper Saddle River, New Jersey 07458: Prentice Hall, 1993.

[17] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, “Robotic Load Balancing for Mobility-On-Demand Systems,” International Journal of Robotics Research, vol. 31, no. 7, pp. 839–854, Jun. 2012.

[18] 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.

[19] 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.

[20] N. Garg, V. V. Vazirani, and M. Yannakakis, “Primal-dual approximation algorithms for integral flow and multicut in trees,” Journal of Algorithms, vol. 18, no. 1, pp. 3–20, 1997.

[21] 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.

[22] 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.

[23] Narendra Karmarkar, “A New Polynomial-Time Algorithm for Linear Programming,” Combinatorica, vol. 4, no. 4, pp. 373–395, 1984.

[24] M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005.

[25] Bureau of Public Roads, “Traffic Assignment Manual,” U.S. Department of Commerce, Urban Planning Division, Washington, D.C (1964), Tech. Rep., 1964.
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 $G_B(V_B, E_B)$, with $V_B = \{O_B \cup D_B\}$ where nodes $v \in O_B$ correspond to rebalancing origins in the i-CRRP and nodes $v \in D_B$ correspond to rebalancing destinations. Two nodes $s \in O_B, t \in D_B$ are connected by an edge $(u, v) \in E_B$ only if the flow decomposition of $\{f_B(u, v)\}_{u,v}$ contains a path from $s$ to $t$. Each edge $(s, t) \in E_B$ 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 $\{f_B(u,v)\}_{u,v}$: it encodes the origin, destination and intensity of each path flow but not actually the 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 $F$. 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 $G_B(V_B, E_B)$, with $V_B = \{O_B \cup D_B\}, O_B = \{O_B \backslash s_m\}, D_B = \{D_B \backslash t_m\}$.

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

Finding a network flow: We now wish to show that there exists a network flow $\{f_B^*(s, t)\}_{s, t}$ in $G_B$ where each node in $O_B$ is an origin with intensity $\sigma$ and each node in $D_B$ is a destination with intensity $\sigma$ 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 $\{f_B^*(s, t)\}_{s, t}$. We assign intensity $\phi - f_B(s, t_m)$ to every origin node $u \in O_B$. Note that $\phi - f_B(s, t_m) \geq \phi - (\phi - \sigma) \geq \sigma$. We also assign intensity $\phi - f_B(s_m, t) \geq \sigma$ to every destination node $v \in D_B$.

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

The origins and destinations of $\{f_B(s, t)\}$ all have intensity no smaller than $\sigma$, and the network flow satisfies the capacity constraints: then a network flow $\{f_B^*(s, t)\}_{s, t}$ with origin and destination intensities $\sigma$ 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 $\{f_B^*(s, t)\}_{s, t}$ that is a collection of path flows. The flow $\{f_B^*(s, t)\}_{s, t}$ is generated by origins and destinations of intensity $\sigma$ and every edge $(s, t) \in \tilde{E}$ connects an origin to a destination. Therefore, for every edge $(s, t) \in \tilde{E}, f_B^*(s, t) \leq \sigma$. Then, modifying the capacity of each edge $(s, t) \in \tilde{E}$ so that $c_{B}(s,t) = \sigma$ does not affect the feasibility of $\{f_B^*(s, t)\}_{s, t}$.

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 [16]: the LP relaxation of the problem admits a solution where all flows are integral. Then, the problem

Find $\tilde{f}_B^*(s, t) \forall (s, t) \in \tilde{E}$

subject to

$$\begin{align*}
\sum_{s \in V_B} \tilde{f}_B^*(s, t) + \sum_{o \in O_B} 1_{t=o} \sigma &= \sum_{u \in V_B} \tilde{f}_B^*(u, t) + \sum_{d \in D_B} 1_{t=d} \sigma \quad \forall t \in \tilde{V}_B \\
\tilde{f}_B^*(s, t) &\leq c_{B}(s,t) \forall (s,t) \in \tilde{E}_B \\
\tilde{f}_B^*(s, t) &\in \mathbb{R}_{\geq 0} \forall (s,t) \in \tilde{E}_B
\end{align*}$$

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

Remark The collection of path flows in $f_B^*(s, t)$ induces a perfect matching on $\tilde{G}$: each node $o \in \tilde{O}$ is the origin of a path flow and each node $d \in \tilde{D}$ is the destination of a path flow. Thus, $f_B^*(s, t)$ 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 $f_B^*(s, t)$ in $G_B$ with

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

The flow is a rebalancing solution: (i) each edge $(s, t) \in \tilde{E}_B$ with $f_B^*(s, t) > 0$ is a path flow of intensity $\sigma$ 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 $\{f_B^*(s, t)\}_{s, t}$ to a rebalancing solution $\{f_{rs}(u,v)\}_{u,v}$ in the original graph $G(V, E)$. We start with the null network flow $f(u,v) = 0 \forall (u,v) \in \tilde{E}$. Then, for every edge $(s, t) \in \tilde{E}_B$, we consider the corresponding path flow in $G(V, E)$. If $f_B^*(s, t) = \sigma$, we add flow $\sigma$ 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 $\sigma$. Furthermore, every origin and every destination in $G$ has an incident path flow, since every node in $O_B$ and $D_B$ (corresponding to the origin and destination nodes in $G_B$ respectively) has one incident edge with flow $\sigma$ in $\{f_B^*(s, t)\}_{s, t}$. Therefore $\{f_{rs}(u,v)\}_{u,v}$
is a fractional rebalancing solution.

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

Finally, the path flow \( p_{s_m t_m} \in \mathcal{F} \) connecting \( s_m \) and \( t_m \) has intensity \( \sigma \) both in \( f_{rs}(u,v) \) and in \( f_{R}(u,v) \). By construction, for any edge \( (u,v) \in p_{s_m t_m} \), \( f_{rs}(u,v) \geq \sigma \).

Therefore the rebalancing solution contains the path flow \( p_{s_m t_m} \).

This concludes our proof.

---

**Algorithm 4** Flow decomposition algorithm [16, Sec. 3.5].

**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 \( \{(s, u), (u, v), \ldots (w, t)\} \).
- A list of path intensities \( \{\lambda_i\}_i \)

**procedure** \( FLOWDECOMPOSITION(\{f_m(u,v)\}_{(u,v)}) \)

\[
i = 1;
\]

\[
\text{while } \sum_{s \in \mathcal{O}} \sum_{v \in V} f_m(s, v) > 0 \text{ do}
\]

\[
Path_i \leftarrow \text{a path from some } s \in \mathcal{O} \text{ to some } t \in \mathcal{D}
\]

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

\[
f_m(u, v) = f_m(u, v) - \lambda_i
\]

\[
i = i + 1
\]

\[
\text{return the path flows } \{\{Path_i\}, \{\lambda_i\}\}_i
\]