**Autonomous Robots manuscript No.**
(will be inserted by the editor)

# Routing autonomous vehicles in congested transportation networks: structural properties and coordination algorithms

**Federico Rossi · Rick Zhang · Yousef Hindy · Marco Pavone**

Received: date / Accepted: date

**Abstract** This paper considers the problem of routing and rebalancing a shared fleet of autonomous (i.e., self-driving) vehicles providing on-demand mobility within a *capacitated* transportation network, where congestion might disrupt throughput. We model the problem within a network flow framework and show that under relatively mild assumptions the rebalancing vehicles, if properly coordinated, do not lead to an increase in congestion (in stark contrast to common belief). From an algorithmic standpoint, such theoretical insight suggests that the problem of routing customers and rebalancing vehicles can be *decoupled*, which leads to a computationally-efficient routing and rebalancing algorithm for the autonomous vehicles. Numerical experiments and case studies corroborate our theoretical insights and show that the proposed algorithm outperforms state-of-the-art point-to-point methods by avoiding excess congestion on the road. Collectively, this paper provides a rigorous approach to the problem of congestion-aware, system-wide coordination of autonomously driving vehicles, and to the characterization of the sustainability of such robotic systems.

**Keywords** Self-driving cars · intelligent transportation systems · vehicle routing · autonomous systems

An earlier version of this paper was presented at the Robotics: Science and Systems Conference, 2016. This extended and revised version includes a full proof of all theorems and lemmas presented in the paper. It also includes a significantly extended simulation section, including a numerical investigation of the capacity-symmetry of major U.S. cities and a characterization of the performance of the proposed real-time algorithm with a state-of-the-art microscopic agent-based simulator, MATSim.

This research was supported by the National Science Foundation under CAREER Award CMMI-1454737, the Toyota Research Institute (TRI), and the Dr. Cleve B. Moler Stanford Graduate Fellowship.

Rick Zhang worked on this paper while he was a Ph.D. student at Stanford University.

Federico Rossi
Stanford University, Department of Aeronautics and Astronautics
E-mail: frossi2@stanford.edu

Rick Zhang
Zoox Inc.
E-mail: rick@zoox.com

Yousef Hindy
Stanford University, Department of Physics
E-mail: yhindy@stanford.edu

Marco Pavone
Stanford University, Department of Aeronautics and Astronautics
E-mail: pavone@stanford.edu

## 1 Introduction

Autonomous (i.e., robotic, self-driving) vehicles are rapidly becoming a reality and hold great promise for increasing safety and enhancing mobility for those unable or unwilling to drive (Mitchell et al, 2010; Urmson, 2014). A particularly attractive operational paradigm involves coordinating a fleet of autonomous vehicles to provide on-demand service to customers, also called autonomous mobility-on-demand (AMoD). An AMoD system may reduce the cost of travel (Spieser et al, 2014) as well as provide additional sustainability benefits such as increased overall vehicle utilization, reduced demand for urban parking infrastructure, and reduced pollution (with electric vehicles) (Mitchell et al, 2010). The key benefits of AMoD are realized through vehicle sharing, where each vehicle, after servicing a customer, drives itself to the location of the next customer or *rebalances* itself throughout the city in anticipation of future customer demand (Pavone et al, 2012).

In terms of traffic congestion, however, there has been no consensus on whether autonomous vehicles in general, and AMoD systems in particular, will ultimately be beneficial or detrimental. It has been argued that by having faster reaction times, autonomous vehicles may be able to drive

faster and follow other vehicles at closer distances without compromising safety, thereby effectively increasing the capacity of a road and reducing congestion. They may also be able to interact with traffic lights to reduce full stops at intersections (Pérez et al, 2010). On the downside, the process of vehicle rebalancing (empty vehicle trips) increases the total number of vehicles on the road (assuming the number of vehicles with customers stays the same). Indeed, it has been argued that the presence of many rebalancing vehicles may contribute to an *increase* in congestion (Templeton, 2010; Barnard, 2016). These statements, however, do not take into account that in an AMoD system the operator has control over the actions (destination and routes) of the vehicles, and may route vehicles intelligently to avoid increasing congestion or perhaps even decrease it.

Accordingly, the goal of this paper is twofold. First, on an engineering level, we aim to devise routing and rebalancing algorithms for an autonomous vehicle fleet that seek to minimize congestion. Second, on a socio-economic level, we aim to rigorously address the concern that autonomous cars may lead to increased congestion and thus disrupt currently congested transportation infrastructures.

*Literature review:* In this paper, we investigate the problem of controlling an AMoD system within a road network in the presence of congestion effects. Previous work on AMoD systems have primarily concentrated on the rebalancing problem (Pavone et al, 2012; Spieser et al, 2014), whereby one strives to allocate empty vehicles throughout a city while minimizing fuel costs or customer wait times. The rebalancing problem has been studied in (Pavone et al, 2012) using a fluidic model and in (Zhang and Pavone, 2016) using a queueing network model. An alternative formulation is the one-to-one pickup and delivery problem (Berbeglia et al, 2010), where a fleet of vehicles service pickup and delivery requests within a given region. Combinatorial asymptotically optimal algorithms for pickup and delivery problems were presented in (Treleaven et al, 2011, 2013), and generalized to road networks in (Treleaven et al, 2012). Almost all current approaches assume point-to-point travel between origins and destinations (no road network), and even routing problems on road networks (e.g. (Treleaven et al, 2012)) do not take into account vehicle-to-vehicle interactions that would cause congestion and reduce system throughput.

On the other hand, traffic congestion has been studied in economics and transportation for nearly a century. The first congestion models (Wardrop, 1952; Lighthill and Whitham, 1955; Daganzo, 1994) sought to formalize the relationship between vehicle speed, density, and flow. Since then, approaches to modeling congestion have included empirical (Kerner, 2009), simulation-based (Treiber et al, 2000; Yang and Koutsopoulos, 1996; Balmer et al, 2009; Fagnant and Kockelman, 2014), queueing-theoretical (Osorio and Bierlaire, 2009), and optimization (Peeta and Mahmassani, 1995; Janson, 1991). While there have been many high fidelity congestion models that can accurately predict traffic patterns, the primary goal of congestion modeling has been the *analysis* of traffic behavior. Efforts to *control* traffic have been limited to the control of intersections (Le et al, 2015; Xiao et al, 2015) and freeway on-ramps (Papageorgiou et al, 1991) because human drivers behave non-cooperatively. The problem of cooperative, system-wide routing (a key benefit of AMoD systems) is similar to the dynamic traffic assignment problem (DTA) (Janson, 1991) and to (Wilkie et al, 2011, 2014) in the case of online routing. The key difference is that these approaches only optimize routes for passenger vehicles while we seek to optimize the routes of *both* passenger vehicles *and* empty rebalancing vehicles.

*Statement of contributions:* The contribution of this paper is threefold. First, we model an AMoD system within a network flow framework, whereby customer-carrying and empty rebalancing vehicles are represented as flows over a *capacitated* road network (in such model, when the flow of vehicles along a road reaches a critical capacity value, congestion effects occur). Within this model, we provide a cut condition for the road graph that needs to be satisfied for congestion-free customer and rebalancing flows to exist. Most importantly, under the assumption of a *symmetric* road network, we investigate an existential result that leads to two key conclusions: (1) rebalancing does not increase congestion, and (2) for certain cost functions, the problems of finding customer and rebalancing flows can be decoupled. Second, leveraging the theoretical insights, we propose a computationally-efficient algorithm for congestion-aware routing and rebalancing of an AMoD system that is broadly applicable to time-varying, possibly asymmetric road networks. Third, through numerical studies on real-world traffic data, we validate our assumptions and show that the proposed real-time routing and rebalancing algorithm outperforms state-of-the-art point-to-point rebalancing algorithms in terms of lower customer wait times by avoiding excess congestion on the road.

*Organization:* The remainder of this paper is organized as follows: in Section 2 we present a network flow model of an AMoD system on a capacitated road network and formulate the routing and rebalancing problem. In Section 3 we present key structural properties of the model including fundamental limitations of performance and conditions for the existence of feasible (in particular, congestion-free) solutions. The insights from Section 3 are used to develop a practical real-time routing and rebalancing algorithm in Section 4. Numerical studies and simulation results are presented in Section 5, and in Section 6 we draw conclusions and discuss directions for future work.

## 2 Model Description and Problem Formulation

In this section we formulate a network flow model for an AMoD system operating over a capacitated road network. The model allows us to derive key structural insights into the vehicle routing and rebalancing problem, and motivates the design of real-time, congestion-aware algorithms for coordinating the robotic vehicles. We start in Section 2.1 with a discussion of our congestion model; then, in Section 2.2 we provide a detailed description of the overall AMoD system model.

### 2.1 Congestion Model

We use a simplified congestion model consistent with classical traffic flow theory (Wardrop, 1952). In classical traffic flow theory, at low vehicle densities on a road link, vehicles travel at the free flow speed of the road (imposed by the speed limit). This is referred to as the free flow phase of traffic. In this phase, the free flow speed is approximately constant (Kerner, 2009). The flow, or flow rate, is the number of vehicles passing through the link per unit time, and is given by the product of the speed and density of vehicles. When the flow of vehicles reaches an empirically observed critical value, the flow reaches its maximum. Beyond the critical flow rate, vehicle speeds are dramatically reduced and the flow decreases, signaling the beginning of traffic congestion. The maximum stationary flow rate is called the *capacity* of the road link in the literature. In our approach, road capacities are modeled as *constraints on the flow of vehicles*. In this way, the model captures the behavior of vehicles up to the onset of congestion.

This simplified congestion model is adequate for our purposes because the goal is not to analyze the behavior of vehicles in congested networks, but to control vehicles in order to avoid the onset of congestion. We also do not explicitly model delays at intersections, spillback behavior due to congestion, or bottleneck behavior due to the reduction of the number of lanes on a road link. An extension to our model that accommodates (limited) congestion on links is presented in Section 5.2.

### 2.2 Network Flow Model of AMoD system

We consider a road network modeled as a directed graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set. Figure 1 shows one such network. The nodes $v$ in $\mathcal{V}$ represent intersections and locations for trip origins/destinations, and the edges $(u, v)$ in $\mathcal{E}$ represent road links. As discussed in Section 2.1, congestion is modeled by imposing capacity constraints on the road links: each constraint represents the capacity of the road upon the onset of congestion. Specifically, for each road link $(u, v) \in \mathcal{E}$, we denote by $c(u, v) : \mathcal{E} \mapsto \mathbb{N}_{>0}$ the capacity of that link. When the flow rate on a road link is less than the capacity of the link, all vehicles are assumed to travel at the free flow speed, or the speed limit of the link. For each road link $(u, v) \in \mathcal{E}$, we denote by $t(u, v) : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$ the corresponding free flow time required to traverse road link $(u, v)$. Conversely, when the flow rate on a road link is larger than the capacity of the link, the traversal time is assumed equal to $\infty$ (we reiterate that our focus in this section is on avoiding the onset of congestion).

We assume that the road network is *capacity-symmetric* (or symmetric for short): for any cut[1] $(\mathcal{S}, \bar{\mathcal{S}})$ of $G(\mathcal{V}, \mathcal{E})$, the overall capacity of the edges connecting nodes in $\mathcal{S}$ to nodes in $\bar{\mathcal{S}}$ equals the overall capacity of the edges connecting nodes in $\bar{\mathcal{S}}$ to nodes in $\mathcal{S}$, that is

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

It is easy to verify that a network is capacity-symmetric if and only if the overall capacity entering each *node* equals the capacity exiting each node., i.e.

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

If all *edges* have symmetrical capacity, i.e., for all $(u, v) \in \mathcal{E}$, $c(u, v) = c(v, u)$, then the network is capacity-symmetric. The converse statement, however, is not true in general.

Transportation requests are described by the tuple $(s, t, \lambda)$, where $s \in \mathcal{V}$ is the origin of the requests, $t \in \mathcal{V}$ is the destination, and $\lambda \in \mathbb{R}_{>0}$ is the rate of requests, in customers per unit time. Transportation requests are assumed to be stationary and deterministic, i.e., the rate of requests does not change with time and is a deterministic quantity. The set of transportation requests is denoted by $\mathcal{M} = \{(s_m, t_m, \lambda_m)\}_m$, and its cardinality is denoted by $M$.

Single-occupancy vehicles travel within the network while servicing the transportation requests. We denote $f_m(u, v) : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$, $m = \{1, \ldots, M\}$, as the *customer flow* for requests $m$ on edge $(u, v)$, i.e., the amount of flow from origin $s_m$ to destination $t_m$ that uses link $(u, v)$. We also denote $f_R(u, v) : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$ as the *rebalancing flow* on edge $(u, v)$, i.e., the amount of rebalancing flow traversing edge $(u, v)$ needed to realign the vehicles with the asymmetric distribution of transportation requests.

### 2.3 The Routing Problem

The goal is to compute flows for the autonomous vehicles that (i) transfer customers to their desired destinations in

[1] For any subset of nodes $\mathcal{S} \subseteq \mathcal{V}$, we define a *cut* $(\mathcal{S}, \bar{\mathcal{S}}) \subseteq \mathcal{E}$ as the set of edges whose origin lies in $\mathcal{S}$ and whose destination lies in $\bar{\mathcal{S}} = \{\mathcal{V} \setminus \mathcal{S}\}$. Formally, $(\mathcal{S}, \bar{\mathcal{S}}) := \{(u, v) \in \mathcal{E} : u \in \mathcal{S}, v \in \bar{\mathcal{S}}\}$.

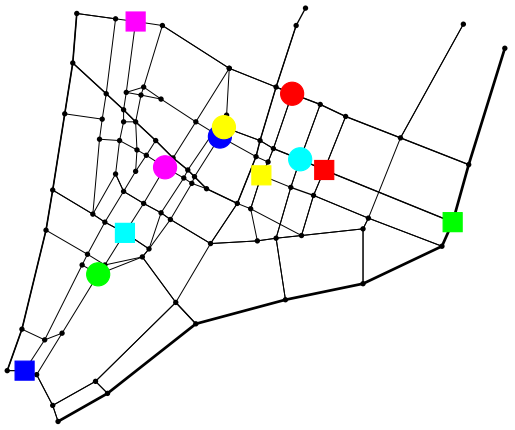

**Fig. 1** A road network modeling Lower Manhattan and the Financial District. Nodes (denoted by small black dots) model intersections; select nodes, denoted by colored circular and square markers, model passenger trips' origins and destinations. Different trip requests are denoted by different colors. Roads are modeled as edges; line thickness is proportional to road capacity.

minimum time (customer-carrying trips) and (ii) rebalance vehicles throughout the network to realign the vehicle fleet with transportation demand (customer-empty trips). Specifically, the *Congestion-free Routing and Rebalancing Problem (CRRP)* is formally defined as follows. Given a capacitated, symmetric network $G(\mathcal{V},\mathcal{E})$, a set of transportation requests $\mathcal{M}=\{(s_m,t_m,\lambda_m)\}_m$, and a weight factor $\rho>0$, solve

$$\underset{f_m(\cdot,\cdot),f_R(\cdot,\cdot)}{\text{minimize}} \sum_{m\in\mathcal{M}}\sum_{(u,v)\in\mathcal{E}} t(u,v)f_m(u,v) + \rho\sum_{(u,v)\in\mathcal{E}} t(u,v)f_R(u,v) \tag{1}$$

$$\text{subject to} \quad \sum_{u\in\mathcal{V}} f_m(u,s_m)+\lambda_m = \sum_{w\in\mathcal{V}} f_m(s_m,w) \;\; \forall m\in\mathcal{M} \tag{2}$$

$$\sum_{u\in\mathcal{V}} f_m(u,t_m) = \lambda_m + \sum_{w\in\mathcal{V}} f_m(t_m,w) \;\; \forall m\in\mathcal{M} \tag{3}$$

$$\sum_{u\in\mathcal{V}} f_m(u,v) = \sum_{w\in\mathcal{V}} f_m(v,w) \quad \forall m\in\mathcal{M}, v\in\mathcal{V}\setminus\{s_m,t_m\} \tag{4}$$

$$\sum_{u\in\mathcal{V}} f_R(u,v) + \sum_{m\in\mathcal{M}} 1_{v=t_m}\lambda_m = \sum_{w\in\mathcal{V}} f_R(v,w) + \sum_{m\in\mathcal{M}} 1_{v=s_m}\lambda_m \quad \forall v\in\mathcal{V} \tag{5}$$

$$f_R(u,v)+\sum_{m\in\mathcal{M}} f_m(u,v) \le c(u,v) \;\; \forall (u,v)\in\mathcal{E} \tag{6}$$

The cost function (1) is a weighted sum (with weight $\rho$) of the overall duration of all passenger trips and the duration of rebalancing trips. Constraints (2), (3) and (4) enforce continuity of each trip (i.e., flow conservation) across nodes. Constraint (5) ensures that vehicles are rebalanced throughout the road network to re-align vehicle distribution with transportation requests, i.e. to ensure that every outbound customer flow is matched by an inbound flow of rebalancing vehicles and vice versa. Finally, constraint (6) enforces the capacity constraint on each link (function $1_x$ denotes the indicator function of the Boolean variable $x=\{\text{true, false}\}$, that is $1_x$ equals one if $x$ is true, and equals zero if $x$ is false). Note that the CRRP is a linear program and, in particular, a special instance of the fractional multi-commodity flow problem (Ahuja et al, 1993).

We denote a customer flow $\{f_m(u,v)\}_{(u,v),m}$ that satisfies Equations (2), (3), (4) and (6) as a *feasible customer flow*. For a given set of feasible customer flows $\{f_m(u,v)\}_{(u,v),m}$, we denote a flow $\{f_R(u,v)\}_{(u,v)}$ that satisfies Equation (5) and such that the combined flows $\{f_m(u,v),f_R(u,v)\}_{(u,v),m}$ satisfy Equation (6) as a *feasible rebalancing flow*. We remark that a rebalancing flow that is feasible with respect to a set of customer flows may be infeasible for a different collection of customer flows.

For a given set of optimal flows $\{f_m^*(u,v)\}_{(u,v),m}$ and $\{f_R^*(u,v)\}_{(u,v)}$, the minimum number of vehicles needed to implement them is given by

$$V_{\min} = \left\lceil \sum_{m\in\mathcal{M}}\sum_{(u,v)\in\mathcal{E}} t(u,v)\Big(f_m^*(u,v)+f_R^*(u,v)\Big) \right\rceil .$$

This follows from a similar analysis done in (Pavone et al, 2012) for point-to-point networks. Hence, the cost function (1) is aligned with the desire of minimizing the number of vehicles needed to operate an AMoD system.

### 2.4 Discussion

A few comments are in order. First, we assume that transportation requests are time invariant. This assumption is valid when transportation requests change slowly with respect to the average duration of a customer's trip, which is often the case in dense urban environments (Neuburger, 1971). Additionally, in Section 4 we will present algorithmic tools that allow one to extend the insights gained from the time-invariant case to the time-varying counterpart. Second, the assumption of single-occupancy for the vehicles models most of the existing (human) one-way vehicle sharing systems (where the driver is considered "part" of the vehicle), and chiefly disallows the provision of ride-sharing or carpooling service (this is an aspect left for future research). Third, as also discussed in Section 2.1, our congestion model is simpler and less accurate than typical congestion models used in the transportation community. However, our model lends itself to efficient real-time optimization and thus it is well-suited to the *control* of fleets of autonomous vehicles. Existing high-fidelity congestion models should be regarded as

complementary and could be used offline to identify the congestion thresholds used in our model. Fourth, while we have defined the CRRP in terms of fractional flows, an integer-valued counterpart can be defined and (approximately) solved to find optimal routes for each *individual* customer and vehicle. Algorithmic aspects will be investigated in depth in Section 4, with the goal of devising practical, real-time routing and rebalancing algorithms. Fifth, trip requests are assumed to be known. In practice, trip requests can be reserved in advance, estimated from historical data, or estimated in real time. Finally, the assumption of capacity-symmetric road networks indeed appears reasonable for a number of major U.S. metropolitan areas (note that this assumption is much less restrictive than assuming every *individual* road is capacity-symmetric). In Section 5.1, by using OpenStreetMap data (Haklay and Weber, 2008), we provide a rigorous characterization in terms of capacity symmetry of the road networks of New York City, Chicago, Los Angeles and other major U.S. cities. The results consistently show that urban road networks are usually symmetric to a *very high* degree. Additionally, several of our theoretical and algorithmic results extend to the case where this assumption is lifted, as it will be highlighted throughout the paper.

## 3 Structural Properties of the Network Flow Model

n this section we provide two key structural results for the network flow model presented in Section 2.2. First, we provide a cut condition that needs to be satisfied for feasible customer and rebalancing flows to exist. In other words, this condition provides a fundamental limitation of performance for congestion-free AMoD service in a given road network. Second, we investigate an existential result (our main theoretical result) that is germane to two key conclusions: (1) rebalancing does not increase congestion in symmetric road networks, and (2) for certain cost functions, the problems of finding customer and rebalancing flows can be *decoupled* – an insight that will be heavily exploited in subsequent sections.

### 3.1 Fundamental Limitations

We start with a few definitions. For a given set of feasible customer flows $\{f_m(u,v)\}_{(u,v),m}$, we denote by $F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})$ the overall flow exiting a cut $(\mathcal{S},\bar{\mathcal{S}})$, i.e.,

$$F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}}) := \sum_{m\in\mathcal{M}} \sum_{u\in\mathcal{S},v\in\bar{\mathcal{S}}} f_m(u,v).$$

Similarly, we denote by $C_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})$ the capacity of the network exiting $\mathcal{S}$, i.e., $C_{\text{out}}(\mathcal{S},\bar{\mathcal{S}}) = \sum_{u\in S,v\in\bar{\mathcal{S}}} c(u,v)$. Analogously, $F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})$ denotes the overall flow entering $\mathcal{S}$ from $\bar{\mathcal{S}}$, i.e., $F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}}) := F_{\text{out}}(\bar{\mathcal{S}},\mathcal{S})$, and $C_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})$ denotes the capacity entering $\mathcal{S}$ from $\bar{\mathcal{S}}$, i.e., $C_{\text{in}}(\mathcal{S},\bar{\mathcal{S}}) := C_{\text{out}}(\bar{\mathcal{S}},\mathcal{S})$. We highlight that the arguments leading to the main result of this subsection (Theorem 1) do not require the assumption of capacity symmetry; hence, Theorem 1 holds for *asymmetric* road networks as well.

The next technical lemma shows that the net flow leaving set $\mathcal{S}$ equals the difference between the flow originating from the origins $s_m$ in $\mathcal{S}$ and the flow exiting through the destinations $t_m$ in $\mathcal{S}$, that is,

**Lemma 1 (Net flow across a cut)** *Consider a set of feasible customer flows $\{f_m(u,v)\}_{(u,v),m}$. Then, for every cut $(\mathcal{S},\bar{\mathcal{S}})$, the net flow leaving set $\mathcal{S}$ satisfies*

$$F_{out}(\mathcal{S},\bar{\mathcal{S}}) - F_{in}(\mathcal{S},\bar{\mathcal{S}}) = \sum_{m\in\mathcal{M}} 1_{s_m\in\mathcal{S}}\lambda_m - \sum_{m\in\mathcal{M}} 1_{t_m\in\mathcal{S}}\lambda_m.$$

*Proof (Proof of Lemma 1)* We compute the sum over all customer flows $m\in\mathcal{M}$ and over all nodes $v\in\mathcal{V}$ of the node balance equation for flow $m$ at node $v$ (Equation (3) if node $v$ is the source of $m$, Equation (4) if node $v$ is the sink of $m$, or Equation (2) otherwise). We obtain

$$\sum_{v\in\mathcal{S}}\sum_{m\in\mathcal{M}}\left(\sum_{u\in\mathcal{V}} f_m(u,v) + 1_{v=s_m}\lambda_m\right) = \sum_{v\in\mathcal{S}}\sum_{m\in\mathcal{M}}\left(\sum_{w\in\mathcal{V}} f_m(v,w) + 1_{v=t_m}\lambda_m\right).$$

For any edge $(u,v)$ such that $u,v\in\mathcal{S}$, the customer flow $f_m(u,v)$ appears on both sides of the equation. Thus the equation above simplifies to

$$\sum_{m\in\mathcal{M}}\sum_{v\in\mathcal{S}}\left(\sum_{u\in\bar{\mathcal{S}}} f_m(u,v) + 1_{v=s_m}\lambda_m\right) = \sum_{m\in\mathcal{M}}\sum_{v\in\mathcal{S}}\left(\sum_{w\in\bar{\mathcal{S}}} f_m(v,w) + 1_{v=t_m}\lambda_m\right),$$

which leads to the claim of the lemma

$$F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}}) + \sum_{m\in\mathcal{M}} 1_{s_m\in\mathcal{S}}\lambda_m = F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}}) + \sum_{m\in\mathcal{M}} 1_{t_m\in\mathcal{S}}\lambda_m.$$

□

We now state two additional lemmas providing, respectively, lower and upper bounds for the outflows $F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})$.

**Lemma 2 (Lower bound for outflow)** *Consider a set of feasible customer flows $\{f_m(u,v)\}_{(u,v),m}$. Then, for any cut $(\mathcal{S},\bar{\mathcal{S}})$, the overall flow $F_{out}(\mathcal{S},\bar{\mathcal{S}})$ exiting cut $(\mathcal{S},\bar{\mathcal{S}})$ is lower bounded according to*

$$\sum_{m\in\mathcal{M}} 1_{s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\lambda_m \le F_{out}(\mathcal{S},\bar{\mathcal{S}}).$$

*Proof* Adding Equations (2), (3) and (4) over all nodes in $\mathcal{S}$ and over all flows whose origin is in $\mathcal{S}$ and whose destination is in $\bar{\mathcal{S}}$, one obtains

$$\sum_{m:s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\sum_{v\in\mathcal{S}}\left(\sum_{u\in\mathcal{V}}f_m(u,v)+1_{v=s_m}\lambda_m\right)=\sum_{m:s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\sum_{v\in\mathcal{S}}\left(\sum_{w\in\mathcal{V}}f_m(v,w)\right).$$

Flows $f_m(u,v)$ such that both $u$ and $v$ are in $\mathcal{S}$ appear on both sides of the equation. Simplifying, one obtains

$$\sum_{m\in\mathcal{M}}1_{s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\lambda_m=\sum_{m:s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\left(\sum_{v\in\mathcal{S},w\in\bar{\mathcal{S}}}f_m(v,w)-\sum_{v\in\mathcal{S},u\in\bar{\mathcal{S}}}f_m(u,v)\right)$$

The first term on the right-hand side represents a lower bound for $F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})$, since

$$\begin{aligned}F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})&=\sum_{m\in\mathcal{M}}\sum_{v\in\mathcal{S},w\in\bar{\mathcal{S}}}f_m(v,w)\\&\geq\sum_{m:s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\sum_{v\in\mathcal{S},w\in\bar{\mathcal{S}}}f_m(v,w).\end{aligned}$$

Furthermore, the second term on the right-hand side is upper-bounded by zero. The lemma follows. □

**Lemma 3 (Upper bound for outflow)** *Assume there exists a set of* feasible *customer and rebalancing flows* $\{f_m(u,v),\ f_R(u,v)\}_{(u,v),m}$. *Then, for every cut* $(\mathcal{S},\bar{\mathcal{S}})$,

1. $F_{out}(\mathcal{S},\bar{\mathcal{S}})\leq C_{out}(\mathcal{S},\bar{\mathcal{S}})$, *and*
2. $F_{out}(\mathcal{S},\bar{\mathcal{S}})\leq C_{in}(\mathcal{S},\bar{\mathcal{S}})$.

*Proof* The first condition follows trivially from equation (6). As for the second condition, consider a cut $(\mathcal{S},\bar{\mathcal{S}})$. Analogously as for the definitions of $F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})$ and $F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})$, let $F^{\text{reb}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})$ and $F^{\text{reb}}_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})$ denote, respectively, the overall rebalancing flow entering (exiting) cut $(\mathcal{S},\bar{\mathcal{S}})$. Summing equation (5) over all nodes in $S$, one easily obtains

$$F^{\text{reb}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})-F^{\text{reb}}_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})=\sum_{m\in\mathcal{M}}1_{s_m\in\mathcal{S}}\lambda_m-\sum_{m\in\mathcal{M}}1_{t_m\in\mathcal{S}}\lambda_m.$$

Combining the above equation with Lemma 1, one obtains

$$F^{\text{reb}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})-F^{\text{reb}}_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})=F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})-F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}}),$$

in other words, rebalancing flows should make up the difference between the customer inflows and outflows across cut $(\mathcal{S},\bar{\mathcal{S}})$. Accordingly, the total inflow of vehicles across $(\mathcal{S},\bar{\mathcal{S}})$, $F^{\text{tot}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})$, satisfies the inequality

$$\begin{aligned}F^{\text{tot}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}}):=&F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})+F^{\text{reb}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})\\=&F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})+F^{\text{reb}}_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})+F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})\\&-F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})\\\geq&F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}}).\end{aligned}$$

Since the customer and rebalancing flows $\{f_m(u,v),\ f_R(u,v)\}_{(u,v),m}$ are feasible, then, by equation (6), $F^{\text{tot}}_{\text{in}}(S,\bar{S})\leq C_{\text{in}}(S,\bar{S})$. Collecting the results, one obtains the second condition. □

We are now in a position to present a *structural* (i.e., flow-independent) necessary condition for the existence of feasible customer and rebalancing flows.

**Theorem 1 (Necessary condition for feasible flows)** *A necessary condition for the existence of a set of* feasible *customer and rebalancing flows* $\{f_m(u,v),\ f_R(u,v)\}_{(u,v),m}$, *is that, for every cut* $(\mathcal{S},\bar{\mathcal{S}})$,

1. $\sum_{m\in\mathcal{M}}1_{s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\lambda_m\leq C_{out}(\mathcal{S},\bar{\mathcal{S}})$, *and*
2. $\sum_{m\in\mathcal{M}}1_{s_m\in\mathcal{S},t_m\in\bar{\mathcal{S}}}\lambda_m\leq C_{in}(\mathcal{S},\bar{\mathcal{S}})$.

*Proof* The theorem is a trivial consequence of Lemmas 2 and 3. □

Theorem 1 essentially provides a structural fundamental limitation of performance for a given road network: if the cut conditions in Theorem 1 are not met, then there is no hope of finding congestion-free customer and rebalancing flows. We reiterate that Theorem 1 holds for both symmetric and asymmetric networks (for a symmetric network, claim 2) in Lemma 3 and condition 2) in Theorem 1 are redundant).

### 3.2 Existence of Congestion-Free Flows

In this section we address the following question: assuming there exists a feasible customer flow, is it always possible to find a feasible rebalancing flow? As we will see, the answer to this question is affirmative and has both conceptual and algorithmic implications.

**Theorem 2 (Feasible rebalancing)** *Assume there exists a set of feasible customer flows* $\{f_m(u,v)\}_{(u,v),m}$. *Then, it is* always *possible to find a set of feasible rebalancing flows* $\{f_R(u,v)\}_{(u,v)}$.

*Proof* We prove the theorem for the special case where no node $v\in\mathcal{V}$ is associated with both an origin and a destination for the transportation requests in $\mathcal{M}$. This is without loss of generality, as the general case where a node $v$ has both an origin and a destination assigned can be reduced to this special case, by associating with node $v$ a "shadow"

node so that (i) all destinations are assigned to the shadow node and (ii) node $v$ and its shadow node are mutually connected via an infinite-capacity, zero-travel-time edge.

We start the proof by defining the concepts of *partial rebalancing flows* and *defective origins and destinations*. Specifically, a partial rebalancing flow, denoted as $\{\hat{f}_R(u,v)\}_{(u,v)}$, is a set of mappings from $\mathcal{E}$ to $\mathbb{R}_{\geq 0}$ obeying the following properties:

1. It satisfies constraint (5) at every node that is not an origin nor a destination, that is $\forall\, v \in \{\mathcal{V} \setminus \{\{s_m\}_m \cup \{t_m\}_m\}\}$,

$$\sum_{u\in\mathcal{V}} \hat{f}_R(u,v) = \sum_{w\in\mathcal{V}} \hat{f}_R(v,w).$$

2. It violates constraint (5) in the "$\leq$ direction" at every node that is an origin, that is $\forall\, v \in \mathcal{V}$ such that $\exists m \in \mathcal{M} : v = s_m$,

$$\sum_{u\in\mathcal{V}} \hat{f}_R(u,v) \leq \sum_{w\in\mathcal{V}} \hat{f}_R(v,w) + \sum_{m\in\mathcal{M}} 1_{v=s_m}\lambda_m.$$

3. It violates constraint (5) in the "$\geq$ direction" at every node that is a destination, that is $\forall\, v \in \mathcal{V}$ such that $\exists m \in \mathcal{M} : v = t_m$,

$$\sum_{u\in\mathcal{V}} \hat{f}_R(u,v) + \sum_{m\in\mathcal{M}} 1_{v=t_m}\lambda_m \geq \sum_{w\in\mathcal{V}} \hat{f}_R(v,w).$$

4. The combined customer and partial rebalancing flows $\{f_m(u,v), \hat{f}_R(u,v)\}_{(u,v),m}$ satisfy Equation (6) for every edge $(u,v) \in \mathcal{E}$.

Note that the trivial zero flow, that is $\hat{f}_R(u,v) = 0$ for all $(u,v) \in \mathcal{E}$, is a partial rebalancing flow (in other words, the set of partial rebalancing flows in not empty). Clearly a feasible rebalancing flow is also a partial rebalancing flow, but the opposite is not necessarily true.

For a given partial rebalancing flow, we denote an origin node, that is a node $v \in \mathcal{V}$ such that $v = s_m$ for some $m = 1,\ldots,M$, as a *defective* origin if Equation (5) is not satisfied at $v = s_m$ (in other words, the strict inequality $<$ holds). Analogously, we denote a destination node, that is a node $v \in \mathcal{V}$ such that $v = t_m$ for some $m = 1,\ldots,M$, as a *defective* destination if Equation (5) is not satisfied at $v = t_m$ (in other words, the strict inequality $>$ holds). The next lemma links the concepts of partial rebalancing flows and defective origins/destinations.

**Lemma 4 (Co-existence of defective origins/destinations)** *For every partial rebalancing flow that is not a feasible rebalancing flow, there exists at least one node $u \in \mathcal{V}$ that is a defective origin,* and *one node $v \in \mathcal{V}$ that is a defective destination.*

*Proof* By contradiction. Since the flow $\{\hat{f}_R(u,v)\}_{(u,v)}$ is not a feasible rebalancing flow, there exists at least one defective origin or a defective destination. Assume that there exists at least one defective destination, say a node $\hat{t}_j$ where Equation (5) is violated:

$$\sum_{u\in\mathcal{V}} \hat{f}_R(u,\hat{t}_j) + \sum_{m\in\mathcal{M}} 1_{\hat{t}_j=t_m}\lambda_m > \sum_{w\in\mathcal{V}} \hat{f}_R(\hat{t}_j,w),$$

Now, assume that there does not exist any defective origin. By summing Equation (5) over all nodes $v \in \mathcal{V}$ and simplifying all flows $\hat{f}_R(u,v)$ (as they appear on both sides of the resulting equation), one obtains

$$\sum_{v\in\mathcal{V}}\sum_{m\in\mathcal{M}} 1_{v=t_m}\lambda_m > \sum_{v\in\mathcal{V}}\sum_{m\in\mathcal{M}} 1_{v=s_m}\lambda_m,$$

that is $\sum_{m\in\mathcal{M}}\lambda_m > \sum_{m\in\mathcal{M}}\lambda_m$, which is a contradiction. Noticing that the symmetric case where we assume that there exists at least one defective destination leads to an analogous contradiction, the lemma follows. □

For a given set of customer flows $\{f_m(u,v)\}_{(u,v),m}$ and partial rebalancing flows $\{\hat{f}_R(u,v)\}_{(u,v)}$, we call an edge $(u,v) \in \mathcal{E}$ *saturated* if equation (6) holds with equality for that edge. We call a path *saturated* if at least one of the edges along the path is saturated. We now prove the existence of a special partial rebalancing flow where defective destinations and defective origins are separated by a graph cut formed exclusively by saturated edges (this result, and its consequences, are illustrated in Figure 2).

**Lemma 5 (Existence of partial rebalancing flows)** *Assume there exists a set of* feasible *customer flows $\{f_m(u,v)\}_{(u,v),m}$, but there does not exist a set of feasible rebalancing flows $\{f_R(u,v)\}_{(u,v)}$. Then, there exists a partial rebalancing flow $\{\hat{f}_R(u,v)\}_{(u,v)}$ that induces a graph cut $(\mathcal{S},\bar{\mathcal{S}})$ with the following properties: (i) all defective destinations are in $\mathcal{S}$, (ii) all defective origins are in $\bar{\mathcal{S}}$, and (iii) all edges in $(S,\bar{\mathcal{S}})$ are saturated.*

*Proof* The proof is constructive and constructs the desired partial rebalancing flow by starting with the trivial zero flow $\hat{f}_R(u,v) = 0$ for all $(u,v) \in \mathcal{E}$. Let $\mathcal{V}_{\text{or, def}} := \{\hat{s}_1,\ldots,\hat{s}_{|\mathcal{V}_{\text{or, def}}|}\}$ and $\mathcal{V}_{\text{dest, def}} := \{\hat{t}_1,\ldots,\hat{t}_{|\mathcal{V}_{\text{dest, def}}|}\}$ be the set of defective origins and destinations, respectively, under such flow. Then, the zero flow is iteratively updated according to the following procedure:

1. Look for a path between a node in $\mathcal{V}_{\text{dest, def}}$ and a node in $\mathcal{V}_{\text{or, def}}$ that is not saturated (note that for rebalancing flows, paths go from destinations to origins). If no such path exists, quit. Otherwise, go to Step 2.

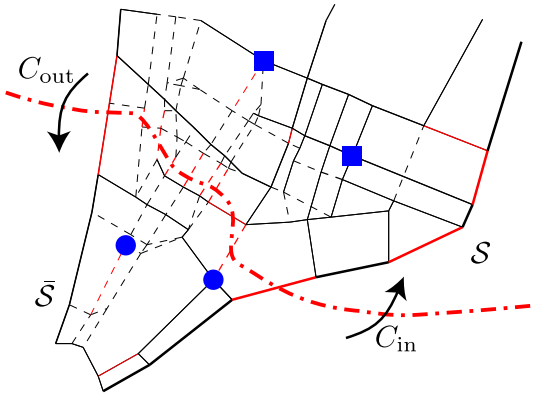


**Fig. 2** A graphical representation of Lemma 5. If there exists a set of feasible customer flows but there does not exist a set of feasible rebalancing flows, one can find a partial rebalancing flow where all the defective origins, represented as blue circles, are separated from all the defective destinations, represented as blue squares, by a cut of saturated edges (shown in red). Note that not all saturated edges necessarily belong to the cut. In the proof of Theorem 2 we show that the capacity of such a cut $(\mathcal{S}, \bar{\mathcal{S}})$ is asymmetric, i.e., $C_{\text{out}} < C_{\text{in}}$ – a contradiction that leads to the claim of Theorem 2.

2. Add the same amount of flow on all edges along the path until either (i) one of the edges becomes saturated or (ii) constraint (5) is fulfilled either at the defective origin or at the defective destination. Note that the resulting flow remains a partial rebalancing flow.
3. Update sets $\mathcal{V}_{\text{or, def}}$ and $\mathcal{V}_{\text{dest, def}}$ for the new partial rebalancing flow and go to Step 1.

The algorithm terminates. To show this, we prove the invariant that if a node is no longer defective for the updated partial rebalancing flow (in other words, Step 2 ends due to condition (ii)), it will not become defective at a later stage. Consider a defective destination node $v$ that becomes non-defective under the updated partial rebalancing flow (the proof for defective origins is analogous). Then, at the subsequent stage it cannot be considered as a destination in Step 1 (as it is no longer in set $\mathcal{V}_{\text{dest, def}}$). If a path that does not contain $v$ is selected, then $v$ stays non-defective. Otherwise, if a path that contains $v$ is selected, then, after Step 2, both the inbound flow (that is the flow into $v$) and the outbound flow (that is the flow out of $v$) will be increased by the same quantity, and the node will stay non-defective. An induction on the stages then proves the claim. As the number of paths is finite, and sets $\mathcal{V}_{\text{or, def}}$ and $\mathcal{V}_{\text{dest, def}}$ cannot have any nodes added, the algorithm terminates after a finite number of stages.

The output of the algorithm (denoted, with a slight abuse of notation, as $\{\hat{f}_R(u,v)\}_{(u,v)}$) is a partial rebalancing flow that is not feasible (as, by assumption, there does not exist a set of feasible rebalancing flows). Therefore, by Lemma 4, such partial rebalancing flow has at least one defective origin and at least one defective destination. Let us define $\mathcal{E}_{ns} := \mathcal{E} \setminus \{(u,v) : (u,v) \text{ is saturated}\}$ as the collection of non-saturated edges under the flows $\{f_m(u,v)\}_{(u,v),m}$ and $\{\hat{f}_R(u,v)\}_{(u,v)}$. For any defective destination and any defective origin, all paths connecting them contain at least one saturated edge (due to the exit condition in Step 1). Therefore, the graph $G_{ns}(\mathcal{V}, \mathcal{E}_{ns})$ has two properties: (i) it is disconnected (that is, it is not possible to find a direct path between every pair of nodes in $\mathcal{V}$ by using edges in $\mathcal{E}_{ns}$), and (ii) a defective origin and a defective destination can not be in the same strongly connected component (hence, graph $G_{ns}(\mathcal{V}, \mathcal{E}_{ns})$ can be partitioned into at least two strongly connected components).

We now find the cut $(\mathcal{S}, \bar{\mathcal{S}})$ as follows. If a strongly connected component of $G_{ns}$ contains defective destinations, we assign its nodes to set $\mathcal{S}$. If a strongly connected component contains defective origins, we assign its nodes to set $\bar{\mathcal{S}}$. If a strongly connected component contains neither defective origins nor destinations, we assign its nodes to $\mathcal{S}$ (one could also assign its nodes to $\bar{\mathcal{S}}$, but such choice is immaterial for our purposes). By construction, $(\mathcal{S}, \bar{\mathcal{S}})$ is a cut, and its edges are all saturated. Furthermore, set $\mathcal{S}$ only contains destination nodes, and set $\bar{\mathcal{S}}$ only contains origin nodes, which concludes the proof. □

We are now in a position to prove Theorem 2. The proof is by contradiction. Assume that a set of feasible rebalancing flows $\{f_R(u,v)\}_{(u,v)}$ does not exist. Then Lemma 5 shows that there exists a partial rebalancing flow $\{\hat{f}_R(u,v)\}_{(u,v)}$ and a cut $(\mathcal{S}, \bar{\mathcal{S}})$ such that all defective destinations under $\{\hat{f}_R(u,v)\}_{(u,v)}$ belong to $\mathcal{S}$ and all defective origins belong to $\bar{\mathcal{S}}$. Let us denote the sum of all partial rebalancing flows across cut $(\mathcal{S}, \bar{\mathcal{S}})$ as

$$\hat{F}^{\text{reb}}_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) := \sum_{u \in \mathcal{S}, v \in \bar{\mathcal{S}}} \hat{f}_R(u,v),$$

and, analogously, define $\hat{F}^{\text{reb}}_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) := \hat{F}^{\text{reb}}_{\text{out}}(\bar{\mathcal{S}}, \mathcal{S})$. Since all edges in the cut $(\mathcal{S}, \bar{\mathcal{S}})$ are saturated under $\{\hat{f}_R(u,v)\}_{(u,v)}$, one has, due to equation (6), the equality

$$C_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) = F_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) + \hat{F}^{\text{reb}}_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}).$$

Additionally, again due to equation (6), one has the inequality

$$F_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) + \hat{F}^{\text{reb}}_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) \leq C_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}).$$

Combining the above equations, one obtains

$$\begin{aligned} F_{\text{in}}(S, \bar{S}) + \hat{F}^{\text{reb}}_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) - F_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) - \hat{F}^{\text{reb}}_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) \\ \leq C_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) - C_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}). \end{aligned}$$

To compute $\hat{F}^{\text{reb}}_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) - \hat{F}^{\text{reb}}_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}})$, we follow a procedure similar to the one used in Lemma 1. Summing equation (5) over all nodes in $\mathcal{S}$, one obtains,

$$\sum_{v\in\mathcal{S}}\left[\sum_{u\in\mathcal{V}}\hat{f}_R(u,v)+\sum_{m\in\mathcal{M}}1_{v=t_m}\lambda_m\right] > \sum_{v\in\mathcal{S}}\left[\sum_{w\in\mathcal{V}}\hat{f}_R(v,w)+\sum_{m\in\mathcal{M}}1_{v=s_m}\lambda_m\right].$$

The strict inequality is due to the fact that for a partial rebalancing flow that is not feasible there exists at least one defective destination (Lemma 4), which, by construction, must belong to $\mathcal{S}$. Simplifying those flows $\hat{f}_R(u,v)$ for which both $u$ and $v$ are in $\mathcal{S}$ (as such flows appear on both sides of the above inequality), one obtains

$$\hat{F}^{\text{reb}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})-\hat{F}^{\text{reb}}_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})>\sum_{m\in\mathcal{M}}\left(1_{s_m\in\mathcal{S}}-1_{t_m\in\mathcal{S}}\right)\lambda_m.$$

Also, by Lemma 1,

$$F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})-F_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})=\sum_{m\in\mathcal{M}}\left(1_{s_m\in\mathcal{S}}-1_{t_m\in\mathcal{S}}\right)\lambda_m.$$

Collecting all the results so far, we conclude that

$$\begin{aligned}0 &< F_{\text{in}}(S,\bar{S})+\hat{F}^{\text{reb}}_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})-F_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})-\hat{F}^{\text{reb}}_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})\\ &= C_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})-C_{\text{out}}(\mathcal{S},\bar{\mathcal{S}}).\end{aligned}$$

Hence, we reached the conclusion that $C_{\text{in}}(\mathcal{S},\bar{\mathcal{S}})-C_{\text{out}}(\mathcal{S},\bar{\mathcal{S}})>0$, or, in other words, the capacity of graph $G(\mathcal{V},\mathcal{E})$ across cut $(\mathcal{S},\bar{\mathcal{S}})$ is *not* symmetric. This contradicts the assumption that graph $G(\mathcal{V},\mathcal{E})$ is capacity-symmetric, and the claim follows. □

The importance of Theorem 2 is twofold. First, perhaps surprisingly, it shows that for symmetric road networks it is *always* possible to rebalance the autonomous vehicles *without* increasing congestion – in other words, the rebalancing of autonomous vehicles in a symmetric road network does *not* lead to an increase in congestion. Second, from an algorithmic standpoint, if the cost function in the CRRP only depends on the customer flows (that is, $\rho = 0$ and the goal is to minimize the customers' travel times), then the CRRP problem can be *decoupled* and the customers and rebalancing flows can be solved separately without loss of optimality. This insight will be instrumental in Section 4 to the design of real-time algorithms for routing and rebalancing.

We conclude this section by noticing that the CRRP, from a computational standpoint, can be reduced to an instance of the Minimum-Cost Multi-Commodity Flow problem (Min-MCF), a classic problem in network flow theory (Ahuja et al, 1993). The problem can be efficiently solved either via linear programming (the size of the linear program is $|\mathcal{E}|(M+1)$), or via specialized combinatorial algorithms (Goldberg et al, 1990; Leighton et al, 1995; Goldberg et al, 1998). However, the solution to the CRRP provides *static fractional* flows, which are not directly implementable for the operation of actual AMoD systems. Practical algorithms (inspired by the theoretical CRRP model) are presented in the next section.

## 4 Real-time Congestion-Aware Routing and Rebalancing

A natural approach to routing and rebalancing would be to periodically resolve the CRRP within a receding-horizon, batch-processing scheme (a common scheme for the control of transportation networks (Seow et al, 2010; Pavone et al, 2012; Zhang et al, 2016)). This approach, however, is not directly implementable as the solution to the CRRP provides *fractional* flows (as opposed to routes for the *individual* vehicles). This shortcoming can be addressed by considering an integral version of the CRRP (dubbed integral CRRP), whereby the flows are *integer*-valued and can be thus easily translated into routes for the individual vehicles, e.g. through a flow decomposition algorithm (Ford and Fulkerson, 1962). The integral CRRP, however, is an instance of the integral Minimum-Cost Multi-Commodity Flow problem, which is known to be NP-hard (Karp, 1975; Even et al, 1976). Naïve rounding techniques are inapplicable: rounding a solution for the (non-integral) CRRP does not yield, in general, feasible integral flows, and hence feasible routes. For example, continuity of vehicles and customers can not be guaranteed, and vehicles may appear and disappear along a route. In general, to the best of our knowledge, there are no polynomial-time approximation schemes for the integral Minimum-Cost Multi-Commodity Flow problem.

On the positive side, the integral CRRP admits a decoupling result akin to Theorem 2: given a set of feasible, *integral* customer flows, one can always find a set of feasible, *integral* rebalancing flows. (In fact, the proof of Theorem 2 does not exploit anywhere the property that the flows are fractional, and thus the proof extends virtually unchanged to the case where the flows are integer-valued). Our approach is to leverage this insight (and more in general the theoretical results from Section 3) to design a heuristic, yet efficient approximation to the integral CRRP that (i) scales to large-scale systems, and (ii) is general, in the sense that can be broadly applied to time-varying, asymmetric networks.

Specifically, we consider as objective the minimization of the customers' travel times, which, from Section 3 and the aforementioned discussion about the generalization of Theorem 2 to integral flows, *suggests* that customer routing can be decoupled from vehicle rebalancing (strictly speaking, this statement is only valid for static and symmetric networks – its generalization beyond these assumptions will be addressed numerically in Section 5). Accordingly, to emulate the real-world operation of an AMoD system, we divide

a given city into geographic regions (also referred to as "stations" in some formulations) (Pavone et al, 2012; Zhang and Pavone, 2016), and each arriving customer is assigned the closest vehicle *within that region* (vehicle imbalance across regions is handled separately by the vehicle rebalancing algorithm, discussed below). We apply a greedy, yet computationally-efficient and congestion-aware approach for customer routing where customers are routed to their destinations using the shortest-time path as computed by an $A^*$ algorithm (Hart et al, 1968). The travel time along each edge is computed using a heuristic delay function that is related to the current volume of traffic on each edge. A popular heuristic is the simple Bureau of Public Roads (BPR) delay model (Bureau of Public Roads, 1964), which computes the travel time on each edge $(u, v) \in \mathcal{E}$ as

$$t_d(u,v) := t(u,v)\left(1 + \alpha\left(\frac{f(u,v)}{c(u,v)}\right)^{\beta}\right),$$

where $f(u,v) := \sum_{m=1}^{M} f_m(u,v) + f_R(u,v)$ is the total flow on edge $(u,v)$, and $\alpha$ and $\beta$ are usually set to $0.15$ and $4$ respectively. Note that customer routing is *event-based*, i.e, a routing choice is made as soon as a customer arrives.

Separately from customer routing, vehicle rebalancing from one region to another is performed every $t_{\text{hor}} > 0$ time units as a batch process (unlike customer routing, which is an event-based process). Denote by $v_i(t)$ the number of vehicles in region $i$ at time $t$, and by $v_{ji}(t)$ the number of vehicles traveling from region $j$ to $i$ that will arrive in the next $t_{\text{vicinity}}$ time units. Let $v_i^{\text{own}}(t) := v_i(t) + \sum_j v_{ji}(t)$ be the number of vehicles currently "owned" by region $i$ (i.e., in the vicinity of such region). Denote by $v_i^e(t)$ the number of excess vehicles in region $i$, or the number of vehicles left after servicing the customers waiting within region $i$. From its definition, $v_i^e(t)$ is given by $v_i^e(t) = v_i^{\text{own}}(t) - c_i(t)$, where $c_i(t)$ is the number of customers within region $i$. Finally, denote by $v_i^d(t)$ the desired number of vehicles within region $i$. For example, for an even distribution of excess vehicles, $v_i^d(t) \propto \sum_i v_i^e(t)/N$, where $N$ is the number regions. Note that the $v_i^d(t)$'s are rounded so they take on integer values. The set of origin regions (i.e., regions that should send out vehicles), $S_R$, and destination regions (i.e., regions that should receive vehicles), $T_R$, for the rebalancing vehicles are then determined by comparing $v_i^e(t)$ and $v_i^d(t)$, specifically,

$$\begin{aligned}&\text{if } v_i^e(t) > v_i^d(t), \quad \text{region } i \in S_R\\ &\text{if } v_i^e(t) < v_i^d(t), \quad \text{region } i \in T_R.\end{aligned}$$

We assume the residual capacity $c_R(u,v)$ of an edge $(u,v)$, defined as the difference between its overall capacity $c(u,v)$ and the current number of vehicles along that edge, is known and remains approximately constant over the rebalancing time horizon. In case the overall rebalancing problem is not feasible (i.e. it is not possible to move all excess vehicles to regions that have a deficit of vehicles while satisfying the congestion constraints), we define slack variables with cost $C$ that allow the optimizer to select a subset of vehicles and rebalancing routes of maximum cardinality such that each link does not become congested. The slack variables are denoted as $ds_i$ for each $i \in S_R$, and $dt_j$ for each $j \in T_R$.

Every $t_{\text{hor}}$ time units, the rebalancing vehicle routes are computed by solving the following integer linear program

$$\begin{aligned}\underset{\substack{f_R(\cdot,\cdot),\\ \{ds_i\},\{dt_j\}}}{\text{minimize}} \quad & \sum_{(u,v)\in\mathcal{E}} t(u,v)\, f_R(u,v)\\ & + \sum_{i\in S_R} C ds_i + \sum_{i\in T_R} C dt_i\\ \text{subject to} \quad & \sum_{u\in\mathcal{V}} f_R(u,v) + 1_{v\in S_R}(v_v^e(t) - v_v^d(t) - ds_v)\\ & = \sum_{w\in V} f_R(v,w) + 1_{v\in T_R}(v_v^d(t) - v_v^e(t) - dt_v),\\ & \qquad\qquad \text{for all } v \in \mathcal{V}\\ & f_R(u,v) \le c_R(u,v), \quad \text{for all } (u,v)\in\mathcal{E}\\ & f_R(u,v) \in \mathbb{N}, \quad \text{for all } (u,v)\in\mathcal{E}\\ & ds_i, dt_j \in \mathbb{N}, \quad \text{for all } i \in S_R, j \in T_R\end{aligned}$$

The set of (integral) rebalancing flows $\{f_R(u,v)\}_{(u,v)}$ is then decomposed into a set of rebalancing paths via a flow decomposition algorithm (Ford and Fulkerson, 1962). Each rebalancing path connects one origin region with one destination region: thus, rebalancing paths represent the set of routes that excess vehicles should follow to rebalance to regions with a deficit of vehicles.

The rebalancing optimization problem is an instance of the Minimum Cost Flow problem. If all edge capacities are integral, the linear relaxation of the Minimum Cost Flow problem enjoys a totally unimodular constraint matrix (Ahuja et al, 1993). Hence, the linear relaxation will necessarily have an integer optimal solution, which will be a fortiori an optimal solution to the original Minimum Cost Flow problem. It follows that an integer-valued solution to the rebalancing optimization problem can be computed efficiently, namely in polynomial time, e.g., via linear programming. Several efficient combinatorial algorithms (Ahuja et al, 1993) are also available, whose computational performance is typically significantly better.

The favorable computational properties of the routing and rebalancing algorithm presented in this section enable application to large-scale systems, as described next.

## 5 Numerical Experiments

In this section, we evaluate the validity of the capacity-symmetry assumption for several major U.S. cities (Section 5.1),

**Table 1** Average fractional capacity disparity for several major urban centers in the United States.

| Urban center | Avg. frac. cap. disp. | Std. dev. |
|---|---|---|
| Chicago, IL | $1.2972 \cdot 10^{-4}$ | $1.003 \cdot 10^{-4}$ |
| New York, NY | $1.6556 \cdot 10^{-4}$ | $1.304 \cdot 10^{-4}$ |
| Colorado Springs, CO | $3.1772 \cdot 10^{-4}$ | $2.308 \cdot 10^{-4}$ |
| Los Angeles, CA | $0.9233 \cdot 10^{-4}$ | $0.676 \cdot 10^{-4}$ |
| Mobile, AL | $1.9368 \cdot 10^{-4}$ | $1.452 \cdot 10^{-4}$ |
| Portland, OR | $1.0769 \cdot 10^{-4}$ | $0.778 \cdot 10^{-4}$ |

characterize the effect of rebalancing on congestion in asymmetric networks (Section 5.2), and explore the performance of the algorithm presented in Section 4 on real-world road topologies with real customer demands (Section 5.3).

### 5.1 Capacity Symmetry within Urban Centers in the US

The existential result in Section 3, namely Theorem 2, relies on the assumption that the road network is capacity-symmetric, i.e., for every cut $(\mathcal{S}, \bar{\mathcal{S}})$, $C_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) = C_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}})$. One may wonder whether this assumption is (approximately) met in practice. From an intuitive standpoint, one might argue that transportation networks within urban centers are indeed *designed* to be capacity symmetric, so as to avoid accumulation of traffic flow in some directions. We corroborate this intuition by computing the imbalance between the outbound capacity (i.e., $C_{\text{out}}$) and the inbound capacity (i.e., $C_{\text{in}}$) for 1000 randomly-selected cuts within several urban centers in the United States. For each edge $(u, v) \in \mathcal{E}$, we approximate its capacity as proportional to the product of the speed limit $v_{\max}(u, v)$ on that edge and the number of lanes $L(u, v)$, that is, $c(u, v) \propto v_{\max}(u, v) \cdot L(u, v)$. The road graph $G(\mathcal{V}, \mathcal{E})$, the speed limits, and the number of lanes are obtained from OpenStreetMap data (Haklay and Weber, 2008).

For a cut $(\mathcal{S}, \bar{\mathcal{S}})$, we define its fractional capacity disparity $D(\mathcal{S}, \bar{\mathcal{S}})$ as

$$D(\mathcal{S}, \bar{\mathcal{S}}) := 2 \, \frac{\left| C_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) - C_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}}) \right|}{C_{\text{out}}(\mathcal{S}, \bar{\mathcal{S}}) + C_{\text{in}}(\mathcal{S}, \bar{\mathcal{S}})}.$$

Table 1 shows the average fractional capacity disparity (over 1000 samples) for several US urban centers. As expected, the road networks for such cities appear to posses a very high degree of capacity-symmetry, which validates the symmetry assumption made in Section 3.

### 5.2 Characterization of Congestion due to Rebalancing in Asymmetric Networks

The theoretical results in Section 3 are proven for capacity-symmetric networks, which are in general a reasonable model for typical urban road networks (as shown in the previous section). Nevertheless, it is of interest to characterize the applicability of our theoretical results (chiefly, the existential result in Theorem 2) to road networks that significantly violate the capacity-symmetry property. In other words, we investigate to what degree rebalancing might lead to an increase in congestion if the network is asymmetric.

To this purpose, we compute solutions to the CRRP for road networks with varying degrees of capacity asymmetry and compare the corresponding travel times to those obtained by computing optimal routes in the absence of rebalancing (as would be the case, e.g., if the vehicles were privately owned). We focus on the road network portrayed in Figure 3, derived from OpenStreetMap data (Haklay and Weber, 2008). With 1351 nodes and 3137 edges, the road network captures all major streets and avenues in Manhattan. Transportation requests are based on actual taxi rides in New York City on March 1, 2012 from 6 to 8 p.m. (courtesy of the New York Taxi and Limousine Commission). We clustered all departures and arrivals into 100 stations and considered only origin-destination pairs with more than 5 customers per hour on average (27,571 or 51.1% of all trips). As in Section 5.1, we approximated the capacity of each road as proportional to the product of the speed limit $v_{\max}(u, v)$ and the number of lanes $L(u, v)$. To ensure that the flow induced by the trips would induce a small amount of congestion *before* introducing any asymmetry, we scaled down the capacities of all roads uniformly. Empirically, a scaling factor of 0.041 (or $25\times$ reduction) introduced sufficient congestion, which is consistent with the observations that (i) we only consider $51.1\%$ of true customer flow due to network filtering and (ii) taxis only constitute a fraction of the vehicles in Manhattan.

To investigate the effects of network asymmetry, we introduce an *artificial capacity asymmetry* into the baseline Manhattan road network by progressively reducing the capacity of all northbound avenues. In order to *quantify* the effect rebalancing has on congestion and travel times, we assign slack variables $\delta_C(u, v)$, associated with a cost $c_c(u, v)$, to each congestion constraint (6). The cost $c_c(u, v)$ is selected such that the optimization algorithm selects a congestion-free solution whenever one is available. Once a solution is found, the actual travel time on each (possibly congested) link is computed using the heuristic BPR delay model (Bureau of Public Roads, 1964) presented in Section 4. This approach maintains feasibility even in the congested traffic regime, and hence it allows us to assess the impact of rebalancing on congestion in asymmetric networks.

Table 2 summarizes the results of our simulations. In the baseline case, no artificial capacity asymmetry is introduced, i.e., the fractional capacity reduction of northbound avenues is equal to 0%. Overall, the difference between the travel times in the two cases is very small (approximately 1.16%),

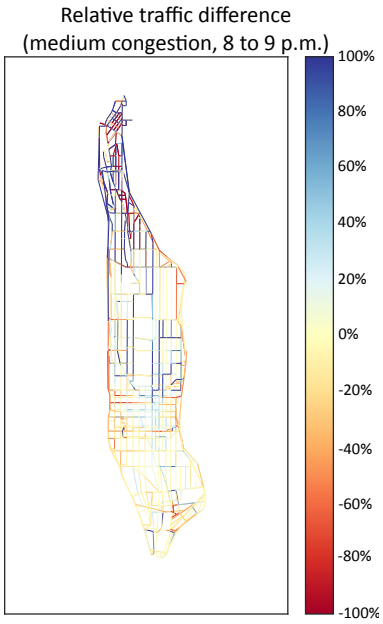


**Fig. 3** Performance of the "real-time congestion-aware rebalancing algorithm" as compared to the baseline algorithm in (Zhang and Pavone, 2016). The color of each road corresponds to the percent *difference* in the number of vehicles traversing it between the congestion-aware and baseline rebalancing algorithms–blue indicating a reduction in congestion using the congestion-aware algorithm.

which is consistent with the fact that New York City's road graph has largely symmetric capacity, as shown in Section 5.1. Interestingly, even with a massive (60%) reduction in northbound capacity, travel times with and without rebalancing vehicles are practically equivalent (within 0.12%). Collectively, these results show that the existential result in Theorem 2, proven under the assumption of a symmetric network, appears to extend (albeit approximately) to asymmetric networks. In particular, it appears that vehicle rebalancing does not lead to an appreciable increase in congestion under very general conditions.

**Table 2** Customer travel times with and without rebalancing for different levels of network asymmetry.

| | Average travel time [s] | | |
|---|---|---|---|
| Cap. reduction | Without reb. | With reb. | Travel time increase |
| 0% | 58.00 | 58.67 | 1.16 % |
| 10% | 58.12 | 59.15 | 1.76 % |
| 20% | 58.49 | 59.67 | 2.02 % |
| 30% | 59.26 | 60.56 | 2.20 % |
| 40% | 60.65 | 61.78 | 1.86 % |
| 50% | 63.66 | 64.55 | 1.40 % |
| 60% | 72.04 | 72.13 | 0.12 % |

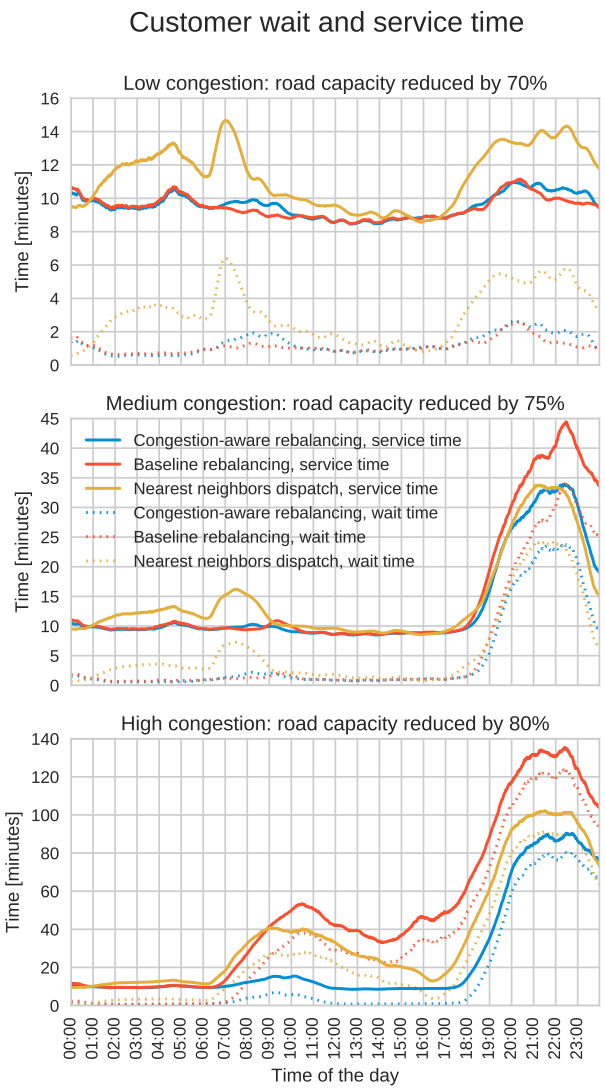


**Fig. 4** Comparison of customer wait and service times from different rebalancing and dispatching algorithms for low, medium, and high levels of congestion. The congestion-aware algorithm recovers the asymptotic behavior of the baseline rebalancing algorithm for low levels of congestion, and it outperforms both the baseline rebalancing algorithm and the nearest-neighbor dispatch algorithm for high levels of congestion.

### 5.3 Congestion-Aware Real-time Rebalancing

In this section we evaluate the performance of the real-time routing and rebalancing algorithm presented in Section 4, and compare it to a baseline approach that does not explicitly take congestion into account. We simulate 8,000 vehicles providing service to approximately 480,000 actual taxi requests over 24 hours on March 1, 2012, using the same Manhattan road network as in the previous section (shown in Figure 3).

We use the MATSim agent-based traffic simulator (Horni et al, 2016) and modify its taxi extension to accommodate

station-based dispatching and rebalancing of idle vehicles[2]. MATSim uses an agent-based, microscopic traffic model where each road is abstracted as a capacitated FIFO queue. Vehicles can enter a road link only if that link has not reached its maximum capacity. Once a vehicle enters a link, it can leave it after (i) the free-flow travel time on the link has elapsed and (ii) it has reached the head of the queue. Other delay factors such as traffic signals, turning times, and pedestrian blocking are not simulated.

Taxi requests are clustered into 100 stations corresponding to subsets of the nodes in the network. The 481,989 trip requests from the same New York Taxi and Limousine Commission data set used in Section 5.2 are simulated using a time step of 1 second.

Three algorithms are evaluated, namely (i) a nearest-neighbor dispatching algorithm that performs no rebalancing of idle vehicles, (ii) the congestion-aware routing and rebalancing algorithm presented in Section 4, and (iii) a baseline rebalancing algorithm. The baseline approach is derived from the real-time rebalancing algorithm presented in (Zhang and Pavone, 2016), which is a point-to-point algorithm that computes rebalancing origins and destinations without considering the underlying road network. In the baseline approach, customer routes are computed in the same way as in Section 4. For rebalancing, the origins and destinations are first solved using the algorithm provided in (Zhang and Pavone, 2016), then the routes are computed using the $A^*$ algorithm much like the customer routes.

In (Zhang et al, 2016), the authors show that, in the low-congestion regime, the baseline algorithm offers near-optimal performance and outperforms several state-of-the-art dispatching and rebalancing algorithms. However, the baseline algorithm ignores the potential for additional congestion induced by rebalancing vehicles, and thus, performs poorly for highly congested networks. On the other hand, this performance penalty is reflected in the nearest-neighbor rebalancing algorithm, causing it to perform much better in high congestion cases where this penalty outweighs the benefit of pre-positioning empty vehicles.

IBM ILOG CPLEX was used to implement the congestion-aware and baseline rebalancing algorithms. The computation time was (on average) under 0.5 s on commodity hardware (Intel Core i7-5960, 64 GB RAM); the maximum computation time was 4.52 s. To account for the computation time, release of the rebalancing routes was delayed by 5 s in the simulation framework.

For each algorithm, we simulated three scenarios corresponding to low, medium and very high levels of congestion (corresponding to a reduction of the road network's nominal capacity of 70%, 75% and 80% respectively). Figure 4 presents a summary of the performance results. Note that the service time represents the total time a customer spends in the system (waiting time plus traveling time).

For low levels of congestion, the performance of the congestion-aware algorithm closely tracks the performance of the baseline rebalancing algorithm. The nearest-neighbor dispatch algorithm performs significantly worse than either rebalancing algorithm in this regime. For medium and high levels of congestion, performance of the baseline algorithm is significantly degraded: as expected, rebalancing trips cause significant congestion in the network, as exemplified in Figure 3. The nearest-neighbor dispatch algorithm offers better performance than the baseline rebalancing algorithm in this regime: when the road network is congested, not rebalancing at all appears to be preferable to excessive rebalancing.

The congestion-aware algorithm recovers the performance of the baseline rebalancing algorithm in the low-congestion regime and the performance of the nearest-neighbor dispatcher in the medium-congestion regime; in the high-congestion regime, it outperforms both. By selectively rebalancing vehicles where and when congestion is low, and by selecting rebalancing routes that do not increase congestion, the algorithm is able to mediate between the behavior of the baseline rebalancing algorithm and the behavior of the nearest-neighbor algorithm depending on the level of congestion: this results in good performance both in terms of network congestion and in terms of customer service times across a wide range of congestion regimes.

## 6 Conclusions and Future Work

In this paper we presented a network flow model of an autonomous mobility-on-demand system on a capacitated road network. We formulated the routing and rebalancing problem and showed that on symmetric road networks, it is always possible to route rebalancing vehicles in a coordinated way that does not increase traffic congestion. Using a model road network of Manhattan, we showed that rebalancing did not increase congestion even for moderate degrees of network asymmetry. We leveraged the theoretical insights to develop a computationally efficient real-time congestion-aware routing and rebalancing algorithm and demonstrated its performance over state-of-the-art point-to-point rebalancing algorithms through simulation. This highlighted the importance of congestion awareness in the design and implementation of control strategies for a fleet of self-driving vehicles.

This work opens the field to many future avenues of research. First, it is of interest to directly use the solution to the integral CRRP as a practical real-time routing algorithm to compute congestion-free routes for customer vehicles and rebalancing vehicles alike. While the integral CRRP is in general intractable, randomized algorithms (Raghavan and Tompson, 1987; Srinivasan, 1999) may be used to compute high-quality approximate solutions for large-scale sys-

[2] The source code for the modified taxi extension is available at `https://github.com/StanfordASL/matsim-AMoD/`

tems. Second, from a modeling perspective, we would like to study the inclusion of stochastic information (e.g., demand prediction, travel time uncertainty) for the routing and rebalancing problem, as well as a richer set of performance metrics and constraints (e.g., time windows to pick up customers). Third, it is worthwhile to study how our results give intuition into business models for autonomous urban mobility (e.g. fleet sizes). Fourth, it is of interest to explore other approaches that may reduce congestion, including ride-sharing, demand staggering, and integration with public transit to create an intermodal transportation network. Fifth, we plan to study the deployment of large *electric* AMoD fleets and, in particular, to characterize the interaction between such fleets and the electric power network. Sixth, we would like to explore decentralized architectures for cooperative routing and rebalancing. Finally, we would like to demonstrate the real-world performance of the algorithms by implementing them on real fleets of self-driving vehicles.

**Acknowledgements** The authors would like to thank Zachary Sunberg for his analysis on the road network symmetry of U.S. cities.

## References

Ahuja RK, Magnanti T, Orlin J (1993) Network Flows: Theory, Algorithms and Applications. Prentice Hall

Balmer M, Rieser M, Meister K, Charypar D, Lefebvre N, Nagel K (2009) MATSim-t: Architecture and simulation times. In: Multi-Agent Systems for Traffic and Transportation Engineering, chap 3, pp 57–78

Barnard M (2016) Autonomous cars likely to increase congestion. Available at `http://cleantechnica.com/2016/01/17/autonomous-cars-likely-increase-congestion`

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

Bureau of Public Roads (1964) Traffic assignment manual. Tech. rep., U.S. Department of Commerce, Urban Planning Division

Daganzo CF (1994) The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological 28(4):269–287

Even S, Itai A, Shamir A (1976) On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing 5(4):691–703

Fagnant DJ, Kockelman KM (2014) The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transportation Research Part C: Emerging Technologies 40:1–13

Ford LR, Fulkerson D (1962) Flows in Networks. Princeton University Press

Goldberg A, Oldham J, Plotkin S, Stein C (1998) An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow. In: Int. Conf. on Integer Programming and Combinatorial Optimization, pp 338–352

Goldberg AV, Tardos E, Tarjan RE (1990) Network flow algorithms. In: Algorithms and Combinatorics. Volume 9: Paths, Flows, and VLSI-Layout

Haklay M, Weber P (2008) OpenStreetMap: User-generated street maps. IEEE Pervasive Computing 7(4):12–18

Hart P, Nilsson N, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems, Science, & Cybernetics 4(2):100–107

Horni A, Nagel K, Axhausen KW (eds) (2016) The Multi-Agent Transport Simulation MATSim. Ubiquity Press

Janson BN (1991) Dynamic traffic assignment for urban road networks. Transportation Research Part B: Methodological 25(2–3):143–161

Karp RM (1975) On the computational complexity of combinatorial problems. Networks 5(1):45–68

Kerner BS (2009) Introduction to modern traffic flow theory and control: the long road to three-phase traffic theory, 1st edn. Springer Berlin Heidelberg

Le T, Kovács P, Walton N, Vu HL, Andrew LLH, Hoogendoorn SSP (2015) Decentralized signal control for urban road networks. Transportation Research Part C: Emerging Technologies 58:431–450

Leighton T, Makedon F, Plotkin S, Stein C, Tardos É, Tragoudas S (1995) Fast approximation algorithms for multicommodity flow problems. Journal of Computer and System Sciences 50(2):228–243

Lighthill MJ, Whitham GB (1955) On kinematic waves. II. a theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 229(1178):317–345

Mitchell WJ, Borroni-Bird CE, Burns LD (2010) Reinventing the automobile: Personal urban mobility for the 21st century. MIT Press

Neuburger H (1971) The economics of heavily congested roads. Transportation Research 5(4):283–293

Osorio C, Bierlaire M (2009) An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. European Journal of Operational Research 196(3):996–1007

Papageorgiou M, Hadj-Salem H, Blosseville JM (1991) ALINEA: A local feedback control law for on-ramp metering. Transportation Research Record: Journal of the Transportation Research Board (1320):58–64

Pavone M, Smith SL, Frazzoli E, Rus D (2012) Robotic load balancing for Mobility-on-Demand systems. Int Journal of Robotics Research 31(7):839–854

Peeta S, Mahmassani HS (1995) System optimal and user equilibrium time-dependent traffic assignment in congested networks. Annals of Operations Research 60(1):81–113

Pérez J, Seco F, Milanés V, Jiménez A, Díaz JC, De Pedro T (2010) An RFID-based intelligent vehicle speed controller using active traffic signals. Sensors 10(6):5872–5887

Raghavan P, Tompson CD (1987) Randomized rounding: A technique for provably good algorithms and algorithmic proofs. Combinatorica 7(4):365–374

Seow KT, Dang NH, Lee DH (2010) A collaborative multiagent taxi-dispatch system. IEEE Transactions on Automation Sciences and Engineering 7(3):607–616

Spieser K, Treleaven K, Zhang R, Frazzoli E, Morton D, Pavone M (2014) Toward a systematic approach to the design and evaluation of Autonomous Mobility-on-Demand systems: A case study in singapore. In: Road Vehicle Automation, Springer

Srinivasan A (1999) A survey of the role of multicommodity flow and randomization in network design and routing. In: Randomization Methods in Algorithm Design, vol 43, pp 271–302

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

Treiber M, Hennecke A, Helbing D (2000) Microscopic simulation of congested traffic. In: Traffic and Granular Flow '99, Springer Berlin Heidelberg, pp 365–376

Treleaven K, Pavone M, Frazzoli E (2011) An asymptotically optimal algorithm for pickup and delivery problems. In: Proc. IEEE Conf. on Decision and Control, Orlando, Florida

Treleaven K, Pavone M, Frazzoli E (2012) Models and efficient algorithms for pickup and delivery problems on roadmaps. In: Proc. IEEE Conf. on Decision and Control, Maui, Hawaii

Treleaven K, Pavone M, Frazzoli E (2013) Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems. IEEE Transactions on Automatic Control 58(9):2261–2276

Urmson C (2014) Just press go: Designing a self-driving vehicle. Available at `http://googleblog.blogspot.com/2014/05/just-press-go-designing-self-driving.html`

Wardrop JG (1952) Some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers 1(3):325–362

Wilkie D, van den Berg JP, Lin MC, Manocha D (2011) Self-aware traffic route planning. In: Proc. AAAI Conf. on Artificial Intelligence

Wilkie D, Baykal C, Lin MC (2014) Participatory route planning. ACM

Xiao N, Frazzoli E, Luo Y, Li Y, Wang Y, Wang D (2015) Throughput optimality of extended back-pressure traffic signal control algorithm. In: Mediterranean Conf. on Control and Automation

Yang Q, Koutsopoulos HN (1996) A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transportation Research Part C: Emerging Technologies 4(3):113–129

Zhang R, Pavone M (2016) Control of robotic Mobility-on-Demand systems: A queueing-theoretical perspective. Int Journal of Robotics Research 35(1-3):186–203

Zhang R, Rossi F, Pavone M (2016) Model predictive control of Autonomous Mobility-on-Demand systems. In: Proc. IEEE Conf. on Robotics and Automation, Stockholm, Sweden