Routing and Rebalancing Intermodal Autonomous Mobility-on-Demand Systems in Mixed Traffic

DOI:
10.1109/TITS.2021.3112106

Document status and date:
Published: 01/08/2022

Document Version:
Author's version before peer-review

Routing and Rebalancing Intermodal Autonomous Mobility-on-Demand Systems in Mixed Traffic

Salomón Wollenstein-Betech[1], Mauro Salazar[2], Arian Houshmand[1],
Marco Pavone[3], Ioannis Ch. Paschalidis[1], and Christos G. Cassandras[1].

***Abstract*— This paper studies congestion-aware route-planning policies for intermodal Autonomous Mobility-on-Demand (AMoD) systems, whereby a fleet of autonomous vehicles provides on-demand mobility jointly with public transit under mixed traffic conditions (consisting of AMoD and private vehicles). Specifically, we first devise a network flow model to jointly optimize the AMoD routing and rebalancing strategies in a congestion-aware fashion by accounting for the endogenous impact of AMoD flows on travel time. Second, we capture the effect of exogenous traffic stemming from private vehicles adapting to the AMoD flows in a user-centric fashion by leveraging a sequential approach. Since our results are in terms of link flows, we then provide algorithms to retrieve the explicit recommended routes to users. Finally, we showcase our framework with two case-studies considering the transportation sub-networks in Eastern Massachusetts and New York City, respectively. Our results suggest that for high levels of demand, pure AMoD travel can be detrimental due to the additional traffic stemming from its rebalancing flows. However, combining AMoD with public transit, walking and micromobility options can significantly improve the overall system performance.**

***Index Terms*—Mobility-on-Demand, System-Optimal Routing, Rebalancing, Mixed Autonomy.**

I. Introduction

In the last century, urban mobility has been dominated by the use of *private* vehicles. The success of this mode of transport relies on its fast and convenient point-to-point transportation service. However, even if this technology has been widely adopted, it has been recently criticized due to its dependency on gasoline, its harmful emissions to the environment, its underutilization (according to [1], private vehicles are parked for more than 95% of the time), its impact on traffic congestion, and its land and infrastructure requirements for wider roads and parking spaces. Hence, some have acknowledged that private vehicles are an unsustainable solution for urban mobility [2]. As we think and plan for the cities of the future, mobility-on-demand (MoD), or Autonomous Mobility-on-Demand (AMoD) systems enabled by autonomous vehicles, offer a new way to provide a comparable fast and comfortable point-to-point service while maintaining low congestion levels. As defined by the Federal Transit Administration of the United States, a MoD system is a "multimodal, integrated, automated, accessible, and connected transportation system with personalized mobility at its core which uses on-demand information, real-time data, and predictive analysis to provide travelers with transportation choices that best serve their needs and circumstances" [3].

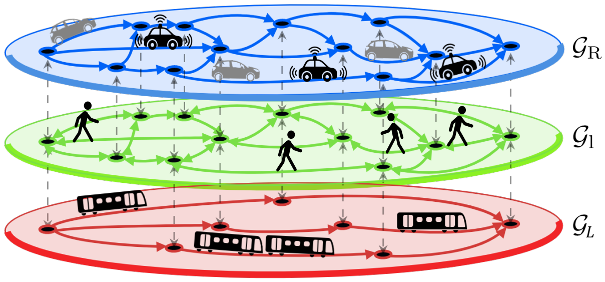


Fig. 1. I-AMoD network (supergraph) consisting of three layers: the road network (blue with black AMoD cars and grey private vehicles, respectively), walking pathways (green) and subway lines (red); the dashed arrows represent switching arcs.

In this paper we focus on methodologies that optimize the operations of AMoD systems with the goal of reducing traffic congestion. To achieve this, we develop a coordinated intermodal routing procedure that seeks to minimize the overall commuters travel time while ensuring that all travelers are being served by the same platform. In particular, we study the *routing* and *load-balancing* processes of a fleet of vehicles belonging to an AMoD service when they interact with self-interested vehicles in the network. In contrast with today's platforms (e.g., Uber, Lyft, DiDi), our objective is to take these two decisions *jointly* rather than separately. If the vehicles belonging to the fleet are controlled by self-interested humans, this joint optimization is harder to carry out as one would have to design compensation schemes in order to steer the selfish behavior to a system-optimal solution [4], [5]. However, as the level of automation of these platforms increases, with many already testing their Connected and Automated Vehicles (CAVs) in our streets [6], [7], thinking whether these processes should be addressed jointly becomes relevant.

Literature review: We first review techniques to solve the routing and load-balancing (also referred to as *rebalancing*) problems individually and then focus on the joint problem.

Current drivers in MoD platforms, such as Uber, Lyft or DiDi, choose their paths by using routing apps (e.g., Waze and Google Maps). These apps recommend routes

This research was supported in part by NSF under grants ECCS-1509084, DMS-1664644, CNS-1645681, IIS-1914792, and CMMI-1454737, by AFOSR under grant FA9550-19-1-0158, by ARPA-E under grant DE-AR0001282, by the NEXTCAR program under grant DE-AR0000796, by the MathWorks, by the ONR under grant N00014-19-1-2571, by the NIH under grant 1R01GM135930, and by the Toyota Research Institute (TRI). This article solely reflects the opinions and conclusions of its authors and not NSF, TRI, or any other entity.

[1] The authors are with the Division of Systems Engineering and the Center for Information and Systems Engineering, Boston University, Boston, MA 02215 USA {salomonw, arianhm, cgc, yannisp}@bu.edu

[2] The author is with the department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, NL m.r.u.salazar@tue.nl

[3] The author is with the department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94325 USA pavone@stanford.edu

using traditional shortest path algorithms such as Dijkstra's [8], Bellman-Ford [9], and incremental graph [10] that find provable optimal routes. Also, widely employed are heuristics such as A-star [11], tabu search [12] and genetic algorithms [13] given that they provide a balance between solution quality and computation time. This *User-Centric* (UC) approach to routing, in which every driver minimizes its own travel time, is suboptimal compared to *System-Optimal* (SO) routing schemes achievable when vehicles are coordinated by a central controller. The bounds on the inefficiencies of the UC solution compared to the SO have been studied in [14], showing that for linear travel time functions, the cost of the UC is bounded by $4/3$ the cost of the SO solution. The gap between UC and SO is commonly known as the *Price of Anarchy* and has been studied in [15] . SO routing algorithms have been studied and it has been established that mild modifications to the UC Traffic Assignment Problem (TAP) can solve the SO [16]. Therefore using algorithms to solve the TAP such as the Method of Successive Averages [16], Frank-Wolfe [17]–[19], or the Traffic Assignment by Paired Alternative Segments (TAPAS) [20] is sufficient to solve a SO problem. A relevant feature of SO routing is its fairness implications (users taking longer routes than their shortest route), which has been studied in [21] where an optimization algorithm termed Partran (a revised version of the Frank-Wolfe method) was proposed. In a mixed traffic setting, the interaction between a fleet of CAVs using SO routing coupled with reactive UC private vehicles has been investigated theoretically by [22], where a reduction in headways is considered thanks to adaptive cruise control technology included in CAVs. However, this analysis requires a network configuration of parallel links and it is not suitable for general transportation networks. To overcome this, [23]–[25] propose an iterative approach to find a solution between these two classes of vehicles known as *diagonalization scheme*. In [23] the authors show that both CAVs and private vehicles can achieve better performance in terms of travel time and energy savings as the percentage of CAVs in the network increases. However, neither of these approaches addresses the rebalancing of CAVs nor does it consider the possibility of intermodal (or multimodal) routing. Both limitations are addressed in this paper.

Aside from routing, rebalancing is tackled in practice by providing drivers with a real-time heat-map of users' demand such that the driver is incentivized to relocate to an area that will maximize its profits. Rebalancing has been studied by researchers using *proactive* (or *planning*) strategies that redistribute the fleet across regions in order to meet a forecasted demand.[1] Using this perspective, [29] shows that rebalancing is necessary to avoid building unbounded customer queues and to stabilize the system. [29] proposes a rebalancing policy that minimizes the empty vehicle travel time under static (steady-state) conditions using a fluidic model. Furthermore, [30] proposed a queueing-theoretical approach to account for customers leaving the system when their waiting times are long. This method solves a Linear Program (LP) recursively that balances the fleet availability across the regions. Moreover, the authors show Pareto optimal curves relating desired quality of service and fleet size. Similarly, [31] proposed a method that minimizes the number of customer dropouts instead of the empty driven miles to focus on service quality. Different than these queueing models, simulation-based methods have also been employed [32]–[34].

More recently, schemes that consider the effects of rebalancing in routing and congestion have been analyzed. Threshold approximations of the travel time function have been used to study congestion effects [35], sometimes capturing the interaction with public transit [36], [37] or with the power-grid [38]. These threshold schemes work as binary decisions allowing or rejecting the use of a road depending on whether the flow has exceeded the threshold or not, but do not capture different travel times for different flow levels on each link. To account for flow-based routing schemes most work leverages the classical Bureau of Public Roads (BPR) congestion model [39] together network optimization methods. In particular, [40] provides a Frank-Wolfe algorithm, where dummy nodes are added to the transportation network to account for rebalancing flows and where the BPR function is evaluated when designing routes. However, this approach cannot include other modes of transportation such as walking, micromobility options, or public transit. Against this backdrop, a piecewise-affine approximation of the travel time function is introduced in [41] which converts the joint problem to a quadratic program. In this work, we extend this approximation in order to account for more accurate, fast and implementable models.

Statement of contributions: The contribution of this paper is threefold. First, we present a method to optimize intermodal congestion-aware routing and rebalancing policies of an AMoD service. The objective is to improve the quality of service by jointly reducing the overall user travel time while ensuring vehicle availability in every region. We allow AMoD users to use multiple modes of transportation such as public transportation, walking or micromobility (e.g., bikes and e-scooters) in order to reduce the overall travel time. To solve the routing and rebalancing problem, we approximate the non-linear travel latency function with a piecewise-affine function. This slight modification allows us to write the problem as a tractable quadratic program and later to a relaxed linear program, making it easier and faster to solve. We prove that this approximation is asymptotically optimal in the number of segments defining the piecewise function and we leverage origin-based formulations of the problem to improve the computational efficiency. Second, we extend the joint formulation to a mixed traffic setting capturing the interaction between AMoD users and private vehicle and providing routing decisions for the AMoD users that anticipate the behavior of the private vehicles. To do so, we leverage a sequential method that finds a steady-state solution for these two user types. Third, given that the proposed methods retrieve solutions expressed in terms of traffic flows, we propose distributed algorithms to convert the flows to viable routes, enabling real-time route recommendations for the AMoD users. Finally, we present experiments to (i) empirically show the asymptotic behavior of the approximated model; (ii) capture the trade-off between the benefits of SO routing and the excess flow due to rebalancing; (iii) observe the effect of intermodality on travel

[1] Note that this process finds good coverage of vehicles over regions of the system and it is not focused on *matching* or *assigning* vehicles to customers. This vehicle-passenger *assignment* has been studied in [26]–[28]

times; and (iv) show the applicability of the route-recovery strategies.

Building on the model and preliminary analysis in [42], this paper provides the following contributions: (i) extension of the approximated model from 3-segment to an n-segment model; (ii) new theoretical and empirical results on the asymptotic behavior of the approximated model; (iii) introduction of an origin-based formulation to improve the computational efficiency; (iv) development of route-recovery algorithms from the flow-based solution; (v) additional experimental results including a case study on a larger network of New York City incorporating the subway system as an intermodal option.

Organization: The rest of the paper is organized as follows: In Section II we present the models used and the problem formulation. In Section III we develop the piecewise-affine approximation formulation along with the main analytical results of this paper. In Sections IV and V, we provide a framework for the mixed traffic problem and route-recovering strategies, respectively. Finally, we present experiments using the Eastern Massachusetts and New York City transportation networks in Section VI and we conclude in Section VII.

II. Problem Formulation

Consider an AMoD system which provides a mobility service through multiple modes of transportation (e.g., autonomous taxi-rides, walking and mass transit). To model the system, let $\mathcal{G}$ be a network composed by the road layer and L additional layers, each representing a different transportation mode (Fig. 1). We denote by $\mathcal{G}_\mathrm{R} = (\mathcal{V}_\mathrm{R}, \mathcal{A}_\mathrm{R})$ the road layer and by $\mathcal{G}_\mathrm{l} = (\mathcal{V}_\mathrm{l}, \mathcal{A}_\mathrm{l})$, for $l = 1, \ldots, L$, the other layers where $(\mathcal{V}_\mathrm{R}, \mathcal{A}_\mathrm{R})$ and $(\mathcal{V}_{\mathrm{l_i}}, \mathcal{A}_{\mathrm{l_i}})$ are the sets of vertices and arcs for each layer. Then, the supergraph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is composed of all layers and a set of *switching* arcs, denoted by $\mathcal{A}_\mathrm{S}$, that connect the network layers to allow AMoD users to switch modes (see dotted lines in Fig. 1). Formally $\mathcal{G}$ is composed of the set of vertices $\mathcal{V} = \mathcal{V}_\mathrm{R} \cup \mathcal{V}_1 \cup \ldots \cup \mathcal{V}_L$ and arcs $\mathcal{A} = \mathcal{A}_\mathrm{R} \cup \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_L \cup \mathcal{A}_\mathrm{S}$.

In order to model the demanded trips, let $\mathbf{w} = (w_s, w_t)$ denote an Origin-Destination (OD) pair and $d_\mathbf{w} \geq 0$ the demand rate at which customers request service per unit time from origin w_s to destination w_t. Let W be the total number of OD pairs and $\mathcal{W} = \{\mathbf{w}_k : \mathbf{w}_k = (w_{sk}, w_{tk}), k = 1, \ldots, W\}$ the set of OD pairs. Let a vectorized version of the demand be $\mathbf{g} = (d_{\mathbf{w}_k}; k = 1, \ldots, W)$.

To keep track of the AMoD user flow on a link, we let $x_{ij}^\mathbf{w}$ denote the AMoD flow induced by OD pair $\mathbf{w}$ on link $(i,j) \in \mathcal{A}$. Given that the AMoD system needs to rebalance its vehicles to ensure service, we let x_{ij}^r be the *rebalancing flow* on link $(i,j) \in \mathcal{A}_\mathrm{R}$. Finally, to consider the interaction between the AMoD provider and the other (private) vehicles, we let x_{ij}^p be the self-interested *private vehicle* flow on $(i,j) \in \mathcal{A}_\mathrm{R}$. We use the term "private" as we assume that self-interested users must arrive at their destination with their vehicle and do not have the option of switching transportation mode since they have a parking constraint. To simplify notation, we let the AMoD user flow on any link to be

$$x_{ij}^u = \sum_{\mathbf{w} \in \mathcal{W}} x_{ij}^\mathbf{w}, \qquad \forall (i,j) \in \mathcal{A}, \tag{1}$$

and the total flow on a link to be

$$x_{ij} = x_{ij}^u + x_{ij}^r + x_{ij}^p, \qquad \forall (i,j) \in \mathcal{A}. \tag{2}$$

Note that neither rebalancing flow $\mathbf{x}^r$ nor private vehicle flow $\mathbf{x}^p$ exist on layers $l = 1, \ldots, L$ nor on the switching links in Fig. 1. Hence, for those links we set $x_{ij}^r = x_{ij}^p = 0$ for all $(i,j) \in \mathcal{A} \setminus \mathcal{A}_\mathrm{R}$.

We specify the time it takes to cross link (i,j) as $t_{ij}(x) : \mathbb{R}_+^{|\mathcal{A}|} \mapsto \mathbb{R}_+$. Using the same structure as in [43], we characterize t_{ij} as a *travel time* function that maps the flow x_{ij} on a link to a travel time as follows:

$$t_{ij}(x_{ij}) = t_{ij}^0 f(x_{ij}/m_{ij}), \tag{3}$$

where m_{ij} is the link capacity, t_{ij}^0 is the free-flow travel time on link (i,j), and $f(\cdot)$ is a strictly increasing, positive, and continuously differentiable function. To ensure that if there is no flow on the link the travel time is equal to the free-flow travel time, we consider functions with $f(0) = 1$. These functions are typically increasing polynomials that are hard to estimate [44]. Despite this, a widely used function by transportation engineers is the *Bureau of Public Roads (BPR)* function [39] denoted by

$$t_{ij}(x_{ij}) = t_{ij}^0 (1 + \alpha (x_{ij}/m_{ij})^\beta). \tag{4}$$

where typically $\alpha = 0.15$ and $\beta = 4$. For a discussion on how to estimate these functions see [45].

Throughout this paper, we will use this function to decide the routes of AMoD users and private vehicles, given the network flow levels. However, our analysis allows for any strictly increasing travel time function. For the L layers (excluding the road layer) we consider a constant travel time (independent of the flow) on every link.

A. System-Optimal Routing and Rebalancing of AMoD Systems

Recall that our goal is to find the system-optimal congestion-aware routes and rebalancing policies of an AMoD provider. Let $d_\mathbf{w}^u$ be customer rate requests to the AMoD system for passengers traveling from w_s to w_t, and $\mathbb{1}_{i=j}$ be the indicator function equal to 1 when $i = j$ and 0 otherwise. The problem we aim to solve is then expressed by

$$\min_{\mathbf{x}^\mathcal{W}, \mathbf{x}^r} \quad J(\mathbf{x}) := \sum_{(i,j) \in \mathcal{A}} t_{ij}(x_{ij}) x_{ij}^u + \sum_{(i,j) \in \mathcal{A}_\mathrm{R}} c_{ij} x_{ij}^r \tag{5a}$$

$$\text{s.t.} \quad \sum_{i:(i,j) \in \mathcal{A}} x_{ij}^\mathbf{w} + \mathbb{1}_{j=w_s} d_\mathbf{w}^u = \sum_{k:(j,k) \in \mathcal{A}} x_{jk}^\mathbf{w} + \mathbb{1}_{j=w_t} d_\mathbf{w}^u, \quad \forall \mathbf{w} \in \mathcal{W}, j \in \mathcal{V}, \tag{5b}$$

$$\sum_{i:(i,j) \in \mathcal{A}_\mathrm{R}} \left(x_{ij}^r + x_{ij}^u \right) = \sum_{k:(j,k) \in \mathcal{A}_\mathrm{R}} \left(x_{jk}^r + x_{jk}^u \right), \ \forall j \in \mathcal{V}_\mathrm{R}, \tag{5c}$$

$$\mathbf{x}^\mathcal{W}, \mathbf{x}^r \geq 0, \tag{5d}$$

where we use bold notation $\mathbf{x}$ to represent a vector containing all the elements of x_{ij}. The dimensions of the decision vectors $\mathbf{x}^r$ and $\mathbf{x}^\mathcal{W}$ are given by $\mathbf{x}^r \in \mathbb{R}^{|\mathcal{A}_\mathrm{R}|}$, and $\mathbf{x}^\mathcal{W} = \{\mathbf{x}^\mathbf{w} \in \mathbb{R}^{|\mathcal{A}|} \mid \mathbf{w} \in \mathcal{W}\}$. Constraints (5b) take care of flow conservation and demand compliance as in a multi-commodity transportation setting. Constraints (5c) ensure the rebalancing of the AMoD fleet (only on the road network). The last sets of constraints (5d) restrict the flows to non-negative values.

The objective $J(\mathbf{x})$ is composed of two terms. The first term considers the total travel time of AMoD users. This evaluates the travel time function $t_{ij}(x_{ij})$ with respect to the total flow given by (2) which includes variables corresponding to private vehicle flow x^p_{ij} (assumed to be fixed), and the rebalancing flow x^r_{ij}. When taking the product $t_{ij}(x_{ij})x^u_{ij}$, we obtain a non-convex function which makes the problem hard to solve. To address this issue, we use a piecewise-affine approximation of $t_{ij}(x_{ij})$ which is further developed in Section III. The second term acts as a linear regularizer whose purpose is to penalize rebalancing flows. This will ensure that a cost for rebalancing of the fleet is taken into account in the optimization problem. We use $c_{ij} = \lambda t^0_{ij}$ where λ is a constant. Therefore, we use a small λ to guide the rebalancing flow through good paths, without dominating the AMoD user routing decisions. Note that, if normalization is needed to ensure a good regularization parameter, we can always bound each component on (5a) using the link capacities an a large enough value for $t(\cdot)$.

B. Private Vehicles Flow Modeling

Aiming to understand the interaction between a SO AMoD fleet and private vehicles, we assume some user-choice model behind private vehicle decisions. To do so we use the *User-Centric* (UC) routing as in the Traffic Assignment Problem (TAP) [16]. Given OD demands, this model finds the flows in the network which achieve a Wardrop equilibrium [46]. This is equivalent to each private user deciding to take the route that minimizes their own travel time. In addition to this, we impose that private vehicles can travel exclusively through the road network $\mathcal{G}_\mathrm{R}$ as opposed to in the full network $\mathcal{G}$. Let $x^{p,\mathbf{w}}_{ij}$ be the flow on link (i,j) induced by private vehicle demand $d^p_\mathbf{w}$ of OD pair $\mathbf{w}$. Then, we assume private vehicles decide their routes by using the UC approach. This is equivalent to solving the following (see more details of the derivation of this model in [16])

$$\min_{\mathbf{x}^p} \quad \sum_{(i,j)\in\mathcal{A}_\mathrm{R}} \int\limits_{x^u_{ij}+x^r_{ij}}^{x_{ij}} t_{ij}(s)ds \tag{6a}$$

$$\text{s.t} \sum_{i:(i,j)\in\mathcal{A}_\mathrm{R}} x^{p,\mathbf{w}}_{ij} + d^p_\mathbf{w}\mathbb{1}_{j=w_s} = \sum_{k:(j,k)\in\mathcal{A}_\mathrm{R}} x^{p,\mathbf{w}}_{jk} + d^p_\mathbf{w}\mathbb{1}_{j=w_t}, \quad \forall \mathbf{w}\in\mathcal{W}, j\in\mathcal{V}_\mathrm{R}, \tag{6b}$$

$$\mathbf{x}^{p,\mathbf{w}} \geq \mathbf{0}. \tag{6c}$$

Notice that this version of the UC TAP is slightly different from the classic one [16] since it considers the AMoD flow in its objective (see limits of the integral on (6a)). To solve this problem we assume that the AMoD flow is fixed and private vehicles plan their routes considering AMoD flows as exogenous. By assuming this, we can use the *Frank-Wolfe* algorithm [17] to solve (6). Let us use the shorthand notation of $\mathtt{TAP}(\mathbf{g},\mathbf{x}^e)$ to indicate the solution of (6) with $\mathbf{x}^e$ equal to any generic exogenous flow. Hence $\mathbf{x}^p \in \mathtt{TAP}(\mathbf{g}^p, \mathbf{x}^u + \mathbf{x}^r)$.

C. AMoD in Mixed Traffic

Critically, AMoD flows react to the decisions made by private vehicles and these, in turn, react to private vehicles' flows. Hence, whenever private vehicles make their routing decisions, the AMoD fleet adjusts theirs, and vice versa. This creates a nested optimization problem between these two classes of vehicles. To give a formal definition of this game-theoretical problem we use the following bilevel optimization formulation

$$\min_{\{\mathbf{x^w}\}_{\mathbf{w}\in\mathcal{W}},\mathbf{x}^r,\mathbf{x}^p} \; J(\mathbf{x}) \tag{7a}$$

$$\begin{aligned} \text{s.t.} \quad & (5\text{b}) - (5\text{d}), \\ & \mathbf{x}^p \in \mathtt{TAP}(\mathbf{g}^p, \mathbf{x}^u + \mathbf{x}^r), \end{aligned} \tag{7b}$$

which has the same structure as (5) with the additional constraint (7b). The latter constraint refers to the TAP (the lower-level problem), which depends on the solution of the full problem (upper-level). Note that the upper-level problem is minimizing over the AMoD users, rebalancing, and privately-owned vehicle flows. This phenomenon has been identified and is often described as a *Stackelberg game* framework [47]. In this setting, there is a *leader* agent (in our case the AMoD manager) and a *follower* (the private vehicles). In the context of transportation networks, [47] derived sufficient conditions to solve this problem when the network has parallel links. Although these models enable for a better understanding of the phenomenon, they are not applicable to general networks and one can hardly assess mixed traffic routing in realistic networks. To address this limitation, we leverage the iterative approach in [23] to compute the private vehicles' and AMoD flows. The formal convergence of this sequential method is not studied in this paper.

III. AMoD Routing and Rebalancing Problem

The problem of routing and rebalancing as stated in (5) is non-convex for typical travel time functions such as the BPR. This happens due to the term $t(x_{ij})x^r_{ij}$ in the objective function. To address this issue, we approximate the travel time function with a piecewise-affine function.

A. Piecewise-affine Approximation (CARSn)

Let the function approximating $t(x)$ be of the form:

$$\hat{t}_{ij}(x) = \begin{cases} t^0_{ij}\Big(1 + a_1\dfrac{(x-\theta^{(1)}_{ij})}{m_{ij}}\Big), & \text{if } \theta^{(1)}_{ij} \leq x \leq \theta^{(2)}_{ij} \\ \quad\vdots & \\ t^0_{ij}\Big(1 + \displaystyle\sum_{l=1}^{n}\Big(\frac{a_l(\theta^{(l)}_{ij}-\theta^{(l-1)}_{ij})}{m_{ij}}\Big) + \frac{a_n(x-\theta^{(n)}_{ij})}{m_{ij}}\Big), & \\ & \text{if } \theta^{(n)}_{ij} \leq x, \end{cases}$$

where a_l is the slope of segment $l = 1,\dots,n$ of $\hat{t}$ with $a_1 \leq \dots \leq a_n < \infty$, and $\theta^{(l)}_{ij}$ is a threshold dividing segments on the travel time function for link (i,j). In our case, we let $\theta^{(l)}_{ij} = \theta^{(l)}m_{ij}$ where $\theta^{(l)}$ is the normalized threshold in the travel time and capacity normalized function depicted in Fig. 2.

To model this piecewise-affine function in the optimization problem, we introduce the following set of slack variables

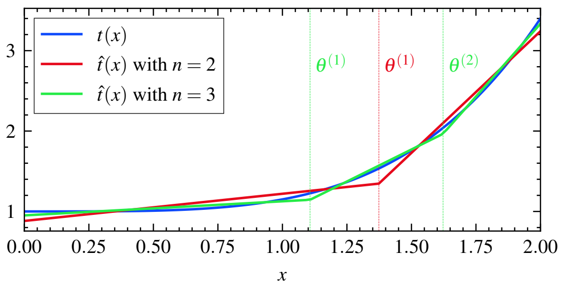


Fig. 2. Travel time function approximation

$$\varepsilon_{ij}^{(n)} = \max\{0, x_{ij} - \theta_{ij}^{(n)}\}, \tag{9a}$$
$$\vdots$$
$$\varepsilon_{ij}^{(k)} = \max\{0, x_{ij} - \theta_{ij}^{(k)} - \sum_{l=k+1}^{n} \varepsilon_{ij}^{(l)}\}, \tag{9b}$$
$$\vdots$$
$$\varepsilon_{ij}^{(0)} = \max\{0, x_{ij} - \sum_{l=1}^{n} \varepsilon_{ij}^{(l)}\}, \tag{9c}$$

where each $\varepsilon_{ij}^{(k)}$ denotes the extra flow exceeding threshold $\theta_{ij}^{(k)}$ and up to $\theta_{ij}^{(k+1)} - \theta_{ij}^{(k)}$, thus, $\varepsilon_{ij}^{(k)} \in [0, \theta_{ij}^{(k+1)} - \theta_{ij}^{(k)}]$. We include these variables in the problem by adding the linear constraints $\varepsilon_{ij}^{(k)} \geq 0$ and $\varepsilon_{ij}^{(k)} \geq \theta_{ij}^{(k)} - \sum_{l=k+1}^{n} \varepsilon_{ij}^{(l)}$, provided that the objective is a function of $\varepsilon_{ij}^{(k)}$.

Using these definitions we can generate a tractable cost function. We focus our attention on an element-wise analysis of the first term (non-convex part) of objective (5a) using $\hat{t}$ instead of t for which we obtain the objective function

$$\hat{J}(x_{ij}, \boldsymbol{\varepsilon}_{ij}) := t_{ij}^{0}\Big(x_{ij}^{u} + \sum_{l=1}^{n}(a_l \varepsilon_{ij}^{(l)}/m_{ij})\Big)\Big(\sum_{k=1}^{n}\varepsilon_{ij}^{(k)} - x^{c}\Big), \tag{10}$$

which is derived as follows:

$$\hat{t}_{ij}(x_{ij})x_{ij}^{u} = t_{ij}^{0}\Big(x_{ij}^{u} + \sum_{l=1}^{n}(a_l \varepsilon_{ij}^{(l)}/m_{ij})\Big)x_{ij}^{u}, \tag{11a}$$
$$= t_{ij}^{0}\Big(x_{ij}^{u} + \sum_{l=1}^{n}(a_l \varepsilon_{ij}^{(l)}/m_{ij})\Big)\Big(\sum_{k=1}^{n}\varepsilon_{ij}^{(k)} - x^{r} - x^{c}\Big), \tag{11b}$$
$$\leq t_{ij}^{0}\Big(x_{ij}^{u} + \sum_{l=1}^{n}(a_l \varepsilon_{ij}^{(l)}/m_{ij})\Big)\Big(\sum_{k=1}^{n}\varepsilon_{ij}^{(k)} - x^{c}\Big). \tag{11c}$$

In (11b) we express x_{ij}^{u} by using a combination of (2) and (9). In the last step (11c), we add to $\hat{J}_{ij}$ the term $\sum_{l=1}^{n} t_{ij}^{0} a_l \varepsilon_{ij} x_{ij}^{r}/m_{ij}$. By adding this term, we consider a relaxation of the original problem (i.e., an upper bound of $\hat{J}_{ij}$). This modification, for which we provide intuition later (see Remark III.1), allows the proposed objective to be a convex quadratic function.

Hence, we define the AMoD problem to be

$$\min_{\mathbf{x}^{\mathcal{W}}, \mathbf{x}^{r}, \boldsymbol{\varepsilon}} \sum_{(i,j)\in\mathcal{A}} \hat{J}(x_{ij}, \boldsymbol{\varepsilon}_{ij}) + \sum_{(i,j)\in\mathcal{A}_{\mathrm{R}}} c_{ij} x_{ij}^{r}, \tag{12a}$$
$$\text{s.t.}\quad (5\text{b}) - (5\text{d})$$
$$\varepsilon_{ij}^{(k)} \geq \theta_{ij}^{(k)} - x_{ij}, \quad \forall (i,j)\in\mathcal{A}, \quad k = 1,\dots,n, \tag{12b}$$
$$\varepsilon_{ij}^{(k)} \geq 0, \quad \forall (i,j)\in\mathcal{A}, \quad k = 1,\dots,n, \tag{12c}$$

where $\boldsymbol{\varepsilon} = \{\varepsilon_{ij}^{(j)} \mid (i,j)\in\mathcal{A}, \quad k = 1,\dots,n\}$.

Theorem III.1. *Problem* (12) *is a linearly constrained convex Quadratic Program (QP) with linear equality constraints.*

Proof. We prove this by construction. We show that the $\mathbf{Q}$ matrix in the QP standard form (i.e., $\min_{\mathbf{x}} \mathbf{x}'\mathbf{Q}\mathbf{x}$, s.t. $\mathbf{A}\mathbf{x} \leq \mathbf{b}$) can be modified to be positive semidefinite (PSD). Note, that in (12a), the only quadratic term is of the form $\varepsilon_{ij}^{(l)}\varepsilon_{ij}^{(k)}$ and its matrix representation (i.e., $\boldsymbol{\varepsilon}'\mathbf{Q}\boldsymbol{\varepsilon}$) does not guarantee that $\mathbf{Q}$ is PSD. However, we observe that since we are minimizing, when $x_{ij} \leq \theta_{ij}^{(k)}$ then $\varepsilon_{ij}^{(k)} = 0$ and that when $x_{ij} \geq \theta_{ij}^{(k+1)}$ then $\varepsilon_{ij}^{(k)} = (\theta_{ij}^{(k+1)} - \theta_{ij}^{(k)})$. Therefore,

$$\varepsilon_{ij}^{(l)}\varepsilon_{ij}^{(k)} = \begin{cases} (\theta_{ij}^{(l+1)} - \theta_{ij}^{(l)})\varepsilon_{ij}^{(k)}, & \text{if } l < k, \\ \varepsilon_{ij}^{(l)}\varepsilon_{ij}^{(l)}, & \text{if } l = k, \\ \varepsilon_{ij}^{(l)}(\theta_{ij}^{(k+1)} - \theta_{ij}^{(k)}), & \text{if } l > k, \end{cases}$$

where the first case comes from the fact that in order for $\varepsilon_{ij}^{(k)}$ to be greater than zero, the flow x_{ij} must have exceeded $\theta_{ij}^{(l+1)}$ for $l < k$. Therefore, $\varepsilon_{ij}^{(l)}$ is at its maximum value of $(\theta_{ij}^{(l+1)} - \theta_{ij}^{(l)})$. The same analogy applies to the third case. Hence, the link-wise objective function of the QP without the rebalancing term is rewritten as

$$\hat{J}_{ij}^{\mathrm{QP}}(x_{ij}^{u}, \boldsymbol{\varepsilon}_{ij}) = t_{ij}^{0}\Bigg(x_{ij}^{u} + \sum_{l=1}^{n}\frac{a_l}{m_{ij}}\Bigg(\sum_{k=1}^{l-1}(\theta_{ij}^{(k+1)} - \theta_{ij}^{(k)})\varepsilon_{ij}^{(l)} + (\varepsilon_{ij}^{(l)})^2 + \sum_{k=l+1}^{n}(\theta_{ij}^{(l+1)} - \theta_{ij}^{(l)})\varepsilon_{ij}^{(k)}\Bigg)\Bigg).$$

Using this new formulation, we note that the $\mathbf{Q}$ matrix is the identity matrix which is PSD and therefore J_{ij}^{QP} is convex quadratic using [48, Prop. 3.1.1]. □

In contrast with our previous 3-segment method in [42], we obtain a better approximation of the original travel latency function while formulating the problem as a QP.

Remark III.1. We observe that the effect of adding $\sum_{l=1}^{n} t_{ij}^{0} a_l \varepsilon_{ij} x_{ij}^{r}/m_{ij}$ to (11a) implies taking into account congestion-aware rebalancing routing. However, this congestion-aware routing of the rebalancing vehicles has a lower impact in J_{ij}^{QP} than the AMoD users. This is because $a_0 = 0$ for $\mathbf{x}^{r}$ (i.e., the first term in (11c) does not include $\mathbf{x}^{r}$). Hence, the interpretation of this addition is that the rebalancing flows evaluate the travel latency function with the same structure as the AMoD flows but with $t_{ij}^{0} = 0$.

Remark III.2. A relevant trade-off worth noting is on the number of piecewise affine segments used to approximate the travel function. Even though a larger n will provide better approximations of $t(\cdot)$, and hence a better solution to the problem, this implies adding $|\mathcal{A}|$ additional variables and linear constraints to the formulation.

B. Linear Relaxation

Seeking a simpler formulation and faster computation performance of (12), we notice that it is possible to relax the QP to a Linear Program (LP) by modifying the only quadratic term in (12a), i.e., $(\varepsilon_{ij}^{(l)})^2$. We approximate this using $\varepsilon_{ij}^{(l)}\theta_{ij}^{(l+1)}$ and observe that when $x_{ij} \leq \theta_{ij}^{(l)}$ or $x_{ij} \geq \theta_{ij}^{(l+1)}$ we recover exactly $(\varepsilon_{ij}^{(l)})^2$. However, a gap exists when $x_{ij} \in (\theta_{ij}^{(l)}, \theta_{ij}^{(l+1)})$ which can be diminished by adding more linear segments to $\hat{t}(\cdot)$ and consequently decreasing the range of $(\theta_{ij}^{(l)}, \theta_{ij}^{(l+1)})$.

Lemma III.1. *Assuming the distance between the break points of the linear segments is uniform, i.e., $\theta_{ij}^{(l+1)} - \theta_{ij}^{(l)} = \theta^{(n)}/n$, for $l = 1, \dots, n-1$, then the objective function of the LP formulation approximates the QP objective function by an error upper-bounded by $a_n(\theta^{(n)})^2/4n^2) \sum_{(i,j)\in\mathcal{A}} t_{ij}^0 m_{ij}$.*

Proof. Notice that the maximum total error between the LP and QP is expressed by:

$$\begin{aligned}
&\sum_{(i,j)\in\mathcal{A}} \max_{l=1,\dots,n} \left\{ \frac{a_l t_{ij}^0}{m_{ij}} ((\theta_{ij}^{(l+1)} - \theta_{ij}^{(l)})\varepsilon_{ij}^{(l)} - (\varepsilon_{ij}^{(l)})^2) \right\} && \text{(14a)}\\
&\quad = \sum_{(i,j)\in\mathcal{A}} \max_{l=1,\dots,n} \left\{ \frac{a_l t_{ij}^0}{m_{ij}} (\frac{\theta_{ij}^n}{n}\varepsilon_{ij}^{(l)} - (\varepsilon_{ij}^{(l)})^2) \right\} && \text{(14b)}\\
&\quad = \sum_{(i,j)\in\mathcal{A}} \max_{l=1,\dots,n} \left\{ \frac{a_l t_{ij}^0}{m_{ij}} (\frac{\theta_{ij}^n}{n}\frac{\theta_{ij}^n}{2n} - (\frac{\theta_{ij}^n}{2n})^2 \right\} && \text{(14c)}\\
&\quad = \sum_{(i,j)\in\mathcal{A}} \frac{a_n t_{ij}^0}{m_{ij}} \Big(\frac{(\theta_{ij}^n)^2}{4n^2} \Big) && \text{(14d)}\\
&\quad = \frac{a_n(\theta^{(n)})^2}{4n^2} \sum_{(i,j)\in\mathcal{A}} t_{ij}^0 m_{ij}, && \text{(14e)}
\end{aligned}$$

where the first equality comes from the fact that we use uniform distances for the piecewise regions, and the second equality comes from the fact that $\varepsilon_{ij}^{(l)*} = \theta_{ij}^2/2n$ maximizes equation (14b). Finally the last step selects the last segment n by observing that a_n has the steepest slope by assumption. □

Theorem III.2. *Let the total flow and the capacity of every link be upper-bounded and assume $a_0 \leq a_1 \leq, \dots, \leq a_n < \infty$. Then, as $n \to \infty$, the solution of the LP problem recovers the solution of the QP.*

Proof. Without loss of generality, let us select the thresholds $\boldsymbol{\theta}$ in a uniform manner as in Lemma III.1. The proof follows immediately after observing in (14e) that, for a bounded a_n, m_{ij} and t_{ij}^0, the error goes to zero as $n \to \infty$. □

Interestingly, these two reformulations, QP and LP, together with Theorems III.1 and III.2 show that a LP can be solved instead of the original convex program described in (5). This LP approximates the solution of the QP which in turn approximates the solution of the original problem. These two are asymptotically optimal in the number of segments used to describe the nonlinear function $t(\cdot)$ in the objective.

C. Origin-based Formulation (Flow-bundling)

So far, we have formulated the problem such that for every OD pair $\mathbf{w} \in \mathcal{W}$ we introduce $|\mathcal{A}|$ decision variables. The total number of variables in our QP (or LP) is then $(n + 1 + |\mathcal{W}|)|\mathcal{A}|$, which is typically dominated by the number of OD pairs $|\mathcal{W}|$. In practice, this number can be very large, sometimes up to $|\mathcal{V}|^2$. Hence, solving the problem using the previous formulations may require large memory capabilities.

To mitigate this issue, we leverage similar ideas to [38] which aggregate flows by origin with the objective to reduce the number of variables and constraints of the QP and LP without losing information. This flow aggregation by origin allows us to reduce the number of variables to be in the order of $(n + 1 + |\mathcal{V}|)|\mathcal{A}|$, which makes the problem significantly faster to solve. Let us denote the set of origin (sources) $\mathcal{S} = \{w_s \mid d_{(w_s,w_t)}^u > 0,\ (w_s, w_t) \in \mathcal{W}\}$ and the flow on the network with s as it source by $\mathbf{x}^s$; the total user flow on a link is then $\mathbf{x}^u = \sum_{s\in\mathcal{S}} \mathbf{x}^s$ and the set of user origin-link variables be $\mathbf{x}^{\mathcal{S}} = \{\mathbf{x}^s \mid s \in \mathcal{S}\}$. For every origin s, let $\psi^s(j)$ be the node *imbalance* describing the excess demand or supply at each node. This is

$$\psi^s(j) = \begin{cases} \displaystyle\sum_{t:(s,t)\in\mathcal{W}} -d_{(s,t)}^u, & \text{if } j = s,\\ 0, & \text{if } j \neq s, t,\\ d_{(s,t)}^u, & \text{if } j = t. \end{cases}$$

With this definition in hand, we establish the origin-based problem as follows

$$\begin{aligned}
\min_{\mathbf{x}^{\mathcal{S}}\geq\mathbf{0},\mathbf{x}^r\geq\mathbf{0}} \ & \sum_{(i,j)\in\mathcal{A}} \hat{J}_{ij}(x_{ij}, \boldsymbol{\varepsilon}_{ij}) + \sum_{(i,j)\in\mathcal{A}_{\mathrm{R}}} c_{ij} x_{ij}^r && \text{(16a)}\\
\text{s.t.} \ & \sum_{i:(i,j)\in\mathcal{A}} x_{ij}^s - \sum_{k:(j,k)\in\mathcal{A}} x_{jk}^s = \psi^s(j), && \text{(16b)}\\
& \qquad\qquad\qquad\qquad \forall j \in \mathcal{N},\ \forall s \in \mathcal{S},\\
& \text{(5c)},\ \text{(12b)},\ \text{(12c)},
\end{aligned}$$

where $x_{ij} = x_{ij}^u + x_{ij}^r + x_{ij}^p = \sum_{s\in\mathcal{S}} x_{ij}^s + x_{ij}^r + x_{ij}^p$.

Now, we show that the resulting flows of the solution of the origin-based problem (16) are the same as the OD-based problem (12). To accomplish this, we use the next result.

Lemma III.2. *Let $\mathbf{x}^{\mathcal{S}*}$ be the solution to the origin-based problem* (16) *and $\mathbf{x}^{s*}$ the flows associated with origin s. Then, the subset of arcs $\mathcal{A}^{s*} = \{(i,j) \mid x_{ij}^s > 0,\ (i,j) \in \mathcal{A})\}$ with positive flow from origin s has no direct cycles.*

Proof. We use contradiction. Assume that there is a cycle $\mathcal{C}$ where $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_1) \in \mathcal{A}^s$ and let h_i^s be the marginal cost (related to the SO solution) of the cheapest path from s to i. Then, consider any $(i,j) \in \mathcal{A}^s$, which implies that (i) there exists a positive flow path from s to i and (ii) $x_{ij}^s > 0$. Since we are minimizing a function where $t_{ij}(x)$ and $\hat{t}_{ij}(x)$ are strictly positive and monotonically increasing for all $(i,j) \in \mathcal{A}$, the path connecting s to i has to be a minimum cost path. Assume this cost to be T_{si} and since $x_{ij}^s > 0$, the cost to j is $T_{sj} = T_{si} + t_{ij}$. Note that by definition $T_{sj} \leq T_{si} + t_{ij}$. However, if $T_{sj} < T_{si} + t_{ij}$ then there must exist a lower-cost path to j than any of those passing through i. Hence, we must have $T_{sj} = T_{si} + t_{ij}$ for all the links in $\mathcal{A}^s$. Using the fact that all travel times are strictly positive, this implies that for the cycle $\mathcal{C}$ we have $T_1 < T_2 < \ldots < T_k < T_1$ which is logically inconsistent. □

Lemma III.3. *The link-flow solution of the origin-based problem* (16) *is equivalent to the solution of the OD-based problem* (12). *i,e,. for any origin s , we have $\sum_{t:(s,t)\in\mathcal{W}} x^{\mathbf{w}*} = \mathbf{x}^{s*}$.*

Proof. Similar to [38], we prove this by construction and use the flow decomposition algorithms and results in [49, Thm. 3.5]. We begin by decomposing the origin-based solution $\mathbf{x}^s$ of origin s to a set of acyclic paths $\mathcal{P}^s$ such that $\mathbf{x}^{(w_s,w_t)} = \mathbf{x}_t^s$ where $\mathbf{x}_t^s$ is the acyclic decomposed flow from $\mathbf{x}^s$ going from

s to t. We conclude the proof by observing that the origin-based solution has no cycle (by Lemma III.2) and the fact that it is possible to decompose the problem to flows using [49, Thm. 3.5]. □

Therefore, using the result of Lemma III.3, we can restate the network model in terms of the origin-based flows which reduces the size of the model, memory requirements, and solution time.

D. Disjoint Strategy

We have discussed methods to solve the SO routing and rebalancing problem jointly. However, a different approach is to tackle these two problems separately. That is, we first solve the routing problem followed by the load-balancing problem. Mathematically, this is to first solve

$$\min_{\mathbf{x}^{\mathcal{W}} \geq 0} \sum_{(i,j)\in\mathcal{A}} t_{ij}(x_{ij})x_{ij}^{u}, \quad \text{s.t.} \quad (5b), \tag{17}$$

and then, use the optimal $\mathbf{x}^{u*}$, to solve

$$\min_{\mathbf{x}^{r} \geq 0} \quad \mathbf{c}'\mathbf{x}^{r}, \quad \text{s.t.} \quad (5c). \tag{18}$$

It is relevant to highlight that this strategy is interesting given its fast computation. Problem (17) is a constrained nonlinear program (NLP) which can be solved using any of the typical algorithms for the TAP, for example Frank-Wolfe or TAPAS; and problem (18) is a LP with $|\mathcal{V}|$ variables.

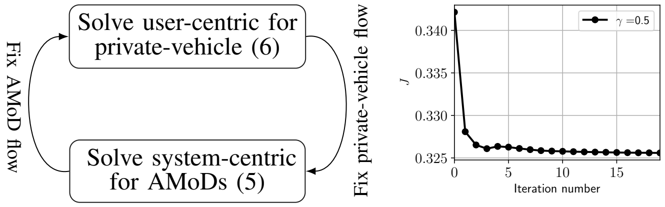
IV. AMoD in Mixed Traffic

We have not yet discussed how to address the nested problem (7) which considers the interaction between the fleet of AMoDs vehicles and self-interested private vehicles. We utilize the framework in [23] which applies a sequential approach (diagonalization scheme [24], [25]) to find an equilibrium between the AMoD and private flows (Fig. 3).

Rather than addressing the bi-level problem (7), we solve (6) for the private vehicles and (5) for the AMoD fleet (using any of the methods in the previous section) and iterate until convergence. Namely, for a private vehicle demand $\mathbf{g}^{u}$ we solve $\mathbf{x}^{p} = \texttt{TAP}(\mathbf{g}^{p}, \mathbf{0})$. Next, we solve (5) for AMoD demand $\mathbf{g}^{u}$ with fixed input $\mathbf{x}^{p}$ (the output of the earlier solved TAP). Since private vehicles were not aware of the AMoD flow in the system while finding their routes, we re-solve the TAP by considering a fixed AMoD flow equal to $\mathbf{x}^{u} + \mathbf{x}^{r}$, i.e., we solve $\mathbf{x}^{p} = \texttt{TAP}(\mathbf{g}^{p}, \mathbf{x}^{u} + \mathbf{x}^{r})$. Then, we iterate this process until it converges. An example is shown in Fig. 3b.

In this paper, we do not establish theoretical results on the stability or uniqueness of the players (AMoD fleet and private vehicles) equilibria. These results are hard to achieve due to the non-separability of the cost function regarding the players' strategies as pointed out in [24]. Still, empirically, this sequential approach always converges in a few iterations.

Remark IV.1. Notice that when using this iterative method, some of the parameters can be updated. In particular if one uses the disjoint strategy in Sec. III-D to solve the routing and rebalancing problem, one could update the $\mathbf{c}$ vector at each subsequent iteration by the calculated travel times $\mathbf{t}(\mathbf{x})$ at the current iteration. When doing this, one obtains a more precise cost function by weighting vector $\mathbf{c}$ with the updated travel times.



(a) (b)

Fig. 3. (a): A sketch of the procedure for solving the bi-level problem (7). (b): An example of the total cost converging for an AMoD penetration rate of 0.5 on the NYC sub-network

V. Route Recovery Strategies

All methods discussed thus far solve the routing and rebalancing problem by choosing $\mathbf{x}^{u}$ and $\mathbf{x}^{r}$ that minimize a performance metric. Even if this flow solution allows us to assess the network deficiencies and to plan for infrastructure improvements, flows do not give explicit routes to a given vehicle. Therefore, we need to extract the routes to implement the desired flow-based solution we derive. An advantage of the proposed models in (12) and (16) in contrast to classical link-based TAP, is that they allow for tracing and recovering the routes (or paths). We present simple and distributed algorithms to recover the routes from the OD-based or origin-based solutions presented in Sec. III, as well as an algorithm to retrieve the rebalancing routes.

A. AMoD User Flow

1) OD-pair model: Let the optimal solution of the routing and rebalancing problem be $(\mathbf{x}^{\mathcal{W}*}, \mathbf{x}^{r*})$ and denote with $\mathcal{R}_{\mathbf{w}}$ a set of routes for OD pair $\mathbf{w}$. For each $\mathbf{w}$, we let $\boldsymbol{\pi} \in [0,1]^{|\mathcal{R}_{\mathbf{w}}|}$ be a vector with elements denoting the fraction of vehicle flow routed trough route $i \in \mathcal{R}_{\mathbf{w}}$. We denote with $\mathbf{A}$ the route-link incidence matrix of $\mathcal{R}_{\mathbf{w}}$. With these definitions, we provide a column-generation approach in which we find the routes of an OD-pair by sequentially solving the linear program:

$$\min_{\boldsymbol{\pi}\in[0,1]} \|\mathbf{A}\boldsymbol{\pi} d_{\mathbf{w}} - \mathbf{x}^{\mathbf{w}*}\| \tag{19a}$$

$$\text{s.t. } \boldsymbol{\pi}'\mathbf{1} = 1, \tag{19b}$$

where the product $\mathbf{A}\boldsymbol{\pi} d_{\mathbf{w}}$ is equal to the estimated link flow induced by routing $d_{\mathbf{w}}\pi_{i}$ flow trough each route and the constraint ensures that the vector $\boldsymbol{\pi}$ is a probability distribution.

The problem of selecting which routes to include in $\mathcal{R}_{\mathbf{w}}$ (column selection) is yet to be addressed. We use the greedy approach of adding the next shortest route to $\mathcal{R}_{\mathbf{w}}$ and re-solving problem (19). To terminate the algorithm, we use a user-defined parameter ξ (as shown in Alg. 1). It is worth pointing out that this procedure can run in parallel for each OD pair. For uncongested networks, we expect it to converge fast. This is because when there is little congestion, the majority of vehicles will be routed trough the shortest paths, which are the first ones to be added into the set $\mathcal{R}_{\mathbf{w}}$. Finally, note that this formulation is only available if we have information on $\mathbf{x}^{\mathbf{w}*}$ for all $\mathbf{w} \in \mathcal{W}$.

Algorithm 1 Route-recovery for a specific OD pair

```
procedure RouteRecovery($\mathbf{A}$, $\mathbf{d_w}$, $\mathbf{x^{w*}}$, $\xi$)
    Initialize: $x_{ij} \leftarrow d_{\mathbf{w}} \mathbb{1}_{(i,j)\in\text{shortest route for } \mathbf{w}}$
    while $\|\mathbf{x} - \mathbf{x^{w*}}\| > \xi$ do
        $\mathcal{R}_{\mathbf{w}} \leftarrow$ append next shortest path
        $\boldsymbol{\pi}_{\mathbf{w}} \leftarrow$ solve (19)
    end while
end procedure
```

2) *Origin-based model:* Let $\mathbf{x}^{s*}$ be the solution of (16) and let $\mathcal{T}_s = \{j \mid \psi_s(j) < 0,\ j \in \mathcal{V}\}$ be the set of destinations (targets) from origin s. Let $\psi_s(j)$ be the node imbalance of node j of the origin-based flows initialized at s. For each origin s, one can decompose its OD-flow solution by solving the following LP:

$$\min_{\{\mathbf{x}^t\}_{t\in\mathcal{T}_s}\geq 0} \quad \mathbf{t}^{0\prime}\mathbf{x} \tag{20a}$$

$$\text{s.t} \sum_{i:(i,j)\in\mathcal{A}} x_{ij} - \sum_{k:(j,t)\in\mathcal{A}} x_{tj} = \psi_s(t), \quad \forall j \in \mathcal{V}, \tag{20b}$$

$$\sum_{i:(i,j)\in\mathcal{A}} x^t_{ij} - \sum_{k:(j,k)\in\mathcal{A}} x^t_{ij} \geq 0, \quad \forall j \in \mathcal{V}\backslash\{s\}, \tag{20c}$$

$$\sum_{i:(i,s)\in\mathcal{A}} x^t_{is} - \sum_{k:(s,k)\in\mathcal{A}} x^t_{sj} = \psi_s(t), \tag{20d}$$

$$\mathbf{x}^{s*} - \mathbf{x} = \mathbf{0}. \tag{20e}$$

Here $\mathbf{x}$ is the origin-based flow (equivalent to $\mathbf{x}^s$) defined as $\mathbf{x} = \sum_{t\in\mathcal{T}_s} \mathbf{x}^t$. The first constraint, (20c), takes care of demand satisfaction and flow conservation. The second constraint, (20c), considers flow conservation but allows certain target nodes to have excess flow, allowing them to be a destination. Constraint (20d) ensures that the decision variables are designed for that specific origin s by ensuring that the required flow is leaving that node. Then, (20e), forces the solution to equal the origin-based flows. Finally, the objective (20a) is defined with the purpose of breaking ties in case multiple combinations of flows can satisfy the constraints (e.g. cycles).

Notice that as a result of Lemma III.3, this problem is always feasible and recovers the OD-based solution. Once this is found, we could use Alg. 1 to find the path-based solution. Problem (20) is stated as a linear program that could be solved in parallel for each origin-based solution s, therefore, we expect this optimization process to be computationally efficient.

B. Rebalancing Flows

The problem of finding the paths of the rebalancing flows is more complex than that of finding the AMoD routes. This is because we have no information about their origin and destinations. Rather, the only information available is the aggregated link flows that the rebalancing vehicles are taking to minimize (5a) and comply with the load-balancing constraint (5c). Hence, a first step to recover the paths is to calculate the rebalancing node *imbalances* $\phi(j)$ for every node j defined over the available rebalancing solution x^r:

$$\phi(j) = \sum_{i:(i,j)\in\mathcal{A}_\text{R}} x^r_{ij} - \sum_{k:(j,k)\in\mathcal{A}_\text{R}} x^r_{jk}.$$

We define a rebalancing origin to be a deficit flow node, and its set $\mathcal{S}_r = \{j \mid \phi(j) < 0, j \in \mathcal{A}_\text{R}\}$; similarly, the rebalancing destination set is defined as $\mathcal{T}_r = \{j \mid \phi(j) > 0, j \in \mathcal{A}_\text{R}\}$. Notice that these definitions are made in $\mathcal{A}_\text{R}$ instead than in $\mathcal{A}$ as the rebalancing vehicles only exist in G_R. Then, we aim to recover an OD rebalancing solution by solving

$$\min_{\{\mathbf{x}^s\}_{s\in\mathcal{S}_r}\geq 0} \quad \mathbf{t}^{0\prime}\mathbf{x} \tag{21a}$$

$$\text{s.t} \sum_{i:(i,j)\in\mathcal{A}_\text{R}} x_{ij} - \sum_{k:(j,k)\in\mathcal{A}_\text{R}} x_{jk} = \phi(j), \quad \forall j \in \mathcal{V}_\text{R}, \tag{21b}$$

$$\sum_{i:(i,j)\in\mathcal{A}_\text{R}} x^s_{is} - \sum_{k:(s,k)\in\mathcal{A}_\text{R}} x^s_{sj} = \phi(s), \quad \forall s \in \mathcal{S}_r, \tag{21c}$$

$$\sum_{i:(i,j)\in\mathcal{A}_\text{R}} x^s_{ij} - \sum_{k:(j,k)\in\mathcal{A}_\text{R}} x^s_{ij} \geq 0,\ \forall j \in \mathcal{V}_\text{R}\backslash\{s\}, \quad \forall s \in \mathcal{S}_r, \tag{21d}$$

$$\mathbf{x} - \mathbf{x}^r = 0, \tag{21e}$$

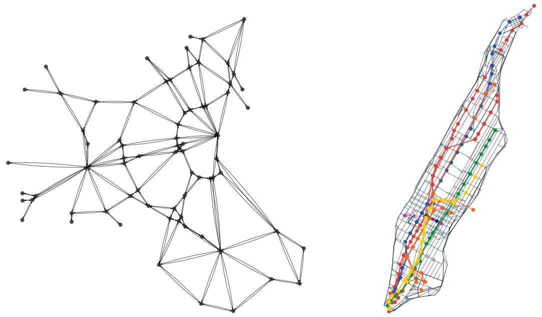
where we define $\mathbf{x} = \sum_{s\in\mathcal{S}_r} \mathbf{x}^s$ and $\mathbf{x}^r$ is the available link flow solution of (5a). Notice that the model follows the same intuition as (20). Constraint (21b) takes care of the total flow conservation of the rebalancing flow, constraint (21c) ensures that, for each origin variables $\mathbf{x}^s$, the outflow of node s is equal to the excess of vehicles. Constraint (21d) allows any node different than s be a potential destination of the rebalancing flow. Finally, (21e) ensures that the aggregated rebalancing flows by origin matches the rebalancing flow obtained in the AMoD users problem.

Once we have decomposed the rebalancing flow by origins, we have for each rebalance origin s an origin-based rebalancing flow. Then, since now we have available the flows in an origin-based form, we can apply (20) in parallel for each $s \in \mathcal{S}_r$ to decompose to an OD-flow solution, and finally use Alg. 1 to recover the routes.

Remark V.1. For both of (20) and (21) it is possible to dualize the last constraint (i.e., penalize $\|\mathbf{x} - \mathbf{x}^r\|$ on the cost function). This will make the optimization less restrictive and improves the solution time by lowering the quality of the solution. It is hard to estimate exactly what would be the impact of this dualization in terms of efficiency. However, for low-traffic networks, we expect (20) and (21) to be faster to solve as we expect the total flow on every link will belong to less OD pairs. In contrast, when dealing with high-traffic scenarios, the total flow on a link might be composed of many OD pairs, making the problem harder (slower) to decompose. In practice we have observed that for low-traffic networks less than 3 routes per OD pair are enough to get an accurate solution, whereas for high-traffic cases, the number of routes required for good solutions are in the order of 6 to 8. Still, the problems as stated in this paper can be solved to optimality.

VI. Numerical Results and Case Studies

To validate our proposed routing algorithms, we consider two data-driven case studies on sub-networks of Eastern Massachusetts interstate highways (EMA) and New York City (NYC). The EMA road network (Fig. 4a) consists of 74 nodes, 258 links, and 1113 OD pairs, and it captures the dynamics in the context of suburban/urban mobility. Complementary to EMA, the NYC network focuses on urban mobility. The



(a) EMA subnetwork (b) NYC subnetwork

Fig. 4. Subnetworks used for the experiments.

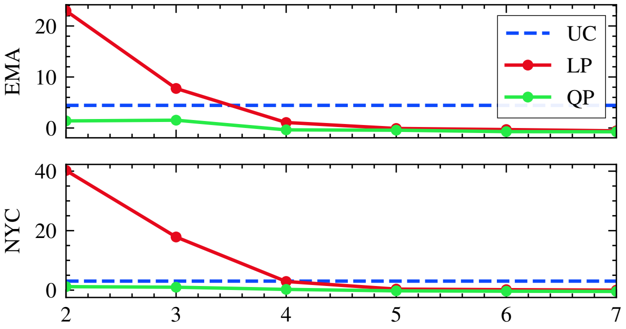
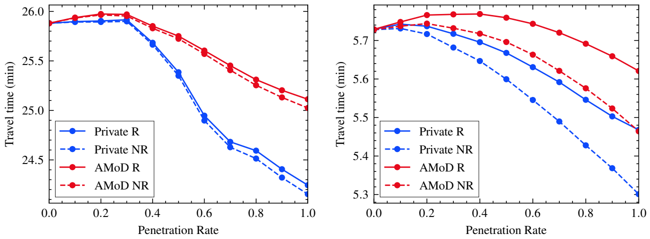


Fig. 5. Deviation in percentage terms between the approximated model and the optimal solution of the non-rebalancing SO problem (baseline). UC indicates how much the solution of the UC deviates from the SO. This gap between the UC and SO models is referred to as the *Price of Anarchy* [15].



(a) EMA (b) NYC

Fig. 6. Travel times for AMoD users, private vehicles and all vehicles (total) for different penetration rates of AMoDs in the network."R" stands for an approach that considers rebalancing while "NR" does not.

NYC topology was constructed using OpenStreetMaps [50] and contains 3317 arcs, 1351 nodes. The OD demand was built using historical data taxi rides (courtesy of the New York Taxi Commission [51]) that occurred on March 1, 2012, between 18:00 and 20:00 hrs which accounts for 8658 OD pairs.

A. Convergence of the approximated model

Our first experiment shows empirically our results of Theorem III.1 and the observation that as n increases, the approximation of $\hat{t}(\cdot)$ to $t(\cdot)$ is tighter and therefore the QP and LP problems approximate the original problem more accurately. To generate this experiment, we consider a problem with no rebalancing (not including the rebalancing constraints) and with no exogenous flow (i.e., $\mathbf{x}^c = 0$). This is exactly the SO formulation of the TAP for which we use the Frank-Wolfe algorithm to find its solution (we solve the UC problem using the same method). Then, we solve the QP and LP versions of the CARSn model for different values of n and observe that, as we increase n, the objective of CARSn converges to the objective of the SO. For example, for both networks shown in Fig. 5, we observe that for $n = 6$ the objective of the approximated models QP and LP are very close to the SO solution.

B. Joint vs. Disjoint Solution

This experiment aims to compare the solution of the joint and disjoint formulation of the problem. That is, solving (12) against the disjoint method in Sec. III-D. We compare this by showing the improvement (ratio between the value of the objective functions) of the joint over the disjoint approach. For EMA and NYC we observe an improvement in the objective of 3.85% and 0.91%, respectively. Moreover, we consider the case of NYC network with a higher demand, which we simulate by multiplying the demand vector $\mathbf{g}$ by 2. The improvement of the joint formulation over the disjoint model for this demand level is 5.85%. These results highlight the achievable benefits, especially for high demand scenarios, of jointly solving the routing and rebalancing problems, rather than separately.

C. System-Optimal Routing and Rebalancing Trade-off

Considering the existence of selfish privately-owned vehicles and centrally-controlled AMoD vehicles, we analyze the trade-off that exists between system-optimal AMoD routing and the additional traffic due to AMoD rebalancing in terms of average travel times. We tackle the bilevel Problem (7) following the iterative methodology presented in Sec. IV. We use different penetration rates of AMoD customers with respect to the total demand, i.e., a penetration rate of 0.3 will indicate that 30% of total demand uses the AMoD service while the rest use private vehicle. More specifically, we let $\gamma \in [0, 1]$ be the penetration rate and $\mathbf{g}$ the total OD demand. Then, we assume that $\mathbf{g}^u = \gamma\mathbf{g}$ and $\mathbf{g}^p = (1-\gamma)\mathbf{g}$ are the AMoD's and private vehicles' demand, respectively. However, different demand separation criteria can be readily implemented in this framework.

As shown in Fig. 6, the introduction of AMoD users into the system not only improves the overall travel time of AMoD users themselves, *but reduces the travel time of private vehicles even more*. This is because "smart" routing decisions of AMoD vehicles reduce the traffic intensity on congested roads, which consequently allows private vehicles to travel faster. For EMA, the impact of rebalancing is negligible, and increasing the percentage of AMoD users in the network allows to reduce travel time by up to 3%. For NYC, we observe that rebalancing indeed is detrimental for low penetration rates, but as the percentage of SO vehicles increases, social routing improves travel times for both AMoD users and private vehicles. Yet, in general, the impact of rebalancing on the system-level performance depends on the network topology, and on the symmetry and intensity of the OD demand distribution.

D. Intermodal AMoD

We study the impact of intermodal SO routing against UC private vehicle routing for the NYC network. We consider high congestion levels and run the experiments by multiplying the demand distribution vector $\mathbf{g}$ by a factor of 1.5 (see details of the demand in the online repository [52]). Similar to our last experiment, we run the analysis for different penetration rates. We assume that AMoD users are able to take public transit (subway), walk, or bike towards their destination and switch between modes in their route. In contrast to the AMoD users, we limit the flexibility of private vehicles to exclusively use the road network (no subway, biking or walking) due to parking constraints. The top row of plots on Fig. 7 display, on the left, the travel time for the two user types as the penetration rate of AMoD users increases, and on the right, the modal distribution of the total kilometers traveled. The top row shows the results when only taxi-type service is offered to AMoD users (no subway, walking or biking). We observe that the extra rebalancing flow increases the overall travel times of the system more than what SO routing can reduce it. This result confirms the fact that pure vehicle-based MoD systems can have detrimental effects on the overall travel time [53]. The subsequent plots show that by considering the flexibility of other modes of transportation, AMoD mobility can reduce traffic congestion. The second row of plots in Fig. 7 includes a public transit option, the third row adds a pedestrian option (6 km/h), and the last one also considers biking (10 km/h) as an option[2]. In general, we see that the more modes of transportation are offered, the lower the travel times for everyone. In addition, when new options for mobility are offered, AMoD users could reach lower travel times than private vehicles, something which is impossible to achieve when only taxi-rides are available (due to the assumption on UC routing). This happens because they are more flexible and their overall transportation capacity is larger than the available capacity for private vehicles. However, at almost 100% penetration rates, it still seems that being selfish is benefited, raising interesting questions on how to incentivize users to act in a system-centric fashion. Finally, one can observe by comparing the first three rows of plots in Fig. 7, that when a tiny fraction of flow is accessible via subway or walking, the travel times were reduced by almost 50%.

To account for more traffic intensities, Table I presents the results for an AMoD system with taxi-type (Veh), subway (Sub), pedestrian (Ped), and biking (Bike) layers when demand is multiplied by a factor of 1, 1.5, (corresponding to the last subplots of Fig 7) and 2. The Table shows results for the overall travel times and modal distributions of the I-AMoD kilometers traveled for penetration rates equal to 0, 50%, and 100%. In general we see that the higher the congestion, the larger benefit in travel times due to the enlarged capacity resulting form intermodal options. In addition, we see that subway and biking options are critical to improve travel times.

TABLE I
INTERMODAL AMOD RESULTS FOR DIFFERENT TRAFFIC INTENSITIES.

Demand	Pen. Rate	Avg. Travel Time (min)		I-AMoD Modal Distribution				
		I-AMoD	Private Veh	Veh	Reb	Bike	Sub	Ped
1	0	5.2	5.7	80%	15%	0%	5%	0%
	0.5	5.2	5.4	81%	14%	0%	5%	0%
	1	5.0	5.0	82%	13%	0%	4%	0%
1.5	0	7.5	8.8	69%	23%	2%	6%	0%
	0.5	7.0	6.9	74%	17%	4%	5%	0%
	1	6.3	5.7	78%	13%	4%	5%	0%
2	0	10.7	15.8	52%	28%	12%	7%	1%
	0.5	9.1	8.5	68%	13%	12%	6%	0%
	1	7.7	6.2	75%	8%	11%	6%	0%

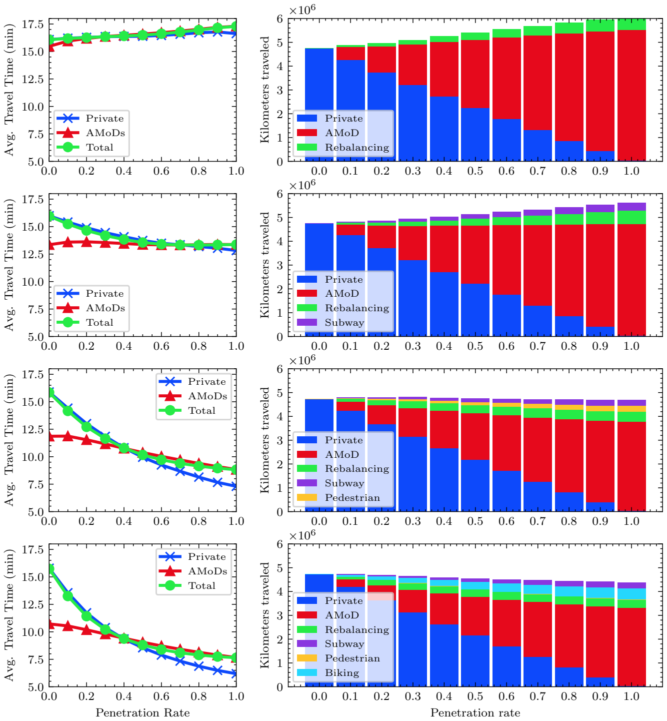


Fig. 7. System performance with alternative modes of transport for a relatively high-demand scenario in NYC (we increase demand by a factor of 1.5). The first column of plots show the average travel time for different AMoD penetration rates while the second row depicts the miles traveled per mode of transportation for each penetration rate.

In conclusion, we observe that while pure AMoD systems might decrease the system-level performance due to the additional congestion resulting from rebalancing, intermodal centralized-routing can significantly improve the overall travel times. Especially at high levels of demand, we see that, while SO intermodal routing can significantly improve travel times, it comes with the social dilemma that, from a UC perspective, being selfish would still be optimal.

E. Route Recovery Example

We show the applicability of our route-recovery strategies presented in Section V. We implement the distribute version of the route-recovery algorithm described in Section V-A2 on the solution flows of the origin-based problem (16). We compute the routes using a commercial laptop with 8 cores for which we recover the routes in the order of 30 seconds to one minute making it accessible for real-time implementation. Fig. 8 shows the different SO routes connecting a single OD pair. The left plot shows the recommended routes which only include taxi-type service. Furthermore, the right plot shows an intermodal route composed of taking taxi (solid lines) and subway (dotted line).

[2] For biking, we include a set of constraints in the same spirit as (5c) but for the bike layer. This ensures the balance between the incoming and outgoing flow of bikes at each node which goes in line with the dynamics of bike sharing systems [54].

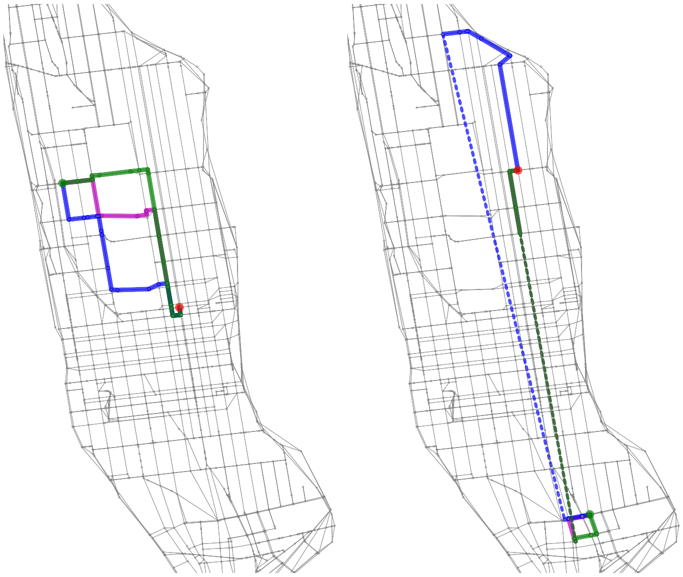


Fig. 8. Example of the SO routes connecting an OD pair. Green and red dots represent origin and destinations, respectively. Solid lines portray traveling flow in the road network while dotted lines describe flow traveling via subway.

VII. Conclusions

In this paper, we proposed a methodology to optimize the routes and rebalancing policies of a congestion-aware intermodal Autonomous Mobility-on-Demand (AMoD) system when it interacts with exogenous private traffic. To address the issue of non-convexity for this problem, we used a piecewise affine approximation of the travel latency function and proved that as the number of piecewise affine segments increases, the solution to the problem converges to the solution of the relaxed original problem. Using examples with the Eastern Massachusetts Area (EMA) and New York City (NYC) networks, (i) we empirically showed that the piecewise affine relaxation is asymptotically optimal, (ii) we captured the benefits of centrally controlling an intermodal AMoD system under mixed traffic conditions when different modes of transportation are available, (iii) we measured the advantage of using the approximated joint method versus a method that separately optimizes the routing and rebalancing policies, (iv) we revealed the existing trade-off between extra rebalancing flow and smart routing decisions, and (v) we tested the applicability of our proposed route-recovery algorithms in a real case study using the NYC network.

This paper opens the field for the following extensions. First, we would like to use these methodologies for solving a larger class of problems characterized as Traffic Assignment Problem with *side constraints* (TAPSC), i.e., Traffic Assignment Problems (TAPs) with arbitrary constraints such as the link-capacitated TAP [55]. Second, we are interested in leveraging our route-recovery strategies for real-time routing. Finally, we would like to devise pricing and incentive schemes to align the interests of selfish users with the system optimum and realize the full potential of smart intermodal mobility systems [45], [56].

References

[1] J. Bates and D. Leibling, "Spaced out," *Perspectives on parking policy*, vol. 9, 2012.

[2] W. J. Mitchell, C. E. Borroni-Bird, and L. D. Burns, *Reinventing the automobile: Personal urban mobility for the 21st century*. MIT Press, 2010.

[3] FTA, "Mobility on demand sandbox program," https://www.transit.dot.gov/research-innovation/mobility-demand-mod-sandbox-program, Jul 2020.

[4] S. Banerjee, R. Johari, and C. Riquelme, "Pricing in ride-sharing platforms: A queueing-theoretic approach," in *ACM Conf. on Economics and Computation*, 2015, pp. 639–639.

[5] K. Bimpikis, O. Candogan, and D. Saban, "Spatial pricing in ride-sharing networks," *Operations Research*, vol. 67, no. 3, pp. 744–769, 2019.

[6] A. J. Hawkins, "Robotaxis get the green light for paid rides in california," Nov 2020. [Online]. Available: https://www.theverge.com/2020/11/23/21591045/california-robotaxi-paid-rides-cpuc-permits

[7] M. Toh, "Self-driving robotaxis are taking off in china," Dec 2020. [Online]. Available: https://edition.cnn.com/2020/12/03/tech/autox-robotaxi-china-intl-hnk/index.html

[8] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[9] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[10] G. Ramalingam and T. Reps, "An incremental algorithm for a generalization of the shortest-path problem," *Journal of Algorithms*, vol. 21, no. 2, pp. 267–305, 1996.

[11] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[12] A. Hertz, G. Laporte, and M. Mittaz, "A tabu search heuristic for the capacitated arc routing problem," *Operations Research*, vol. 48, no. 1, pp. 129–135, 2000.

[13] C. W. Ahn and R. S. Ramakrishna, "A genetic algorithm for shortest path routing problem and the sizing of populations," *IEEE Transactions on evolutionary computation*, vol. 6, no. 6, pp. 566–579, 2002.

[14] T. Roughgarden and É. Tardos, "Bounding the inefficiency of equilibria in nonatomic congestion games," *Games and Economic Behavior*, vol. 47, no. 2, pp. 389–403, 2004.

[15] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies," *Proceedings of the IEEE*, vol. 106, no. 4, pp. 538–553, 2018.

[16] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*. Dover Publications, 1994, vol. 54, no. 2.

[17] M. Frank, P. Wolfe *et al.*, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[18] L. J. LeBlanc, R. V. Helgason, and D. E. Boyce, "Improved efficiency of the frank-wolfe algorithm for convex network programs," *Transportation Science*, vol. 19, no. 4, pp. 445–462, 1985.

[19] M. Florian, J. Guálat, and H. Spiess, "An efficient implementation of the "partan" variant of the linear approximation method for the network equilibrium problem," *Networks*, vol. 17, no. 3, pp. 319–339, 1987.

[20] H. Bar-Gera, "Traffic assignment by paired alternative segments," *Transportation Research Part B: Methodological*, vol. 44, no. 8-9, pp. 1022–1046, 2010.

[21] O. Jahn, R. H. Möhring, A. S. Schulz, and N. E. Stier-Moses, "System-optimal routing of traffic flows with user constraints in networks with congestion," *Operations research*, vol. 53, no. 4, pp. 600–616, 2005.

[22] D. A. Lazar, S. Coogan, and R. Pedarsani, "Capacity modeling and routing for traffic networks with mixed autonomy," in *Proc. IEEE Conf. on Decision and Control*. IEEE Press, 2017, pp. 5678–5683.

[23] A. Houshmand, S. Wollenstein-Betech, and C. G. Cassandras, "The penetration rate effect of connected and automated vehicles in mixed traffic routing," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 1755–1760.

[24] P. T. Harker, "Multiple equilibrium behaviors on networks," *Transportation Science*, vol. 22, no. 1, pp. 39–46, 1988.

[25] H. Yang, X. Zhang, and Q. Meng, "Stackelberg games and multiple equilibrium behaviors on networks," *Transportation Research Part B: Methodological*, vol. 41, no. 8, pp. 841–861, 2007.

[26] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.

[27] R. Chen and C. G. Cassandras, "Optimal assignments in mobility-on-demand systems using event-driven receding horizon control," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[28] X. Bei and S. Zhang, "Algorithms for trip-vehicle assignment in ride-sharing." in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 18, 2018, pp. 3–9.

[29] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for Mobility-on-Demand systems," *Int. Journal of Robotics Research*, vol. 31, no. 7, pp. 839–854, 2012.

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

[31] K. Spieser, S. Samaranayake, W. Gruel, and E. Frazzoli, "Shared-vehicle mobility-on-demand systems: A fleet operator's guide to rebalancing empty vehicles," in *Annual Meeting of the Transportation Research Board*, no. 16-5987, 2016.

[32] R. M. Swaszek and C. G. Cassandras, "Load balancing in mobility-on-demand systems: Reallocation via parametric control using concurrent estimation," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2148–2153.

[33] M. W. Levin, K. M. Kockelman, S. D. Boyles, and T. Li, "A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application," *n Systems*, vol. 64, pp. 373 – 383, 2017.

[34] S. Hörl, C. Ruch, F. Becker, E. Frazzoli, and K. W. Axhausen, "Fleet control algorithms for automated mobility: A simulation assessment for Zurich," in *Annual Meeting of the Transportation Research Board*, 2018, pp. 18–02 171.

[35] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone, "Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms," *Autonomous Robots*, vol. 42, no. 7, pp. 1427–1442, 2018.

[36] M. Salazar, F. Rossi, M. Schiffer, C. H. Onder, and M. Pavone, "On the interaction between autonomous mobility-on-demand and the public transportation systems," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2262–2269.

[37] M. Salazar, N. Lanzetti, F. Rossi, M. Schiffer, and M. Pavone, "Intermodal autonomous mobility-on-demand," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[38] F. Rossi, R. Iglesias, M. Alizadeh, and M. Pavone, "On the interaction between Autonomous Mobility-on-Demand systems and the power network: Models and coordination algorithms," *IEEE Transactions on Control of Network Systems*, pp. 384–397, 2018.

[39] T. A. Manual, "Bureau of public roads," *US Department of Commerce*, 1964.

[40] K. Solovey, M. Salazar, and M. Pavone, "Scalable and congestion-aware routing for autonomous mobility-on-demand via frank-wolfe optimization," in *Robotics: Science and Systems*, 2019.

[41] M. Salazar, M. Tsao, I. Aguiar, M. Schiffer, and M. Pavone, "A congestion-aware routing scheme for autonomous mobility-on-demand systems," in *European Control Conference*, 2019, pp. 3040–3046.

[42] S. Wollenstein-Betech, M. Houshmand, A. Salazar, M. Pavone, C. Cassandras, and I. Paschalidis, "Congestion-aware Routing and Rebalancing of Autonomous Mobility-on-Demand Systems in Mixed Traffic," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2020, pp. 2293–2299.

[43] M. J. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*. Yale University Press, 1955.

[44] S. Wollenstein-Betech, C. Sun, J. Zhang, and I. Paschalidis, "Joint Estimation of OD Demands and Cost Functions in Transportation Networks from Data," in *Proc. IEEE Conf. on Decision and Control*, 2019.

[45] S. Wollenstein-Betech, C. Cassandras, and I. Paschalidis, "Joint Pricing and Rebalancing of Autonomous Mobility-on-Demand systems," in *Proc. IEEE Conf. on Decision and Control*, 2020, pp. 2573–2578.

[46] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research." *Proc. of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, 1952.

[47] Y. A. Korilis, A. A. Lazar, and A. Orda, "Achieving network optima using stackelberg routing strategies," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 161–173, 1997.

[48] D. Bertsekas, *Nonlinear programming*, 3rd ed. Athena Scientific, 2016.

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

[50] OpenStreetMap, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org , 2017.

[51] NYC taxi and Limousine Commission, "Taxi and limousine commission trip record data," https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2020.

[52] S. Wollenstein-Betech, "mixed-traffic-amod-route-rebalance," https://github.com/salomonw/mixed-traffic-amod-route-rebalance, Jan. 2020.

[53] E. Fitzsimmons and W. Hu, "The downside of ridehailing: more new york city gridlock," *New York Times*, 2017.

[54] R. M. Swaszek and C. G. Cassandras, "Receding horizon control for station inventory management in a bike-sharing system," *IEEE Transactions on Automation Sciences and Engineering*, vol. 17, no. 1, pp. 407–417, 2019.

[55] T. Larsson and M. Patriksson, "An augmented lagrangean dual algorithm for link capacity side constrained traffic assignment problems," *Transportation Research Part B: Methodological*, vol. 29, no. 6, pp. 433–455, 1995.

[56] M. Salazar, D. Paccagnan, A. Agazzi, and W. P. M. H. Heemels, "Urgency-aware optimal routing in repeated games through artificial currencies," *arXiv preprint arXiv:2011.11595*, 2020.