

A Vehicle Coordination and Charge Scheduling Algorithm for Electric Autonomous Mobility-on-Demand Systems

Felix Boewing^{1,2}, Maximilian Schiffer³, Mauro Salazar² and Marco Pavone²

Abstract—This paper presents an algorithmic framework to optimize the operation of an Autonomous Mobility-on-Demand system whereby a centrally controlled fleet of electric self-driving vehicles provides on-demand mobility. In particular, we first present a mixed-integer linear program that captures the joint vehicle coordination and charge scheduling problem, accounting for the battery level of the single vehicles and the energy availability in the power grid. Second, we devise a heuristic algorithm to compute near-optimal solutions in polynomial time. Finally, we apply our algorithm to realistic case studies for Newport Beach, CA. Our results validate the near optimality of our method with respect to the global optimum, whilst suggesting that through vehicle-to-grid operation we can enable a 100% penetration of renewable energy sources and still provide a high-quality mobility service.

I. INTRODUCTION

CURRENTLY, personal urban mobility is undergoing a paradigm shift which can be attributed to two major trends. First, ride-hailing companies such as Uber and Lyft are gaining momentum and are increasingly replacing taxi and car-sharing fleets, as well as public transport [1]. Second, major players are boosting the development of self-driving cars, with first fleets envisioned to be operable in 2025 [2]. In this context, experts foresee Autonomous Mobility-on-Demand (AMoD) systems as a central element of future mobility, especially in metropolitan areas. In such systems, fleets of self-driving cars are coordinated by a central operator and provide on-demand mobility in line with the emerging one-way car-sharing paradigm [3]. Especially in urban scenarios, where municipal authorities are beginning to impose new taxi vehicles to be zero-emission [4], AMoD fleets are expected to be battery electric.

Bearing various potential benefits, this setting also yields an inherent complexity resulting from the interdependencies between the transportation network and the power network, since the charging activities of a whole fleet can cause significant loads on the power network. Hence, besides operating a fleet to serve customers' transportation requests, an operator must consider the vehicles' interactions with the power network. Furthermore, from a smart grid perspective, idling vehicles could even be used as small decentralized energy storage systems, helping to buffer renewable energy surplus in a vehicle-to-grid (V2G) fashion [5], [6]. In this context, an AMoD operator must take three central decisions: *i*) assigning transportation requests to vehicles, *ii*) rebalancing empty vehicles by pre-emptively moving them

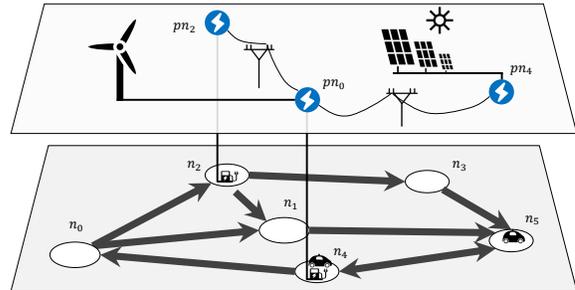


Fig. 1. The electric AMoD system comprises a road network (lower level) and an electric power grid (upper level). In the road network, the arrows represent roads and the white circles denote intersections, charging locations and pick-up points. In this scenario, the power grid connects the charging stations with renewable energy power plants.

where transportation requests are likely to appear, and *iii*) scheduling each vehicle's recharging slots to keep the fleet operational. Critically, the vehicle dispatching tasks *i*) and *ii*) may conflict with charge scheduling decisions *iii*), as a single vehicle can either (dis-)charge its battery or drive. Efficiently solving such a conflict is crucial to enable the operation of an electric AMoD fleet.

In this context, this paper presents an algorithmic framework to jointly optimize the vehicle dispatching decisions and the charging schedules of an electric AMoD system (as shown in Fig. 1), also allowing V2G operation to keep the power grid balanced in the presence of electric demand mismatch. The proposed framework enables us to analyze the potential benefits of such systems in an offline fashion and provides a basis to devise online control algorithms.

Related literature: Our work intersects with two different research streams, namely, routing and charge scheduling algorithms for electric vehicles, as well as recent works on AMoD systems. In the following, we review these fields.

The majority of works in the field of electric vehicle routing and charge scheduling has been focusing on logistics fleets, studying several variants of the vehicle routing problem and metaheuristic solution approaches to solve such NP-hard combinatorial problems. We refer the interested reader to [7] for an overview of these works. Besides them, there are some publications that are more closely related to our work. The authors of [8] developed a heuristic algorithm for the electric traveling salesman problem with time windows considering a single vehicle without power grid interdependencies. Energy constrained shortest paths for a single vehicle including charging stops were efficiently computed in [9] leveraging contraction hierarchies. The authors of [10] analyzed the interplay between the power grid and an electric fleet but assumed vehicle routes to be prefixed. Algorithms for large-scale fleet dispatching were studied for ride-hailing

¹Visiting student researcher at the Autonomous Systems Lab³ and M.Sc. student at ETH Zürich, Switzerland, boewingf@ethz.ch

²Autonomous Systems Lab, Stanford University, Stanford, United States, {samauro, pavone}@stanford.edu

³TUM School of Management, Technical University of Munich, 80333 Munich, Germany, schiffer@tum.de

fleets in [11] and for car-sharing fleets in [12]. Only [11] and [12] focus on large-scale fleet dispatching but are limited to conventional vehicles.

Focusing on the field of AMoD systems, most publications present mesoscopic analyses based on continuous network flow models. Some works focus on the interaction between AMoD and public transport [13], [14] or on congestion-aware routing [15]. There are only few publications available that focus on the interaction between an AMoD fleet and the power grid. The authors of [16] developed a network flow modeling approach that considers the power transmission network, while the power distribution network was included in [17]. Finally, an online controller for vehicle rebalancing and recharging was presented in [18]. However, these approaches rely on heavily aggregated transportation networks and are not amenable to fleet coordination in a microscopic vehicle-centric fashion.

Statement of contributions: To close the research gap among microscopic single-vehicle approaches, conventional fleet routing approaches, and mesoscopic system-level approaches, we present a framework to solve an integrated vehicle dispatching and charge scheduling problem for an electric AMoD fleet. This framework preserves the system-level perspective of [16] but vastly enhances its computational tractability for large-sized applications by integrating recently developed concepts to reduce the complexity of vehicle dispatching [12] and charge scheduling [9] problems. Specifically, our contribution is threefold: First, we present a mixed-integer linear program (MILP) that integrates vehicle dispatching and recharging decisions for an electric AMoD fleet, coupling the transportation system with the power grid, and capturing V2G operations to balance the power grid in the presence of electricity demand mismatch. Second, we develop a heuristic algorithm to compute near-optimal solutions in polynomial time. Finally, we validate the performance of our algorithm with a realistic case-study for Newport Beach, CA. Our results suggest that an optimized V2G operation can enable a complete penetration of renewable energy sources without affecting the mobility service quality.

Organization: The remainder of this paper is structured as follows: We present a MILP to formally define our planning problem in Section II. Then, we develop an approximation algorithm to compute near-optimal solutions in polynomial time in Section III. Section IV details our case study and presents numerical results. Section V concludes this paper with a summary and an outlook on future research.

II. MODEL DESCRIPTION

This section outlines a model jointly capturing the vehicle coordination and charging scheduling problem for an electric AMoD system including its interconnection with the power grid. In particular, we first introduce a graph representation capturing the tasks of the AMoD fleet in Section II-A and leverage it in Section II-B to formulate the vehicle dispatching problem as a MILP. Thereafter, we extend the model in Section II-C to account for energy consumption, battery charging and V2G operation. Finally, we discuss assumptions and model limitations in Section II-D.

A. Graph Representation

We consider a road network modeled as a directed graph $G_r = (V_r, A_r)$ with a set of vertices V_r and a set of arcs $A_r \subseteq V_r \times V_r$. Each vertex $v \in V_r$ represents a road intersection, a charging station (CS), or a customer's pick-up or drop-off point. We refer to the set of CS vertices as $N \subseteq V_r$. Each arc $(n_1, n_2) \in A_r$ represents a road between n_1 and n_2 , associated to a fixed travel time $T_{n_1 n_2}$ and an energy consumption $E_{n_1 n_2}$.

In this network, we model customer transportation demand as a set of trips S . A trip is defined as a triple $s = (o_s, d_s, t_{s,s}) \in S$ containing its origin and destination $o_s \in V_r$ and $d_s \in V_r$, respectively, and its start time $t_{s,s}$. Implicitly, the end time of a trip results to $t_{e,s} := t_{s,s} + T_{o_s d_s}$, whereby the travel time $T_{o_s d_s}$ results from the shortest path completing the trip and is considered to be fixed. We denote the energy consumed on trip s as $E_{\text{fix},s}$. The AMoD operator controls a fleet modeled as a set of vehicles $i \in I$. Each vehicle starts its route at an origin o_i at the beginning of the planning horizon with an initial state of charge (SoC) $E_{\text{init},i}$ and stops immediately after finishing the last trip. In the following, we introduce a graph representation similar to [12] that captures precedence constraints for vehicle to job allocations in the graph itself (see Fig. 2). This way, we reduce the computational complexity of the resulting MILP.

To extend this concept for charging stops, we introduce a directed multigraph $G_s := (V_s, A_s)$. In this graph, a vertex represents either a trip request $s \in S \subseteq V_s$ or a vehicle's initial location $i \in I \subseteq V_s$. Additionally, we add a dummy source O which is connected to the initial vehicle locations in I . Arcs in G_s represent time-related precedence constraints when serving different requests, i.e., an arc (u, v, n) denotes that one vehicle can serve request u and request v , while visiting CS $n \in N$ after finishing u and before starting v (see Fig. 3). Here, $(u, v, 0)$ stands for a direct relocation from the destination of u to the origin of v without visiting a CS. Specifically, an arc $(u, v, 0)$ with a starting time $t_{s,uv0} := t_{e,u}$ and end time $t_{e,uv0} := t_{s,v}$ exists if $t_{s,uv0} + T_{d_u o_v} < t_{e,uv0}$. An arc with charging stop (u, v, n) exists if $t_{s,uvn} + T_{d_u n} + T_{no_v} < t_{e,uvn}$. Collectively, A_s denotes the set of feasible relocations in-between trips created with a k -neighborhood search by connecting each trip to the closest k succeeding and preceding trips with one non-charging arc and, if possible, at least one charging arc per trip. Fig. 4 shows an example of the extension of G_s with charging operations.

We define the relocation energy E_{uvn} as the change of SoC from the end of trip u to the end of trip v : It is the sum of the fixed consumption $E_{\text{fix}1,uvn}$ used to reach the CS n , the energy recharged $E_{\text{ch},uvn}$, the energy to drive from n to the next customer $E_{\text{fix}2,uvn}$ and the energy to take her to destination $E_{\text{fix},v}$. For the case $n = 0$, the energy terms $E_{\text{ch},uv0}$ and $E_{\text{fix}2,uv0}$ are zero, whereas $E_{\text{fix}1,uv0}$ captures the energy to directly transfer from the drop-off point of u to the pick-up location of v . Fig. 3 illustrates these energy components.

For each trip $u \in S$, it holds that $t_{s,u} < t_{e,u}$. Similarly, for each $(u, v, n) \in A_s$ it holds that $t_{s,uvn} < t_{e,uvn}$. Therefore, there exist no paths in G_s that go backwards in time and G_s is *acyclic*. This allows us to reformulate our problem as a minimum cost maximum flow problem (cf. [12]), which can be solved in polynomial time [19].

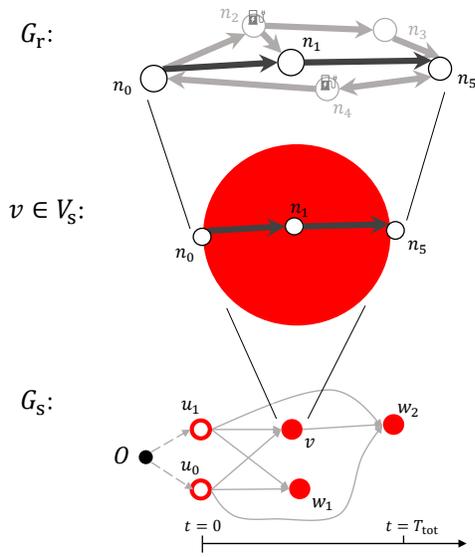


Fig. 2. Transformation of the road graph (upper graph) to the schedule graph G_S (lower graph): A route in the road graph from n_0 to n_5 (upper plot in black) is converted into the node $v \in V_S$ (in the middle) and added as a node $v \in V_S$ of the schedule graph (below). In this case, the trips w_1 and w_2 are constructed in a similar fashion, whilst u_0 and u_1 are the initial locations of the vehicles and O is the artificial source node.

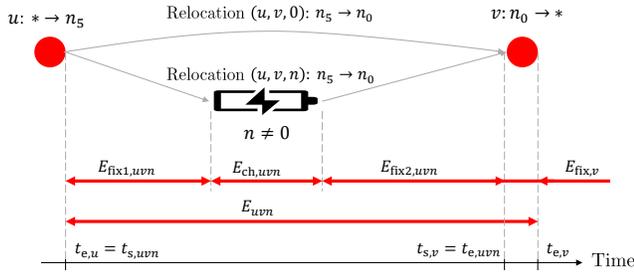


Fig. 3. A charging relocation (u, v, n) and a non-charging relocation $(u, v, 0)$ between two trips $u, v \in V_S$. The energy and time of a relocation are defined between the ends of the two consecutive trips.

B. Vehicle Coordination Problem

To frame the vehicle coordination problem, we introduce binary variables x_{uvn} indicating whether an arc (u, v, n) is traversed ($x_{uvn} = 1$) or not ($x_{uvn} = 0$) and maximize the number of serviced customers subject to flow conservation constraints as follows.

Problem 1 (Vehicle Coordination):

$$\max_x \sum_{v \in V_S} \sum_{u, n: (u, v, n) \in A_S} x_{uvn} \quad (1)$$

$$\text{s.t.} \quad \sum_{u, n: (u, v, n) \in A_S} x_{uvn} \leq 1 \quad \forall v \in V_S \quad (2)$$

$$\mathbb{1}_{v \in I} + \sum_{u, n: (u, v, n) \in A_S} x_{uvn} \geq \sum_{w, n: (v, w, n) \in A_S} x_{vwn} \quad \forall v \in V_S \quad (3)$$

$$x_{uvn} \in \{0, 1\} \quad \forall (u, v, n) \in A_S. \quad (4)$$

The objective (1) maximizes the number of traversed arcs, which corresponds to servicing as many customers as possible. Constraint (2) ensures that at most one vehicle serves a single trip, while we guarantee that each route is continuous and starts at a vehicle's initial location with (3).

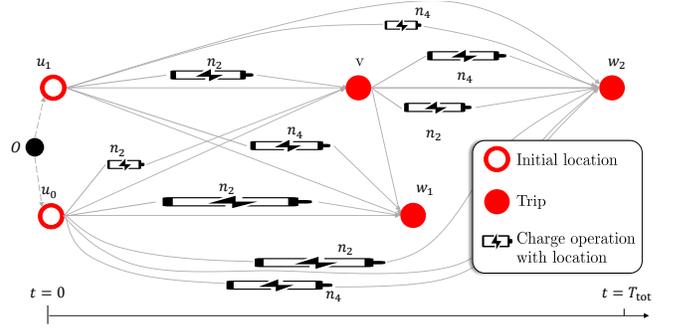


Fig. 4. Example of a schedule graph with vehicles u_0 and u_1 and rides v, w_1 and w_2 to serve. The battery symbols on an arc indicates a charge operation. The length of the battery symbol is proportional to the time spent at the CS.

C. Energy Model

In the following, we model the charging process and the interaction of the vehicles with the power grid. We use a discretized time horizon with time steps $t \in \mathcal{T}$ denoting all points in time at which a charging operation in G_S can start or end and the power flow can change. Each CS n adds a load to the power network at time step t when consuming a charging load $p_{ch,n}(t)$, which is negative in case of V2G operation. The total charging load in the grid is given by

$$p_{ch,tot}(t) = \sum_{n \in N} p_{ch,n}(t) \quad \forall t \in \mathcal{T}. \quad (5a)$$

The distribution grid in urban settings is usually over-dimensioned such that power-line flow limits do not have to be actively monitored. Therefore, we assume that power can flow freely between all charging nodes $n \in N$ as long as the demand matches the supply. The total power supplied to the vehicles $p_{ch,tot}(t)$ is limited by the power available from the power grid $p_a(t)$ as

$$p_{ch,tot}(t) \leq p_a(t) \quad \forall t \in \mathcal{T}. \quad (6a)$$

For the sake of brevity, we condense all transmission losses in $p_a(t)$ without further discussion.

In order to map the charging loads $p_{ch,n}(t)$ to charged energies $E_{ch,uvnt}$ for each charging relocation (u, v, n) and time t , we introduce the set of charging relocations $C_{t,n} := \{(u, v, n) | (u, v, n) \in A_S^\dagger(t)\}$, whereby we denote all charging arcs which can charge a vehicle at time t as $A_S^\dagger(t): \mathcal{T} \rightarrow P(A_S), t \mapsto \{a \in A_S | t \in \mathcal{T}_a\}$. The set $C_{t,n}$ contains all arcs (u, v, n) connecting a vehicle to CS n at time t . The charging load $p_{ch,n}(t)$ consists of all charging operations at station n at time t and is limited by thermal constraints:

$$p_{ch,n}(t) = \left(\sum_{(u,v,n) \in C_{t,n}} E_{ch,uvnt} \right) / T_t \quad \forall n \in N, t \in \mathcal{T} \quad (7a)$$

$$p_{ch,min} \leq p_{ch,n}(t) \leq p_{ch,max} \quad \forall t \in \mathcal{T}, n \in N, \quad (7b)$$

where T_t is the duration of time-step t .

To keep track of the energy balance in G_S , we derive aggregated energy values. For a relocation $(u, v, n) \in A_S$ and considering the set of all time-steps in \mathcal{T}_{uvn} during the charging operation of (u, v, n) , we assign a charged energy value $E_{ch,uvn}$ for the whole arc as

$$E_{ch,uvn} = \sum_{t \in \mathcal{T}_{uvn}} E_{ch,uvnt} \quad \forall (u, v, n) \in A_S. \quad (8a)$$

We allow charging (positive or negative) only when a given charging route is assigned to a vehicle with

$$E_{\text{ch},uvnt} \leq b_{\text{max}}x_{uvn} \quad \forall t \in \mathcal{T}_{uvn}, (u, v, n) \in A_s \quad (9a)$$

$$E_{\text{ch},uvnt} \geq -b_{\text{max}}x_{uvn} \quad \forall t \in \mathcal{T}_{uvn}, (u, v, n) \in A_s, \quad (9b)$$

whereby b_{max} is the battery capacity of the vehicles.

The energy consumption of arc (u, v, n) consists of the energy used to go from a customer trip to a CS $E_{\text{fix}1,uvn}$, the energy used to drive from the station to the next customer $E_{\text{fix}2,uvn}$, and of the energy $E_{\text{fix},v}$ consumed during trip v . The overall energy change E_{uvn} over an arc (u, v, n) from the end of trip u to the end of v is therefore

$$E_{uvn} = E_{\text{ch},uvn} - x_{uvn}(E_{\text{fix}1,uvn} + E_{\text{fix}2,uvn} + E_{\text{fix},v}) \quad \forall (u, v, n) \in A_s. \quad (10)$$

To keep the fleet operational, we need to track the SoC for each vehicle at each point in time. For each node $v \in V_s$, we introduce the SoC at the end of v b_v . In order to propagate the SoC through the graph, we introduce the variable y_{uvn} denoting the SoC of a vehicle at the end of (u, v, n) if a vehicle chooses a route with this relocation, and remaining zero otherwise. Specifically, it holds that

$$y_{uvn} = x_{uvn}(b_u + E_{uvn}) \quad \forall (u, v, n) \in A_s,$$

which can be reformulated in linear form as

$$y_{uvn} \leq x_{uvn}b_{\text{max}} \quad \forall (u, v, n) \in A_s \quad (11a)$$

$$y_{uvn} \leq b_u + E_{uvn} \quad \forall (u, v, n) \in A_s \quad (11b)$$

$$y_{uvn} \geq b_u + E_{uvn} - (1 - x_{uvn})b_{\text{max}} \quad \forall (u, v, n) \in A_s \quad (11c)$$

$$y_{uvn} \in [0, b_{\text{max}}] \quad \forall (u, v, n) \in A_s. \quad (11d)$$

The aggregation of y_{uvn} to the SoC b_v and its initialization with the initial SoC $E_{\text{init},i}$ is defined as

$$b_v = \sum_{u,n:(u,v,n) \in A_s} y_{uvn} \quad \forall v \in S \quad (12a)$$

$$b_i = E_{\text{init},i} \quad \forall i \in I. \quad (12b)$$

Finally, we ensure that the SoC of each vehicle stays always between zero and the battery capacity b_{max} as

$$0 \leq b_v + x_{uvn}E_{\text{fix}1,uvn} + \sum_{t \in \mathcal{T}_{uvn}[0:k]} E_{\text{ch},uvnt} \leq b_{\text{max}} \quad \forall (u, v, n) \in A_s, k \in [|\mathcal{T}_{uvn}|] \quad (13a)$$

$$0 \leq b_v + x_{uvn}E_{\text{fix}1,uvn} + E_{\text{ch},uvnt} \leq b_{\text{max}} \quad \forall (u, v, n) \in A_s \quad (13b)$$

$$b_v \in [0, b_{\text{max}}] \quad \forall v \in V_s. \quad (13c)$$

Using these constraints, we can extend Problem 1 to account for energy consumption, charging activities and V2G operation. This yields the optimal vehicle coordination and charge scheduling problem.

Algorithm 1: CEPAMoDS Algorithm

Input: $G_s = (V_s, A_s)$, p_a

Output: Set of feasible Routes \mathcal{R} for all vehicles

- 1 $\forall u, v, n \in A_s : p_{\text{pred},uvn} \leftarrow \text{EnergyPrediction}(G_s)$
 - 2 $\mathcal{R}, P_{\text{ch,tot,routed}} \leftarrow \text{Routing}(G_s, P_{\text{pred}}, p_a, A_s)$
 - 3 $\mathcal{R} \leftarrow \text{Adaptation}(\mathcal{R}, P_{\text{ch,tot,routed}})$
-

Problem 2 (Vehicle Coordination and Charge Scheduling):

$$\begin{aligned} \max_{x, E, E_{\text{ch}}, P_{\text{ch}}, P_{\text{ch,tot}}, b} \quad & \sum_{v \in V_s} \sum_{u,n:(u,v,n) \in A_s} x_{uvn} \quad (14) \\ \text{s.t.} \quad & (2)-(10), (11)-(13). \end{aligned}$$

D. Discussion

A few comments on this modeling approach are in order. First, we assume that at each charging station there are always enough free slots, so that AMoD vehicles do not have to wait in line to recharge their battery. Assuming that most parking places will be electrified in the future, this can be interpreted by the AMoD vehicles having a priority over regular electric cars. Second, we neglect power-line flow limits. This assumption is adequate for urban scenarios, whereby the distribution grid is usually over-dimensioned and active bottle-neck monitoring is not yet required. Finally, we only capture exogenous congestion effects, whilst neglecting the endogenous impact of AMoD vehicles on travel time. This assumption is adequate for small to medium sized fleets [12]. This way, we model exogenous congestion by adjusting the travel time on each road link during the course of the day.

III. SOLUTION ALGORITHM

In this section, we develop a Convolutional Energy Predicting AMoD Scheduler (CEPAMoDS) to approximately solve Problem 2 in polynomial time. Alg. 1 gives an overview of this algorithm consisting of three steps: *i*) predicting minimal energy demands between two subsequent charging stops, *ii*) dispatching vehicles through energy-feasible routes considering the information of step *i*), and *iii*) adapting the solution of step *ii*) to reduce the residual supply-demand mismatch in the power grid via V2G operation.

A. Step i): Prediction

In the first step, for each charging stop on an arc $(u, v, n) \in A_s$ we predict an estimate of the charging load in case the arc is traversed by a vehicle (see Alg. 2). This charging load corresponds to the energy needed to reach the closest succeeding charging stop after v with the same SoC as when reaching the charging stop between u and v (Alg. 2 line 2). For this prediction, we use a graph convolution over a neighborhood of $v \in V_s$ (see Fig. 5), similarly to a convolution of a kernel with an image and a pooling layer in a convolutional neural net [20]. In particular, we replace the image with a graph, the kernel matrix is the identity, and the pooling is a min-pooling layer. Note that the prediction $p_{\text{pred},uvn}$ is purely based on G_s and not on the available charging power p_a .

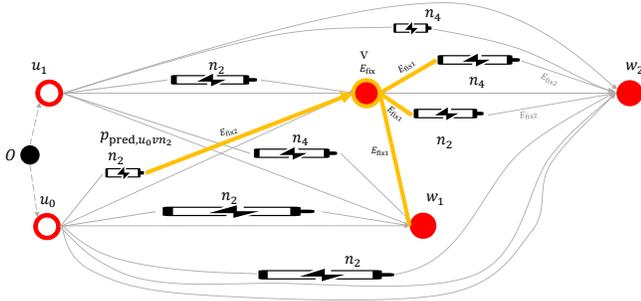


Fig. 5. Prediction step: The yellow arcs highlight the participating arcs for the graph convolution of arc $(u_0, v, n_2) \in A_s$: The energy used while driving on the yellow arcs in the schedule graph determines the power prediction $p_{\text{pred}, u_0 v n_2}$ according to the formula in Alg. 2

Algorithm 2: EnergyPrediction(G_s)

Input: G_s
Output: $p_{\text{pred}, uvn} \quad \forall (u, v, n) \in A_s$
1 **foreach** $u, v, n \in A_s$ **do**
2 $p_{\text{pred}, uvn} := \frac{1}{T_{\text{uvn}}} (E_{\text{fix}2, uvn} + E_v + \min_{w, \tilde{n} \in \text{Successors}(v)} E_{\text{fix}1, v \tilde{n}})$
3 **end**
Result: $p_{\text{pred}, uvn} \quad \forall (u, v, n) \in A_s$

B. Step ii): Routing

In this second step, we determine the route of each vehicle in order to maximize the number of customers' trips. Alg. 3 finds routes in the schedule graph for all vehicles in an iterative manner, whereby one iteration works as follows: First, we calculate the expected global charging load of all not yet routed vehicles $p_{\text{pred}}(t)$ by taking the average of all possible charging loads in $A_s^\dagger(t)$ for time $t \in \mathcal{T}$ and multiplying it by the number of unscheduled vehicles m (Alg. 3, line 6). Second, we calculate the available charging load on each arc $\tilde{p}_{\text{pred}, uvn}(t)$ by correcting the predicted charging load $p_{\text{pred}, uvn}(t)$ from step *i*) in Section III-A (Alg. 3, lines 7–9). This correction considers the total charging load of already routed vehicles $p_{\text{ch}, \text{tot}, \text{routed}}(t)$, the currently available power $p_a(t)$ and the expected charging load of all unscheduled vehicles $p_{\text{pred}}(t)$. Third, we traverse G_s in a breadth-first order starting from the artificial source node O to explore all possible routes (Alg. 4, lines 3–7) and choose the longest route r_{longest} (Alg. 4, lines 9–10). Hereby, O is used to find the longest route among all possible routes of all vehicles. Therefore, it is not necessary to choose which vehicle we explore the possible routes first. At each node $v \in G_s$, we remove non-Pareto-optimal routes (whereby we denote the Pareto optimality of a route in terms of the two objectives *number of trips served* and *energy at the end of the route*). Out of the Pareto optimal routes we store only the l with the highest final SoC (Alg. 4, line 8). Fourth, we add the charging load along r_{longest} to $p_{\text{ch}, \text{tot}, \text{routed}}$ (Alg. 3, lines 11–13). Finally, the nodes traversed by r_{longest} in G_s are removed from V_s together with their adjacent arcs in A_s (Alg. 3, line 14). Fig. 6 shows an example of two iterations of this algorithm, highlighting the longest routes found in green.

C. Step iii): Adaptation

A mismatch between the available power $p_a(t)$ and the assigned charging power $p_{\text{ch}, \text{tot}, \text{routed}}(t)$ might occur after all vehicles have been routed. Assuming a completely re-

Algorithm 3: Routing($G_s, p_{\text{pred}}, p_a, A_s$)

Input: $G_s, \forall u, v, n \in A_s : p_{\text{pred}, uvn}$
Output: \mathcal{R}
1 $\mathcal{R} := \emptyset$
2 $m := |I|$
3 $p_{\text{ch}, \text{tot}, \text{routed}} := 0$
4 **while** $m > 0$ **do**
5 $\forall t \in \mathcal{T} : p_{\text{pred}}(t) := \frac{m}{|A_s^\dagger(t)|} \sum_{u, v, n \in A_s^\dagger(t)} p_{\text{pred}, uvn}(t)$
6 **foreach** $u, v, n \in A_s : t \in \mathcal{T}_{\text{uvn}}$ **do**
7 $\tilde{p}_{\text{pred}, uvn}(t) := p_{\text{pred}, uvn}(t) + \frac{p_a(t) - p_{\text{ch}, \text{tot}, \text{routed}}(t) - p_{\text{pred}}(t)}{m}$
8 **end**
9 $\mathcal{R} \leftarrow r_{\text{longest}} = \text{FeasibleLongestPath}(G_s)$
10 **foreach** $u, v, n, t \in \text{ChargingsAtRoute}(r_{\text{longest}})$ **do**
11 $p_{\text{ch}, \text{tot}, \text{routed}}(t) := p_{\text{ch}, \text{tot}, \text{routed}}(t) + \tilde{p}_{\text{pred}, uvn}(t)$
12 **end**
13 $\text{RemoveRouteFromGraph}(r_{\text{longest}}, G_s)$
14 $m := m - 1$
15 **end**

Algorithm 4: FeasibleLongestPath(G_s)

Input: G_s with energy predictions \tilde{p}_{pred} , max number of Pareto optimal routes $l \in \mathbb{Z}^{>0}$
Output: Feasible longest route r_{longest}
1 $O := \text{GetSource}(G_s), \mathcal{R}_O := \{(E_{\text{init}, O}, \emptyset)\}, \forall u \in V_s : \mathcal{R}_u := \emptyset, r_{\text{longest}} := \emptyset$
2 **foreach** $u, v, n \in \text{BreathFirstOrdered}(O, A_s)$ **do**
3 **foreach** E, r in \mathcal{R}_u **do**
4 // Extend route if feasible
5 **if** $0 < (E + E_{\text{fix}1, uvn}, E + E_{\text{fix}1, uvn} + \tilde{p}_{\text{pred}, uvn}(t)T_{\text{uvn}}, E + E_{\text{fix}, uvn} + \tilde{p}_{\text{pred}, uvn}(t)T_{\text{uvn}}) < b_{\text{max}}$ **then**
6 $\mathcal{R}_v \leftarrow (E + E_{\text{uvn}}, r \circ v)$
7 **end**
8 $\mathcal{R}_v := \text{ParetoOptimalRoutes}(\mathcal{R}_v, l)$
9 **if** $\text{LongestRouteLength}(\mathcal{R}_v) > |r_{\text{longest}}|$ **then**
10 $r_{\text{longest}} := \text{LongestRoute}(\mathcal{R}_v)$
11 **end**
Result: r_{longest}

newable energy production, we devise Alg. 5 to minimize this mismatch and enable active V2G operation. In general, a mismatch may arise for two reasons: First, there might be residual power available in the power grid, i.e., $p_a(t) > p_{\text{ch}, \text{tot}, \text{routed}}(t)$, inducing power plant curtailment. In this case, the algorithm increases the charging load of all charging operations up to the maximum allowed power (Alg. 7 lines 2–3), because the marginal cost of solar and wind power is zero. Second, the available renewable energy production is less than the non-vehicle demand in the grid, i.e., $p_a(t) < 0$. In this case, the fleet is requested to inject power back into the grid. However, Alg. 4 would never choose routes along charging stations under these conditions as it still tries to maximize the length of the routes. Therefore, we actively introduce detours to charging stations to enable V2G operation and balance the power grid (Alg. 5, lines 6–10). In these detours, vehicles feed energy back to the grid, whilst still preserving SoC constraints for their overall route as well as respecting the power limits of each CS.

D. Computational Complexity

Alg. 2 iterates once over all arcs A_s , yielding a complexity of $\mathcal{O}(|A_s|)$. Alg. 3 iterates over A_s once for every vehicle in I . Alg. 4 checks for all arcs $(u, v, n) \in A_s$ which routes at node u in the set of Pareto optimal routes \mathcal{R}_u have a feasible extension with v via CS n . As there are l possible routes at most, Alg. 4 has a complexity of $\mathcal{O}(|A_s|)$, with $l \ll |A_s|$ being neglected as it is instance-independent. Alg. 3

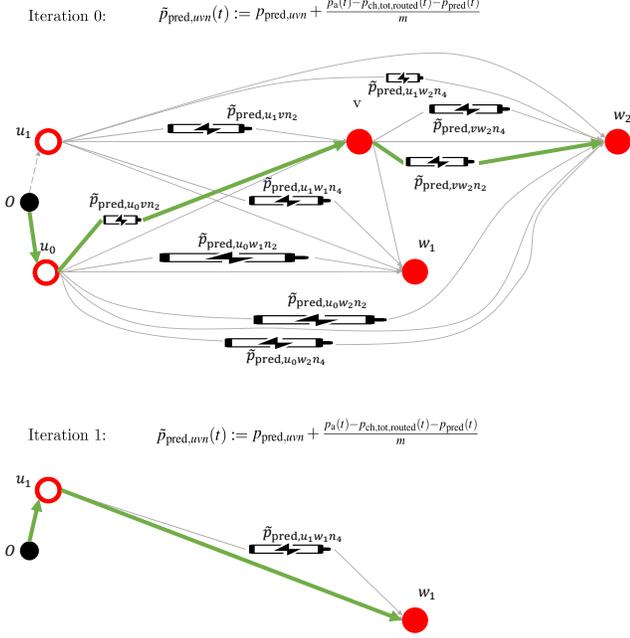


Fig. 6. Routing step: For all vehicles the energy-feasible routes based on the corrected charging loads on each arc \tilde{p}_{pred} are calculated. Then the nodes along the longest route are removed from the graph. This process repeats for the remaining nodes and vehicles in the graph until all vehicles are routed or all nodes are visited, i.e., all trips are completed. Note that u_0 is not necessarily routed first: If u_1 would reach a longer route, the algorithm would route u_1 first.

Algorithm 5: Adaptation($\mathcal{R}, p_{\text{ch,tot,routed}}$)

Input: $G_s, \forall u, v, n \in A_s : p_{\text{pred},uvn}, p_{\text{ch,tot,routed}}(\cdot)$
Output: Set of routes \mathcal{R} for all vehicles

```

1 foreach  $r \in \mathcal{R}$  do
2   foreach  $u, v, n, t \in \text{ChargingsAtRoute}(r)$  do
3      $r := \text{IncreaseCharging}(r, p_a(t) - p_{\text{ch,tot,routed}}(t), (u, v, n, t))$ 
4   end
5 end
6 foreach  $r \in \mathcal{R}$  do
7   foreach  $u, v, n, t \in r$  where  $p_a(t) < p_{\text{ch,tot,routed}}(t)$  do
8      $r :=$ 
9        $\text{MakeDetourToCharger}(r, p_a(t) - p_{\text{ch,tot,routed}}(t), (u, v))$ 
10  end
11 end
```

calls Alg. 4 once for every vehicle in I . Accordingly, the complexity of Alg. 3 is $\mathcal{O}(|A_s||I|)$. Alg. 5 iterates over the relocations of each route and of each vehicle. We denote the maximal possible length of a route by R_{longest} . Since the number of routes is $|I|$, the complexity of Alg. 5 is $\mathcal{O}(R_{\text{longest}}|I|)$. Finally, as a whole, the CEPAMoDS Alg. 1, combines Algorithms 2, 3, and 5. Since in Alg. 3 every arc of the graph has to be visited once, whereas Alg. 5 visits only the traversed arcs (a subset of all arcs), the overall complexity is dominated by Alg. 3. Therefore, the CEPAMoDS Alg. 1 has a complexity of $\mathcal{O}(|A_s||I|)$, which is bilinear in the number of possible relocations and in the number of vehicles.

IV. COMPUTATIONAL STUDIES

In this section, we present computational studies to analyze the performance of our algorithm for synthetic bench-

Algorithm 6: ChargingAtRoute(r)

Input: Route r
Output: Set of arcs C which contain a charging operation

```

1  $C := \emptyset$ 
2 foreach  $(u, v, n) \in r$  do
3   if  $n \neq 0$  then
4      $C \leftarrow (u, v, n)$ 
5   end
6 end
7 Result:  $C$ 
```

Algorithm 7: IncreaseCharging($r, p_{\text{residual}}, (u, v, n, t)$)

Input: $r, p_{\text{residual}}(t), (u, v, n, t)$
Output: Route with increased charging load r

```

1 if  $p_{\text{residual}}(t) > 0$  then
2    $b_{uvnt} = \text{SocBeforeCharging}((u, v, n, t), r)$ 
3    $p_{\text{add}} = \min(p_{\text{ch,max}} - p_{\text{ch},uvn}(t), (b_{\text{max}} - b_{uvnt})/T_t, p_{\text{residual}})$ 
4    $p_{\text{ch},uvn}(t) = p_{\text{ch},uvn}(t) + p_{\text{add}}$ 
5    $r = \text{SetChargingPower}(r, t, p_{\text{ch},uvn}(t))$ 
6 end
7 Result:  $r$ 
```

marking scenarios and realistic case studies. We first introduce all the scenarios in Section IV-A and present numerical results in Section IV-B.

A. Experimental Design

We analyze the hypothetical deployment of an electric AMoD fleet in Newport Beach, CA. The road network data is taken from OpenStreetMap [21] and has 3575 nodes. We developed four different case studies, summarized in Table I. We define the total available (+) and requested energy (-) in the power grid as

$$E_{a,\text{tot}}^{\pm} := \pm \int_0^{T_{\text{tot}}} \max\{0, \pm p_a(t)\} dt,$$

respectively, where T_{tot} is the time length of a scenario.

a) *Synthetic Cases:* We use the first two scenarios to benchmark our algorithm against a state-of-the-art MILP solver. We randomly generate 100 trips within a time-frame of 3 h that need to be serviced by a fleet of 20 vehicles with a battery capacity of 50 kWh, a low initial SoC randomly chosen between 3 and 7 kWh, and a constant available power $p_a(t)$. We construct a *high-energy* scenario whereby the available power is high, namely, $p_a(t) = 300$ kW, and a *low-energy* scenario with a scarcer power availability of $p_a(t) = 85$ kW. In the former case, we benchmark the vehicle coordination capability of our algorithm, while in the latter case we stress-test our charge scheduling approach.

b) *Realistic 24h Cases:* In this realistic case-study we optimize the day-long operation of an electric AMoD fleet, considering data for July 25, 2018. We took traffic data from TomTom [22] and generated a *small* case with 250 trips and a *large* one with 750 trips. For both cases we scheduled the trips according to the traffic intensity during the course of the day. These scenarios correspond to about 3% and 9% of the demand of the area, respectively. We empirically chose a fleet of 15 and 45 vehicles, which is sufficient to service nearly all customers. We set the battery size to 50 kWh (which lies in-between a Tesla Model 3 and a BMW i3) and we randomly chose an initial SoC between 10 and 20 kWh. Using data for California from CAISO [23], we took the available power $p_a(t)$ as the difference between a sixfold of

TABLE I
PARAMETERS OF THE CASE STUDIES.

Case Study	Trips	Cars	Time	$p_a(t)$ [kW]	$E_{a,\text{tot}}^\pm$ [kWh]
Synthetic <i>high-energy</i>	100	20	3 h	300	0, 900
Synthetic <i>low-energy</i>	100	20	3 h	85	0, 255
Realistic <i>small</i>	250	15	24 h	$\in [-21, 61]$	-129, 525
Realistic <i>large</i>	750	45	24 h	$\in [-63, 183]$	-387, 1575

TABLE II
SYNTHETIC CASE STUDIES: THE CEPAMODS (CEPA) SOLUTION
AND THE GLOBALLY OPTIMAL SOLUTION FROM THE MILP.

Synthetic Case Study	Runtime		Trips Served	
	CEPA	MILP	CEPA	MILP
<i>High-energy</i>	13 s	4.6 h	91%	95%
<i>Low-energy</i>	11 s	4.3 h	88%	97%

the renewable-only generation of the day (whereby we assumed no conventional power plants) and the non-transport-related electricity demand. This available power would allow to power every vehicle in California by renewable electric energy. By doing so, we consider a scenario with 100% penetration of renewables, whereby we empirically scaled down the power grid to a level which can keep the average SoC of the chosen fleet balanced. Finally, for the *large* case, we also study fleets equipped with smaller batteries.

B. Results

Table II and III summarize the numerical results obtained for each experiment. For the synthetic cases, we used Gurobi 8.1 as the benchmark MILP solver. All experiments ran on an i5 2.5 GHz processor with 8 GB of memory. In the following, we discuss each case individually.

a) *Synthetic Cases*: As shown in Table II, the proposed CEPAMoDS Alg. 1 solved Problem 2 for both synthetic cases in less than 15 seconds, whereas the state-of-the-art MILP solver took more than 4 hours. Notably, the suboptimality gap is below 5% for the *high-energy* scenario, and below 10% for the *low-energy* scenario, underlining the impact of energy availability on the problem complexity. Overall, these scenarios showed that our algorithm can compute near-optimal solutions a thousand times faster than state-of-the-art MILP solvers.

b) *Realistic 24h Cases*: These realistic case studies could not be solved as MILPs due to the problem size. Conversely, as shown in Table III, the CEPAMoDS algorithm solved Problem 2 in about 4 minutes for the *small* case and in approximately 2 to 3 hours for the *large* cases.

As shown in Section III-D, the algorithmic complexity is bilinear in the number of cars and in the number of arcs of G_s . Since the number of cars was increased by factor of 3 and the number of arcs by factor of 9 (due to the increase in number of nodes and neighbors in the k -neighborhood by a factor of 3 each), our analysis predicts a computational complexity increase by a factor of 27. This is in line with the increase in computational time by a factor of 30 to 40.

Fig. 7 shows a snapshot of the vehicles' operation and the total power drawn from the grid for the *large* case study with 100% battery size (the *small* case study achieved very similar results – a link to the full video is provided in Section VI). The charging power $p_{\text{ch,tot}}(t)$ matches the available power

TABLE III
CEPAMODS SOLUTION FOR THE REALISTIC CASE STUDIES

Scenario	Battery Size	Runtime [min]	Trips Served	$E_{\text{curtailed}}$ [kWh]	E_{deficit} [kWh]	E_{adapt} [kWh]
<i>Small</i>	100%	4	90%	0	0.5	45
<i>Large</i>	100%	147	93%	0.1	0.7	151
<i>Large</i>	50%	146	93%	0.2	2.7	161
<i>Large</i>	20%	163	84%	159	0	219
<i>Large</i>	10%	128	73%	417	159	131

$p_a(t)$ both when the latter is positive and negative, i.e., the necessary curtailment of the power plants

$$E_{\text{curtailed}} := \int_{t:p_a(t) \geq 0} \max\{0, p_a(t) - p_{\text{ch,tot}}(t)\} dt$$

and the energy missing in the grid during V2G operation

$$E_{\text{deficit}} := \int_{t:p_a(t) < 0} \max\{0, p_{\text{ch,tot}}(t) - p_a(t)\} dt$$

are both almost zero (see Table III). This implies that the intra-day volatility of the power grid resulting from the 100% renewable energy sources can be balanced solely with the batteries of AMoD vehicles and no significant stationary grid storage is needed. Critically, this is enabled by the adaptation step of Section III-C, which increases the charged energy by

$$E_{\text{adapt}} := \int_{t:p_a(t) \geq 0} p_{\text{ch,tot}}(t) - p_{\text{ch,tot,routed}}(t) dt$$

and introduces V2G detours in order to lower the energy deficit E_{deficit} to zero. The increase of charged energy E_{adapt} in the adaptation step is especially useful in scenarios with a small battery size, where charge scheduling is critical.

Notably, the same results in terms of mobility service and grid balancing were observed also for the *large* scenario with 50% the battery size. Further decreasing the battery size to 20% worsened the number of customers serviced by 9% and wasted about 10% of available energy, due to lacking storage and consequent power plant curtailment. Nevertheless, the adaptation step still enabled full V2G operation. Finally, a fleet with 10% of the battery size could only service 73% of the travel demand. Moreover, it was no longer able to balance the grid with V2G operation due to lacking energy in the vehicles, which led to an energy deficit of more than 40% of the requested energy $E_{a,\text{tot}}^-$. Overall, these case studies suggest that the battery size of the vehicles could be halved without causing service performance losses.

V. CONCLUSION

In this paper we explored the possibility of jointly optimizing the vehicle routes and the charging schedules of a fleet of electric self-driving cars providing on-demand mobility. The proposed model combines task-allocation methods for the coordination of AMoD fleets with energy-constrained longest path approaches for single vehicles, and can be framed as a mixed-integer linear program (MILP). To overcome scalability issues, we devised an algorithm that is able to approximately solve the presented vehicle coordination and charge scheduling problem in polynomial time. Specifically, our numerical results empirically showed that for both benchmark cases the quality of our solution is not more than 10% lower than the globally optimal one. Moreover, our

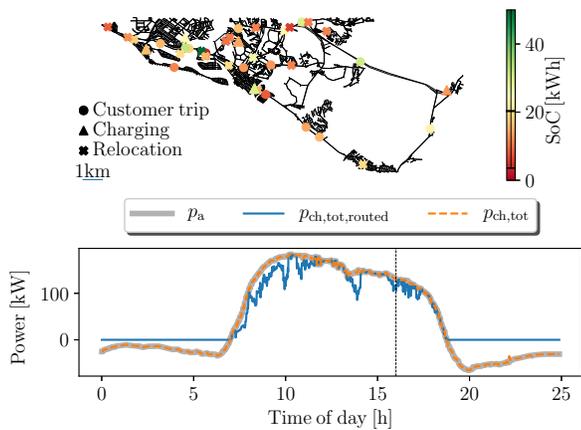


Fig. 7. Results for the realistic case study with 45 vehicles equipped with a 50 kWh battery and 750 trips. The map shows a snapshot at 4pm of the vehicles' positions, their current task (indicated by their shape) and SoC level (indicated by their color). In addition, the color-bar shows the average, maximum and minimum SoC level of the fleet. The plot below shows the charging load before ($p_{ch,tot,routed}(t)$, solid blue) and after ($p_{ch,tot}(t)$, dashed orange) the adaptation step (Sec. III-C), whereby the charging load $p_{ch,tot}(t)$ follows the available power $p_a(t)$ (solid grey). A link to a full video of the case study can be found in Section VI at the end of the paper.

algorithm is about a thousand times faster than state-of-the-art MILP solvers. Finally, we investigated a realistic case study suggesting that an electric fleet could be used as a free-floating energy storage system to completely enable a full penetration of renewable energy sources in the power grid, whilst still providing a high-quality mobility service.

This work can be extended in several directions. First, we would like to provide theoretical guarantees on problem feasibility (in terms of grid balancing) and on solution sub-optimality, also computing an upper bound on the total execution time. Second, we are interested in extending the model to account for power-line flow limits and voltage limitations on the grid. Third, it would be of interest to improve the computational performance of the algorithm by leveraging its highly parallelizable structure. Fourth, we plan to devise a real-time model predictive controller by implementing this algorithmic framework in a receding-horizon fashion, potentially capturing stochastic phenomena such as uncertain travel demand and electricity production. Finally, we would like to leverage this approach to perform sensitivity analysis on the system characteristics, including charging infrastructure placement and achievable vehicle speed, and include it in a co-design framework [24].

VI. MEDIA MATERIAL

A video with the results of the realistic case study can be found at the following link: vimeo.com/362510230

ACKNOWLEDGMENTS

We would like to thank Dr. K. Solovey, D. Scott, J. T. Luke, D. Jalota and S. Laamimach for insightful discussions. Furthermore, we thank Dr. I. New for proofreading this paper. The first author would like to express his gratitude to the Bakala Foundation for their support.

This research was supported by the National Science Foundation under CAREER Award CMMI-1454737 and CPS Award-1837135, and the Toyota Research Institute

(TRI). This article solely reflects the opinions and conclusions of its authors and not NSF, TRI, or any other entity.

REFERENCES

- [1] P. Berger. (2018) Mta blames uber for decline in new york city subway, bus ridership. The Wall Street Journal. Available online.
- [2] J. D. Power. (2019) Mobility pipe dreams? j. d. power and survey-monkey uncover shaky consumer confidence about the future. J. D. Power. Available at <https://www.jdpower.com/business/press-releases/2019-mobility-confidence-index-study-fueled-surveymonkey-audience>.
- [3] M. Pavone, "Autonomous Mobility-on-Demand systems for future urban mobility," in *Autonomes Fahren*. Springer, 2015.
- [4] P. Collinson, "London's ultra-low emission zone: what you need to know," *The Guardian*, 2019, available online at <https://www.theguardian.com/politics/2019/jan/05/londons-ultra-low-emission-zone-what-you-need-to-know>.
- [5] W. Kempton and J. Tomić, "Vehicle-to-grid power fundamentals: Calculating capacity and net revenue," *Journal of Power Sources*, vol. 144, no. 1, pp. 268–279, 2005.
- [6] —, "Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy," *Journal of Power Sources*, vol. 144, no. 1, pp. 280–294, 2005.
- [7] M. Schiffer, M. Schneider, G. Walther, and G. Laporte, "Vehicle routing and location-routing with intermediate stops: A review," *Transportation Science*, vol. 53, no. 2, 2019.
- [8] R. Roberti and M. Wen, "The electric traveling salesman problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 89, pp. 32–52, 2016.
- [9] M. Baum, J. Dibbelt, D. Gamsa, A. Wagner, and T. Zündorf, "Shortest feasible paths with charging stops for battery electric vehicles," in *ACM SIGSPATIAL*, 2015.
- [10] B. Scheiper, M. Schiffer, and G. Walther, "The flow refueling location problem with load flow control," *Omega: The International Journal of Management Science*, vol. 83, pp. 50–69, 2019.
- [11] D. Bertsimas, P. Jaillet, and S. Martin, "Online vehicle routing: The edge of optimization in large-scale applications," *Operations Research*, vol. 67, no. 1, pp. 143–162, 2019.
- [12] F. Rüdél, M. Schiffer, and G. Walther, "User-based relocation strategies in free-floating car sharing systems," 2019, in preparation.
- [13] 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, Extended Version, Available at <https://arxiv.org/abs/1804.11278>.
- [14] M. Salazar, N. Lanzetti, F. Rossi, M. Schiffer, and M. Pavone, "Intermodal autonomous mobility-on-demand," *IEEE Transactions on Intelligent Transportation Systems*, 2019, in press.
- [15] 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, in press.
- [16] 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*, 2018, in Press. Extended version available at <https://arxiv.org/abs/1709.04906>.
- [17] A. Estandia, M. Schiffer, F. Rossi, E. C. Kara, R. Rajagopal, and M. Pavone, "On the interaction between autonomous mobility on demand systems and power distribution networks – an optimal power flow approach," *arXiv preprint arXiv:1905.00200*, 2019.
- [18] N. Tucker, B. Turan, and M. Alizadeh, "Online charge scheduling for electric vehicles in autonomous mobility on demand fleets," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2019.
- [19] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the Association for Computing Machinery*, vol. 19, no. 2, pp. 248–264, 1972.
- [20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.
- [21] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.
- [22] TomTom. (2019) Los angeles in the traffic index. TomTom. Available at https://www.tomtom.com/en_gb/traffic-index/los-angeles-traffic.
- [23] CAISO. (2019) Today's outlook: Current and forecasted demand. California ISO. Available at <http://www.caiso.com/TodaysOutlook/Pages/default.aspx>.
- [24] G. Zardini, N. Lanzetti, M. Salazar, A. Censi, E. Frazzoli, and M. Pavone, "Towards a co-design framework for future mobility systems," in *Annual Meeting of the Transportation Research Board*, 2020, submitted.