Dynamic Pricing and Management for Electric Autonomous Mobility on Demand Systems Using Reinforcement Learning

Berkay Turan  Ramtin Pedarsani  Mahnoosh Alizadeh

Abstract—The proliferation of ride sharing systems is a major drive in the advancement of autonomous and electric vehicle technologies. This paper considers the joint routing, battery charging, and pricing problem faced by a profit-maximizing transportation service provider that operates a fleet of autonomous electric vehicles. We define the dynamic system model that captures the time dependent and stochastic features of an electric autonomous-mobility-on-demand system. To accommodate for the time-varying nature of trip demands, renewable energy availability, and electricity prices and to further optimally manage the autonomous fleet, a dynamic policy is required. In order to develop a dynamic control policy, we first formulate the dynamic progression of the system as a Markov decision process. We argue that it is intractable to exactly solve for the optimal policy using exact dynamic programming methods and therefore apply deep reinforcement learning to develop a near-optimal control policy. Furthermore, we establish the static planning problem by considering time-invariant system parameters. We define the capacity region and determine the optimal static policy to serve as a baseline for comparison with our dynamic policy. While the static policy provides important insights on optimal pricing and fleet management, we show that in a real dynamic setting, it is inefficient to utilize a static policy. The two case studies we conducted in Manhattan and San Francisco demonstrate the efficacy of our dynamic policy in terms of network stability and profits, while keeping the queue lengths up to 200 times less than the static policy.

I. INTRODUCTION

The rapid evolution of enabling technologies for autonomous driving coupled with advancements in eco-friendly electric vehicles (EVs) has facilitated state-of-the-art transportation options for urban mobility. Owing to these developments in automation, it is possible for an autonomous-mobility-on-demand (AMoD) fleet of autonomous EVs to serve the society’s transportation needs, with multiple companies now heavily investing in AMoD technology [1].

The introduction of autonomous vehicles for mobility on demand services provides an opportunity for better fleet management. Specifically, idle vehicles can be rebalanced throughout the network in order to prevent accumulating at certain locations and to serve induced demand at every location. Autonomous vehicles allow rebalancing to be performed centrally by a platform operator who observes the state of all the vehicles and the demand, rather than locally by individual drivers. Furthermore, EVs provide opportunities for cheap and environment-friendly energy resources (e.g., solar energy). However, electricity supplies and prices differ among the network both geographically and temporally. As such, this diversity can be exploited for cheaper energy options when the fleet is operated by a platform operator that is aware of the electricity prices throughout the whole network. Moreover, a dynamic pricing scheme for rides is essential to maximize profits earned by serving the customers. Coupling an optimal fleet management policy with a dynamic pricing scheme allows the revenues to be maximized while reducing the rebalancing cost and the waiting time of the customers by adjusting the induced demand.

We consider a model that captures the opportunities and challenges of an AMoD fleet of EVs, and consists of complex state and action spaces. In particular, the platform operator has to consider the number of customers waiting to be served at each location (queue lengths), the electricity prices, and the states of the EVs (locations, battery energy levels) in order to make decisions. These decisions consist of pricing for rides for every OD-pair and routing/charging decision for every vehicle in the network. Upon taking an action, the state of the network undergoes through a stochastic transition due to the randomness in customer behaviour and exogenously-determined electricity prices.

Due to the continuous and high dimensional state-action spaces, it is infeasible to develop an optimal policy using exact dynamic programming algorithms. As such, we utilize deep reinforcement learning (RL) to develop a near-optimal policy. Specifically, we show that it is possible to learn a policy via Trust Region Policy Optimization (TRPO) [2] that increases the total profits generated by jointly managing the fleet of EVs (by making routing and charging decisions) and pricing for the rides. We demonstrate the performance of our policy by using the total profits generated and the queue lengths as metrics.

Our contributions are as follows:

- We formalize a vehicle and network model that captures the aforementioned characteristics of an AMoD fleet of EVs as well as the stochasticity in demand and electricity prices.
- We employ deep RL methods to learn a joint pricing, routing and charging policy that effectively stabilizes the queues and increases the profits.
- We analyze the static problem, where we consider a time-invariant environment (time-invariant arrivals, electricity prices, etc.), to gain insight towards the actual dynamic problem and to further provide a baseline for comparison.
Fig. 1: The schematic diagram of our framework. Our deep RL agent processes the state of the vehicles, queues and electricity prices and outputs a control policy for pricing as well as autonomous EVs’ routing and charging.

Fig. 2: (a) The optimal static policy manages to stabilize the queues over a very long time period but is unable to clear them whereas (b) RL control policy stabilizes the queues and manages to keep them significantly low (note the scales).

We visualize our framework as a schematic diagram in Figure 1 and preview our results in Figure 2 showing that the RL policy successfully keeps the queue lengths 200 times lower than the static policy.

**Related work:** Comprehensive research perceiving various aspects of AMoD systems is being conducted in the literature. Studies surrounding fleet management focus on optimal EV charging in order to reduce electricity costs as well as optimal vehicle routing in order to serve the customers and to rebalance the empty vehicles throughout the network so as to reduce the operational costs and the customers’ waiting times. Time-invariant control policies adopting queueing theoretical [3], fluidic [4], network flow [5], and Markovian [6] models have been developed by using the steady state of the system. The authors of [7] consider ride-sharing systems with mixed autonomy. However, the proposed control policies in these papers are not adaptive to the time-varying nature of the future demand. As such, there is work on developing time-varying model predictive control (MPC) algorithms [8]–[12]. The authors of [10], [11] propose data-driven algorithms and the authors of [12] propose a stochastic MPC algorithm focusing on vehicle rebalancing. In [8], the authors also consider a fleet of EVs and hence propose an MPC approach that optimizes vehicle routing and scheduling subject to energy constraints. Using a fluid-based optimization framework, the authors of [13] investigate tradeoffs between fleet size, rebalancing cost, and queuing effects in terms of passenger and vehicle flows under time-varying demand. The authors in [14] develop a parametric controller that approximately solves the intractable dynamic program for rebalancing over an infinite-horizon. Aside from these, there are studies that aim to develop dynamic policies for rebalancing as well as ride request assignment via decentralized reinforcement learning approaches [15]–[17]. In these works, the policies are developed and applied locally by each autonomous vehicle, and dynamic pricing and charging strategy are not considered. Dynamic routing of autonomous vehicles using reinforcement learning with the goal of reducing congestion in mixed autonomy traffic networks is proposed in [18].

Regarding charging strategies for large populations of EVs, [19]–[21] provide in-depth reviews and studies of smart charging technologies. An agent-based model to simulate the operations of an AMoD fleet of EVs under various vehicle and infrastructure scenarios has been examined in [22]. The authors of [23] propose an online charge scheduling algorithm for EVs providing AMoD services. By adopting a static network flow model in [24], the benefits of smart charging have been investigated and approximate closed form expressions that highlight the trade-off between operational costs and charging costs have been derived. Furthermore, [25] studies...
interactions between AMoD systems and the power grid. In addition, [25] studies the implications of pricing schemes on an AMoD fleet of EVs. In [27], the authors propose a dynamic joint pricing and routing strategy for non-electric shared mobility on demand services. [29] studies a quadratic programming problem in order to jointly optimize vehicle dispatching, charge scheduling, and charging infrastructure, while the demand is defined exogenously.

**Paper Organization:** The remainder of the paper is organized as follows. In Section [II] we present the system model and define the platform operator’s optimization problem. In Section [III] we first formulate the dynamics of the system as a Markov decision process and then explain the idea of reinforcement learning method as well as the algorithm we adopted. In Section [IV] we discuss the static planning problem associated with the system model and characterize the capacity region as well as the optimal static policy. In Section [V] we present the numerical results of the case studies we have conducted in Manhattan and San Francisco to demonstrate the performance of our dynamic control policy. Finally, we conclude the paper in Section [VI].

## II. System Model and Problem Definition

**Network and Demand Models:** We consider a fleet of AMoD EVs operating within a transportation network characterized by a fully connected graph consisting of $\mathcal{M} = \{1, \ldots, m\}$ nodes that can each serve as a trip origin or destination. We study a discrete-time system with time periods normalized to integral units $t \in \{0, 1, 2, \ldots\}$. In this discrete-time system, we model the arrival of the potential customers with origin-destination (OD) pair $(i, j)$ as a Poisson process with an arrival rate of $\lambda_{ij}$ per period, where $\lambda_{ii} = 0$. Moreover, we assume that these riders are heterogeneous in terms of their willingness to pay. In particular, if the price for receiving a ride from node $i$ to node $j$ in period $t$ is set to $\ell_{ij}(t)$, the induced arrival rate for rides from $i$ to $j$ is given by $\Lambda_{ij}(t) = \lambda_{ij}(1 - F(\ell_{ij}(t)))$, where $F(\cdot)$ is the cumulative distribution function of riders’ willingness to pay with a support of $[0, \ell_{\max}]$. Thus, the number of new ride requests in time period $t$ is $\Lambda_{ij}(t) \sim \text{Pois}(\Lambda_{ij}(t))$ for OD pair $(i, j)$.

**Vehicle Model:** To capture the effect of trip demand and the associated charging, routing, and rebalancing decisions on the fleet size, we assume that each autonomous vehicle in the fleet has a per period operational cost of $\beta$. Furthermore, as the vehicles are electric, they have to sustain charge in order to operate. Without loss of generality, we assume there is a charging station placed at each node $i \in \mathcal{M}$. To charge at node $i$ during time period $t$, the operator pays a price of electricity $p_{\ell}(t)$ per unit of energy. We assume that all EVs in the fleet have a battery capacity denoted as $v_{\max} \in \mathbb{Z}^+$; therefore, each EV has a discrete battery energy level $v \in \mathcal{V}$, where $\mathcal{V} = \{v \in \mathbb{N} | 0 \leq v \leq v_{\max}\}$. In our discrete-time model, we assume each vehicle takes one period to charge one unit of energy and $\tau_{ij}(t)$ periods to travel between OD pair $(i, j)$ if the ride is starting at time period $t$, while consuming $v_{ij}$ units of energy.

**Ride Sharing Model:** The platform operator dynamically routes the fleet of EVs in order to serve the demand at each node. Customers that purchase a ride are not immediately matched with a ride, but enter the queue for OD pair $(i, j)$. After the platform operator executes routing decisions for the fleet, the customers in the queue for OD pair $(i, j)$ are matched with rides and served in a first-come, first-served discipline. A measure of the expected wait time is not available to each arriving customer. However, the operator knows that longer wait times will negatively affect their business and hence seeks to minimize the total wait time experienced by users. Denote the queue length for OD pair $(i, j)$ by $q_{ij}(t)$. If after serving the customers, the queue length $q_{ij}(t) > 0$, the platform operator is penalized by a fixed cost of $w$ per person at the queue to account for the value of time of the customers.

**Platform Operator’s Problem:** We consider a profit-maximizing AMoD operator that manages a fleet of EVs that make trips to provide transportation services to customers. The operator’s goal is to maximize profits by 1) setting prices for rides and hence managing customer demand at each node; 2) optimally operating the AMoD fleet (i.e., charging, routing, and rebalancing) to minimize operational and charging costs. We will study two types of control policies the platform operator utilizes: 1) a dynamic policy, where the pricing, routing and charging decisions are dependent on the system state (such as queue lengths, prices of electricity, and vehicle locations and energy levels); 2) a static policy, where the pricing, routing and charging decisions are time invariant and independent of the state of the system.

## III. The Proposed Dynamic Policy

In this section, we establish a dynamic control policy to optimize the decisions that the platform operator makes given full state information. We first formulate the dynamic evolution of the network state as an MDP. The solution of this MDP is the optimal policy that determines which action to take for each state the system is in, and can nominally be derived using classical exact dynamic programming algorithms (e.g., value iteration). However, considering the complexity and the scale of our dynamic problem, the curse of dimensionality renders the MDP intractable to solve with classical exact dynamic programming algorithms. As such, we resort to approximate dynamic programming methods. Specifically, we define the policy via a deep neural network that takes the current state of the network (such as prices of electricity, queue lengths, and vehicle locations and energy levels) as input and outputs the best action (such as prices for rides and vehicle routing and charging decisions). Subsequently, we apply a reinforcement learning algorithm to train the neural network in order to improve the performance of the policy.

### A. The Dynamic Problem as MDP

We define the MDP by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the state transition operator and $r$ is the reward function. We define these elements as follows:

1. In general, the policy is a stochastic policy and determines the probabilities of taking the actions rather than deterministically producing an action.
1) $S$: The state space consists of prices of electricity at each node, the queue lengths for each origin-destination pair, and the number of vehicles at each node and each energy level. However, since travelling from node $i$ to node $j$ takes $\tau_{ij}(t)$ periods of time, we need to define intermediate nodes. For brevity of exposition, let us assume that $\tau_{ij}(t)$ is a constant $\tau_{ij}$ during the time period for which the dynamic policy is developed. As such, we define $\tau_{ij} - 1$ number of intermediate nodes between each origin and destination pair, for each battery energy level $v$. Hence, the state space consists of $s_d = m^2 + (v_{\text{max}} + 1)(\sum_{i=1}^{m} \sum_{j=1}^{m} \tau_{ij}) - m^2 + 2m)$ dimensional vectors in $\mathbb{R}^{s_d}_{\geq 0}$ (We include all the non-negative valued vectors, however, only $m^2 - m$ entries can grow to infinity because they are queue lengths, and the rest are always upper bounded by fleet size or maximum price of electricity).

As such, we define the elements of the state vector at time $t$ as $s(t) = [p(t) \ q(t) \ s_{\text{veh}}(t)]$, where $p(t) = [p_i(t)]_{i \in M}$ is the electricity prices state vector, $q(t) = [q_{ij}(t)]_{i,j \in M, i \neq j}$ is the queue lengths state vector, and $s_{\text{veh}}(t) = [s_{\text{veh}}^i(t)]_{i \in M, v_i \in V}$ is the vehicle state vector, where $s_{\text{veh}}^i(t)$ is the number of vehicles at vehicle state $(i, j, k, v)$. The vehicle state $(i, j, k, v)$ specifies the location of a vehicle that is travelling between OD pair $(i, j)$ as the $k$'th intermediate node between nodes $i$ and $j$, and specifies the battery energy level of a vehicle as $v$ (The States of the vehicles at the nodes $i \in M$ with energy level $v$ is denoted by $(i, i, 0, v)$).

2) $A$: The action space consists of prices for rides at each origin-destination pair and routing/charging decisions for vehicles at nodes $i \in M$ at each energy level $v$. The price actions are continuous in range $[0, \ell_{\text{max}}]$. Each vehicle at state $(i, i, 0, v)$ (where $i \in M$, $\forall v \in V$) can either stay idle or travel to one of the remaining $m - 1$ nodes. To allow for different transitions for vehicles at the same state (some might charge, some might travel to another node), we define the action taken at time $t$ for vehicles at state $(i, i, 0, v)$ as an $m + 1$ dimensional probability vector with entries in $[0, 1]$ that sum up to 1: $\alpha_i^t = [\alpha_{ij}^t(1) \ldots \alpha_{im}^t(1) \alpha_{i0}^t(1)]$, where $\alpha_{ij}^t(1) = 0$ and $\alpha_{ij}^t(0) = 0$ if $v < v_{ij}$. The action space is then all the vectors $\alpha$ of dimension $s_d = m^2 - m + (v_{\text{max}} + 1)(m^2 + m)$, whose first $m^2 - m$ entries are the prices and the rest are the probability vectors satisfying the aforementioned properties. As such, we define the elements of the action vector at time $t$ as $\alpha(t) = [\ell(t) \ \alpha(t)]$, where $\ell(t) = [\ell_{ij}]_{i,j \in M, i \neq j}$ is the vector of prices and $\alpha(t) = [\alpha_i^t(1)]_{i \in M, 0 \neq j}$ is the vector of routing/charging actions.

3) $T$: The transition operator is defined as $T_{ijk} = Pr(s(t + 1) = j | s(t) = i, \alpha(t) = k)$. We can define the transition probabilities for electricity prices $p(t+1)$, queue lengths $q(t+1)$, and vehicle states $s_{\text{veh}}(t+1)$ as follows:

**Electricity Price Transitions:** Since we assume that the dynamics of prices of electricity are exogenous to our AMoD system, $Pr(p(t+1) = p_{ij} | p(t) = p_{ij}, \alpha(t) = \alpha(t)) = Pr(p(t+1) = p_{ij} | p(t) = p_{ij}, \alpha(t) = \alpha(t))$, i.e., the dynamics of the price are independent of the action taken. Depending on the setting, new prices might either be deterministic or distributed according to some probability density function at time $t$: $p(t) \sim \mathcal{P}(t)$, which is determined by the electricity provider.

**Vehicle Transitions:** For each vehicle at node $i$ and energy level $v$, the transition probability is defined by the action probability vector $\alpha_i^t(v)$. Each vehicle transitions into state $(i, j, 1, v - v_{ij})$ with probability $\alpha_{ij}^t(v)$, stays idle in state $(i, i, 0, v)$ with probability $\alpha_{ii}^t(v)$ or charges and transitions into state $(i, i, v, 0 + 1)$ with probability $\alpha_{i0}^t(v)$.

The vehicles at intermediate states $(i, j, k, v)$ transition into state $(i, j, k + 1, v)$ if $k < \tau_{ij} - 1$ or $(j, j, 0, v)$ if $k = \tau_{ij} - 1$ with probability 1. The total transition probability to the vehicle states $s_{\text{veh}}(t+1)$ given $s_{\text{veh}}(t)$ and $\alpha(t)$ is the sum of all the probabilities of the feasible transitions from $s_{\text{veh}}(t)$ to $s_{\text{veh}}(t+1)$ under $\alpha(t)$, where the probability of a feasible transition is the multiplication of individual vehicle transition probabilities (since the vehicle transition probabilities are independent). Note that instead of gradually dissipating the energy of the vehicles on their route, we immediately discharge the required energy for the trip from their batteries and keep them constant during the trip. This ensures that the vehicles have enough battery to complete the ride and does not violate the model, because the vehicles arrive to their destinations with true value of energy and a new action will only be taken when they reach the destination.

**Queue Transitions:** The queue lengths transition according to the prices and the vehicle routing decisions. For prices $\ell_{ij}(t)$ and induced arrival rate $A_{ij}(t)$, the probability that $A_{ij}(t)$ new customers arrive in the queue $(i, j)$ is:

$$Pr(A_{ij}(t)) = \frac{e^{-A_{ij}(t)} A_{ij}(t)^{A_{ij}(t)}}{(A_{ij}(t))!}$$

Let us denote the total number of vehicles routed from node $i$ to $j$ at time $t$ as $x_{ij}(t)$, which is given by:

$$x_{ij}(t) = \sum_{v=v_{ij}}^{v_{\text{max}}} x_{ij}^v(t) = \sum_{v=v_{ij}}^{v_{\text{max}}} s_{ij}^{v-v_{ij}}(t+1).$$

Given $s_{\text{veh}}(t+1)$ and $x_{ij}(t)$, the probability that the queue length $q_{ij}(t+1) = q$ is:

$$Pr(q_{ij}(t+1) = q | s(t), \alpha(t), s_{\text{veh}}(t+1)) = Pr(A_{ij}(t) = q - q_{ij}(t) + x_{ij}(t)),$$

if $q > 0$, and $Pr(A_{ij}(t) = -q_{ij}(t) + x_{ij}(t))$ if $q = 0$. Since the arrivals are independent, the total probability that the queue vector $q(t+1) = q$ is:

$$Pr(q(t+1) = q | s(t), \alpha(t), s_{\text{veh}}(t+1)) = \prod_{i=1}^{m} \prod_{j \neq i}^{m} Pr(q_{ij}(t+1) | s(t), \alpha(t), s_{\text{veh}}(t+1)).$$

2To account for different traffic conditions during different time periods of the day, we can define different sets of intermediate nodes. According to the traffic conditions, the vehicles take the longer route (higher traffic, more intermediate nodes) or the shorter route (less traffic, less intermediate nodes). Furthermore, to account for stochasticity on the routes (e.g., traffic lights), the number of intermediate nodes a vehicle traverses in one time period can be defined as a random variable.
Fig. 3: The schematic diagram representing the state transition of our MDP. Upon taking an action, a vehicle at state \((i, i, 0, v)\) charges for a price of \(p_i(t)\) and transitions into state \((i, i, 0, v + 1)\) with probability \(\alpha_{ii}^v(t)\), stays idle at state \((i, i, 0, v)\) with probability \(\alpha_{ii}^{0v}(t)\), or starts traveling to another node \(j\) and transitions into state \((i, j, 1, v - v_{ij})\) with probability \(\alpha_{ij}^{1v}(t)\). Furthermore, \(A_{ij}(t)\) new customers arrive to the queue \((i, j)\) depending on the price \(\ell_{ij}(t)\). After the routing and charging decisions are executed for all the EVs in the fleet, the queues are modified.

Hence, the transition probability is defined as:

\[
\begin{align*}
Pr(s(t+1)|s(t), a(t)) &= Pr(p(t+1)|p(t)) \\
&\times Pr(s_{veh}(t+1)|s(t), \alpha(t)) \\
&\times Pr(q(t+1)|s(t), \alpha(t), s_{veh}(t+1))
\end{align*}
\]

We illustrate how the vehicles and queues transition into new states consequent to an action in Figure 3.

4) \(r\): The reward function \(r(t)\) is a function of state-action pairs at time \(t\):

\[
r(t) = r(a(t), s(t)).
\]

Let \(x_{ic}^v(t)\) denote the number of vehicles charging at node \(i\) starting with energy level \(v\) at time period \(t\). The reward function \(r(t)\) is defined as:

\[
r(t) = \sum_{i=1}^{m} \sum_{j=1}^{m} \ell_{ij}(t) A_{ij}(t) - w \sum_{i=1}^{m} \sum_{j=1}^{m} q_{ij}(t) \\
- \sum_{i=1}^{m} \sum_{j=1}^{m} (\beta + p_i) x_{ic}^v(t) \\
- \beta \sum_{i=1}^{m} \sum_{j=1}^{m} x_{ij}(t) \\
- \beta \sum_{i=1}^{m} \sum_{j=1}^{m} \tau_{ij} x_{ij}(t)
\]

The first term corresponds to the revenue generated by the passengers that request a ride for a price \(\ell_{ij}(t)\), the second term is the queue cost of the passengers that have not yet been served, the third term is the charging and operational costs of the charging vehicles and the last two terms are the operational costs of the vehicles making trips. Note that revenue generated is immediately added to the reward function when the passengers enter the network instead of after the passengers are served. Since the reinforcement learning approach is based on maximizing the cumulative reward gained, all the passengers eventually have to be served in order to prevent queues from blowing up and hence it does not violate the model to add the revenues immediately.

Using the definitions of the tuple \((S, A, T, r)\), we model the dynamic problem as an MDP. Observe that aside from having a large dimensional state space (for instance, \(m = 10\), \(v_{max} = 5\), \(\tau_{ij} = 3\) \(\forall i, j\): \(s_q = 1240\)) and action space, the cardinality of these spaces are not finite (queues can grow unbounded, prices are continuous). As such, we cannot solve the MDP using exact dynamic programming methods. As a solution, we characterize the dynamic policy via a deep neural network and execute reinforcement learning in order to develop a dynamic policy.

B. Reinforcement Learning Method

In this subsection, we go through the preliminaries of reinforcement learning and briefly explain the idea of the algorithm we adopted.

1) Preliminaries: The dynamic policy associated with the MDP is defined as a function parameterized by \(\theta\): \(\pi^\theta(a|s) = \pi : S \times A \rightarrow [0, 1]\), i.e., a probability distribution in the state-action space. Given a state \(s\), the policy returns the probability for taking the action \(a\) (for all actions), and samples an action according to the probability distribution. The goal is to derive the optimal policy \(\pi^*\), which maximizes the discounted cumulative expected rewards \(J^\pi\):

\[
J^\pi = \max_{\pi} \mathbb{E}_\pi \sum_{t=0}^{\infty} \gamma^t r(t),
\]

where \(\gamma \in (0, 1]\) is the discount factor. The value of taking an action \(a\) in state \(s\), and following the policy \(\pi\) afterwards is characterized by the value function \(Q^\pi(s, a)\):

\[
Q^\pi(s, a) = \mathbb{E}_\pi \sum_{t=0}^{\infty} \gamma^t r(t) | s(0) = s, a(0) = a.
\]

The value of being in state \(s\) is formalized by the value function \(V^\pi(s)\):

\[
V^\pi(s) = \mathbb{E}_{\pi(a(0))} \sum_{t=0}^{\infty} \gamma^t r(t) | s(0) = s.
\]
and the advantage of taking the action \( a \) in state \( s \) and following the policy \( \pi \) thereafter is defined as the advantage function \( A_\pi(s,a) \):

\[
A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s).
\]

The methods used by reinforcement learning algorithms can be divided into three main groups: 1) critic-only methods, 2) actor-only methods, and 3) actor-critic methods, where the word critic refers to the value function and the word actor refers to the policy [29]. Critic-only (or value-function based) methods (such as Q-learning [30] and SARSA [31]) improve a deterministic policy using the value function by iterating:

\[
\begin{align*}
\mathbf{a}^* &= \arg \max_a Q_{\pi}(s,a), \\
\pi(\mathbf{a}^*|s) &\leftarrow 1.
\end{align*}
\]

Actor-only methods (or policy gradient methods), such as Williams’ REINFORCE algorithm [32], improve the policy by updating the parameter \( \theta \) by gradient ascent, without using any form of a stored value function:

\[
\theta(t + 1) = \theta(t) + \alpha \nabla_{\theta} \mathbb{E}_{\pi(t)} \left[ \sum_{t} \gamma^t r(t) \right].
\]

The advantage of policy gradient methods is their ability to generate actions from a continuous action space by utilizing a parameterized policy.

Finally, actor-critic methods [33], [34] make use of both the value functions and policy gradients:

\[
\theta(t + 1) = \theta(t) + \alpha \nabla_{\theta} \mathbb{E}_{\pi(t)} \left[ Q_{\pi(t)}(s,a) \right].
\]

Actor-critic methods are able to produce actions in a continuous action space, while reducing the high variance of the policy gradients by adding a critic (value function).

All of these methods aim to update the parameters \( \theta \) (or directly update the policy \( \pi \) for critic-only methods) to improve the policy. In deep reinforcement learning, the policy \( \pi \) is defined by a deep neural network, whose weights constitute the parameter \( \theta \). To develop a dynamic policy for our MDP, we adopt a practical policy gradient method called Trust Region Policy Optimization (TRPO).

2) Trust Region Policy Optimization: TRPO is a practical policy gradient method developed in [2], and is effective for optimizing large nonlinear policies such as neural networks. It supports continuous state-action spaces and guarantees monotonic improvement.

Let \( \pi \) and \( \tilde{\pi} \) be two different policies. Then, the following equality indicates the expected return of policy \( \tilde{\pi} \) in terms of the advantage over \( \pi \):

\[
J_{\tilde{\pi}} = J_\pi + \mathbb{E}_{\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_\pi(s,a) \right]. \tag{3}
\]

Let \( \sigma_\pi(s) \) be the discounted visitation frequency of state \( s \) under policy \( \pi \):

\[
\sigma_\pi(s) = Pr(s(0) = s) + \gamma Pr(s(1) = s) + \ldots,
\]

where \( s(0) \) is distributed according to some initial distribution \( \sigma_0 \). Using \( \sigma_\pi(s) \) and writing the expectation explicitly, Equation (3) becomes:

\[
J_{\tilde{\pi}} = J_\pi + \sum_s \sigma_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a) \tag{4}
\]

This implies that any policy \( \tilde{\pi} \) such that \( \sum_s \tilde{\pi}(a|s) A_\pi(s,a) \geq 0 \) at every state \( s \) is at least as good as policy \( \pi \). However, because of dependency of \( \sigma_\pi \) on \( \tilde{\pi} \), it is difficult to optimize Equation (4). Thus a local approximator to \( J_{\tilde{\pi}} \) using the visitation frequencies \( \sigma_\pi(s) \) is introduced:

\[
L_\pi(\tilde{\pi}) = J_\pi + \sum_s \sigma_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a)
\]

Using this approximator \( L_\pi(\tilde{\pi}) \), Algorithm 1 can be applied to utilize policy iteration with \( \epsilon = \max_{s,a} |A_\pi(s,a)| \) and \( D_{KL}^{\max}(\pi_t,\pi) = \max_{s,a} D_{KL}(\pi(s)|\tilde{\pi}(s)|) \) being the KL divergence between two policies maximized over the states. The key idea of Algorithm 1 is to utilize policy iteration

**Algorithm 1**: Policy iteration algorithm guaranteeing non-decreasing expected return

\[
\text{Initialize } \pi.
\]

for \( t = 0, 1, 2, \ldots \) until convergence do

- Compute all advantage values \( A_{\pi_t}(s,a) \).
- Solve the constrained optimization problem:

\[
\pi_{t+1} = \arg \max_{\pi} \left[ L_{\pi_t}(\pi) - C D_{KL}^{\max}(\pi_t,\pi) \right],
\]

where \( C = 4\epsilon \gamma/(1 - \gamma)^2 \),

and

\[
L_{\pi_t}(\pi) = J_{\pi_t} + \sum_s \sigma_{\pi_t}(s) \sum_a \pi(a|s) A_{\pi_t}(s,a).
\]

end

without changing the policy too much by imposing a penalty on the KL divergence. This is the same idea that lies at the heart of TRPO. Instead of penalizing the KL divergence, TRPO imposes a constraint on KL divergence and solves the constrained maximization problem using conjugate gradient. In that sense, it is similar to natural policy gradient methods. We refer the reader to [2] for a comprehensive study.

IV. ANALYSIS OF THE STATIC PROBLEM

In this section, we establish and discuss the static planning problem to provide a measure for comparison and demonstrate the efficacy of the dynamic policy. To do so, we consider the fluid scaling of the dynamic network and characterize the static problem via a network flow formulation. Under this setting, we use the expected values of the variables (travel durations, arrivals, and prices of electricity) and ignore their time dependent dynamics, while allowing the vehicle routing decisions to be flows (real numbers) rather than integers. The static problem is convenient for determining the so-called capacity region of the dynamic problem as well as determining the optimal static pricing, routing, and charging policy of the platform operator.
A. The Capacity Region

We formulate the static optimization problem via a network flow model that characterizes the capacity region of the network for a given set of prices \( \ell_{ij}(t) = \ell_{ij} \) \( \forall t \) (Hence, \( \Lambda_{ij}(t) = \Lambda_{ij} \) \( \forall t \)). The capacity region is defined as the set of all arrival rates \( \Lambda_{ij}, i,j \in \mathcal{M} \), where there exists a charging and routing policy under which the queueing network of the system is stable \( ^{[55]} \). Let \( x_{ij}^v \) be the number of vehicles available at node \( i \), \( \alpha_{ij} \) be the fraction of vehicles at node \( i \) with energy level \( v \) being routed to node \( j \), and \( \alpha_{ic}^v \) be the fraction of vehicles charging at node \( i \) starting with energy level \( v \). We say the static vehicle allocation for node \( i \) and energy level \( v \) is feasible if:

\[
\alpha_{ic}^v + \sum_{j \neq i} \alpha_{ij}^v \leq 1.
\]

The optimization problem that characterizes the capacity region of the network ensures that the total number of vehicles routed from \( i \) to \( j \) is at least as large as the nominal arrival rate to the queue \((i,j)\). Namely, the problem can be formulated as follows:

\[
\begin{aligned}
\min_{x_{ij}^v, \alpha_{ij}^v, \alpha_{ic}^v} & \quad \rho \\
\text{subject to} & \quad \Lambda_{ij} \leq \sum_{v=1}^{v_{\max}} x_{ij}^v \alpha_{ij}^v \quad \forall i,j \in \mathcal{M}, \\
& \quad \rho \geq \alpha_{ic}^v + \sum_{j=1}^{m} \alpha_{ij}^v \quad \forall i \in \mathcal{M}, \forall v \in \mathcal{V}, \\
& \quad x_{ij}^v = x_{ij}^{v-1} - x_{ji}^v - \sum_{j=1}^{n} x_{ij}^{v+\nu_{ij}} \alpha_{ji}^v \quad \forall i \in \mathcal{M}, \forall v \in \mathcal{V}, \\
& \quad \sum_{i=1}^{m} \sum_{j=1}^{m} x_{ij}^v \alpha_{ij}^v \leq N, \\
& \quad \alpha_{ij}^v = 0 \quad \forall v < v_{ij}, \forall i,j \in \mathcal{M}, \\
& \quad x_{ij}^v \geq 0, \alpha_{ij}^v \geq 0, \alpha_{ic}^v \geq 0, \forall i,j \in \mathcal{M}, \forall v \in \mathcal{V}, \\
& \quad x_{ij}^v = \alpha_{ic}^v = \alpha_{ij}^v = 0 \quad \forall v \notin \mathcal{V}, \forall i,j \in \mathcal{M}.
\end{aligned}
\]

The constraint \( ^{(5b)} \) requires the platform to operate at least as many vehicles to serve all the induced demand between any two nodes \( i \) and \( j \) (The rest are the vehicles travelling without passengers, i.e., rebalancing vehicles). We will refer to this as the demand satisfaction constraint. The constraint \( ^{(5d)} \) is the flow balance constraint for each node and each battery energy level, which restricts the number of available vehicles at node \( i \) and energy level \( v \) to be the sum of arrivals from all nodes (including idle vehicles) and vehicles that are charging with energy level \( v - 1 \). The constraint \( ^{(5e)} \) is the fleet size constraint, restricting the total number of operated vehicles in the network to be upper bounded by \( N \). The constraint \( ^{(5f)} \) ensures that the vehicles with full battery do not charge further, and the constraint \( ^{(5g)} \) ensures the vehicles sustain enough charge to travel between OD pair \((i,j)\). Finally, the constraint \( ^{(5h)} \) upper bounds the allocation of vehicles for each node \( i \) and energy level \( v \).

**Proposition 1.** Let the optimal value of \( ^{(5)} \) be \( \rho^* \). Then, \( \rho^* \leq 1 \) is a necessary and sufficient condition of rate stability of the system under some routing and charging policy.

The proof of Proposition \( ^{[1]} \) is provided in Appendix \( ^{[A]} \). By Proposition \( ^{[1]} \) the capacity region \( C_\Lambda \) of the network is the set of all \( \Lambda_{ij} \in \mathbb{R}^+ \) for which the corresponding optimal solution to the optimization problem \( ^{(5)} \) satisfies \( \rho^* \leq 1 \). As long as \( \rho^* \leq 1 \), there exists a routing and charging policy such that the queues will be bounded away from infinity.

B. Static Profit Maximization Problem

The platform operator’s goal is to maximize its profits by setting prices and making routing and charging decisions such that the system remains stable. Setting prices for rides allows the platform operator to shift the induced demand into the capacity region (higher prices decrease the arrival rate and thus maintain stability of the queues). In its most general form, the problem can be formulated as follows:

\[
\begin{aligned}
\max_{\ell_{ij}, x_{ij}^v, \alpha_{ij}^v, \alpha_{ic}^v} & \quad U(\Lambda_{ij}(\ell_{ij}), x_{ij}^v, \alpha_{ij}^v, \alpha_{ic}^v) \\
\text{subject to} & \quad [\Lambda_{ij}(\ell_{ij})]_{i,j} \in \mathcal{M} \in C_\Lambda,
\end{aligned}
\]

where \( U(\cdot) \) is the utility function that depends on the prices, demand for rides and the vehicle decisions.

Next, we explicitly state the platform operator’s profit maximization problem. Instead of imposing a fleet size constraint to the problem, we want to jointly optimize pricing, routing, and charging as well as the fleet size. To account for the effect of fleet size, we assign a per vehicle operational costs of \( \beta \). Let \( x_{ij}^v = x_{ij}^v \alpha_{ij}^v \) and \( x_{ij}^v = x_{ij}^v \alpha_{ij}^v \). Using these new variables and noting that \( \alpha_{ij}^v + \sum_{v=1}^{v_{\max}} \alpha_{ij}^v = 1 \) when \( \rho^* \leq 1 \), the platform operator’s problem can be stated as:

\[
\begin{aligned}
\max_{x_{ij}^v, x_{ij}^v, \ell_{ij}} & \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_{ij} \ell_{ij}(1 - F(\ell_{ij})) \\
& \quad - \sum_{i=1}^{m} \sum_{v=1}^{v_{\max}} (\beta + p_v) x_{ij}^v \\
& \quad - \beta \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{v=v_{ij}}^{v_{\max}} x_{ij}^v \tau_{ij}
\end{aligned}
\]

subject to \( \lambda_{ij}(1 - F(\ell_{ij})) \leq \sum_{v=v_{ij}}^{v_{\max}} x_{ij}^v \quad \forall i,j \in \mathcal{M}, \quad \sum_{i=1}^{m} x_{ij}^v = x_{ij}^v, \quad \forall v \in \mathcal{V}, \quad \forall i,j \in \mathcal{M}.

The stability condition that we are interested in is rate stability of all queues. A queue for OD pair \((i,j)\) is rate stable if \( \lim_{t \to \infty} q_{ij}(t)/t = 0. \)
The first term in the objective function in (7) accounts for the aggregate revenue the platform generates by providing rides for \( \lambda_{ij}(1 - F(\tau_{ij})) \) number of riders with a price of \( \ell_{ij} \). The second term is the operational and charging costs incurred by the charging vehicles (assuming that \( p_i(t) = p_i \forall i \) under the static setting), and the last term is the operational costs of the trip-making vehicles (including rebalancing trips). The constraints are similar to those of (5), with \( x_i^v = x_{i}^{v_{ic}} + \sum_{j=1}^{m} x_{ij}^v \) (excluding the fleet size constraint).

The optimization problem in (7) is non-convex for a general \( F(\cdot) \). Nonetheless, when the platform’s profits are affine in the induced demand \( \lambda_{ij}(1 - F(\cdot)) \), it can be rewritten as a convex optimization problem. Hence, we assume that the rider’s willingness to pay is uniformly distributed in \([0, \lambda]\) (the induced demand).

Marginal Pricing: The prices for rides are a crucial component of the profits generated. The next proposition highlights how the optimal prices \( \ell_{ij}^* \) for rides are related to the network parameters, prices of electricity, and the operational costs.

**Proposition 2.** Let \( v_{ij}^* \) be optimal the dual variable corresponding to the demand satisfaction constraint for OD pair \((i, j)\). The optimal prices \( \ell_{ij}^* \) are:

\[
\ell_{ij}^* = \frac{\lambda_{ij} + v_{ij}^*}{2}. \tag{8}
\]

These prices can be upper bounded by:

\[
\ell_{ij}^* \leq \frac{\ell_{ij}^*}{2} + \frac{\ell_{ij}^*}{2} \beta(\tau_{ij} + \tau_{ji} + \tau_{ij} + \tau_{ji}) + v_{ij}^* + v_{ji}^* \tag{9}
\]

Moreover, with these optimal prices \( \ell_{ij}^* \), the profits generated per period is:

\[
P = \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_{ij} \frac{\ell_{ij}^* - \ell_{ij}^*}{2}. \tag{10}
\]

The proof of Proposition 2 is provided in Appendix B.

Observe that the prices for rides increase. Thus expensive rides generate less profits compared to the cheaper rides and it is more beneficial if the optimal dual variables \( v_{ij}^* \) are small and prices are close to \( \ell_{ij}^*/2 \). We can interpret the dual variables \( v_{ij}^* \) as the cost of providing a single ride between \( i \) and \( j \) to the platform. In the worst case scenario, every single requested ride from node \( i \) requires rebalancing and charging both at the origin and the destination. Hence the upper bound on \( \ell_{ij}^* \) includes the operational costs of passenger-carrying, rebalancing and charging vehicles (both at the origin and the destination); and the energy costs of both passenger-carrying and rebalancing trips multiplied by the price of electricity at the trip destinations. Similar to the taxes applied on products, whose burden is shared among the supplier and the customer; the costs associated with rides are shared among the platform operator and the riders (which is why the price paid by the riders include half of the cost of the ride).

Even though the static planning problem provides important insights on capacity region, fleet size, and pricing, a static policy does not perform well in a real dynamic setting because it does not acknowledge the time-varying dynamics of the system. We demonstrate the performance of both dynamic and static policies in the next section.

V. Numerical Study

In this section, we discuss the numerical experiments and results for the performance of reinforcement learning approach to the dynamic problem and compare with the performance of several static policies, including the optimal static policy outlined in Section IV. We solved for the optimal static policy using CVX, a package for specifying and solving convex programs \(^{56}\). To implement the dynamic setting as an MDP compatible with reinforcement learning algorithms, we used Gym toolkit \(^{37}\) developed by OpenAI to create an environment. For the implementation of the TRPO algorithm, we used Stable Baselines toolkit \(^{38}\).

We chose an operational cost of \( \beta = 0.1 \) (by normalizing the average price of an electric car over 5 years \(^{39}\)) and maximum willingness to pay \( \ell_{max} = $30 \). For prices of electricity \( p_i(t) \), we generated random prices for different locations and different times using the statistics of locational marginal prices in \(^{40}\). We chose a maximum battery capacity of 20kWh. We discretized the battery energy into 5 units, where one unit of battery energy is 4kWh. The time it takes to deliver one unit of charge is taken as one time epoch, which is equal to 5 minutes in our setup. The waiting time cost for one period is \( w = $24 \) (average hourly wage is around $24 in the United States \(^{41}\)).

Observe that the dimension of the state space grows significantly with \( v_{max} \) and \( \tau_{ij} \) (for instance, \( m = 10, v_{max} = 5, \tau_{ij} = 3 \forall i,j: s_d = 1240 \)). Therefore, for computational purposes, we conducted two case studies: 1) Non-electric AMoD case study with a larger network in Manhattan, 2) Electric AMoD case study with a smaller network in San Francisco. Both experiments were performed on a laptop computer with Intel® Core™ i7-8750H CPU (6×2.0 GHz) and 16 GB DDR4 2666MHz RAM.

A. Case Study in Manhattan

In a non-electric AMoD network, the energy dimension \( v \) vanishes. Because there is no charging action \(^{4}\), we can perform coarser discretizations of time. Specifically, we can allow each discrete time epoch to cover \( 5 \times \min_{i,j} \tau_{ij} \) minutes, and normalize the travel times \( \tau_{ij} \) and \( w \) accordingly (For EV’s, because charging takes a non-negligible but shorter time than travelling, in general we have \( \tau_{ij} > 1 \), and larger number

\(^{4}\)The vehicles still refuel, however this takes negligible time compared to the trip durations.
of states). The static profit maximization problem in (7) for AMoD with non-electric vehicles can be rewritten as:

\[
\max \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_{ij} x_{ij} (1 - F(l_{ij})) - \beta_g \sum_{i=1}^{m} \sum_{j=1}^{m} x_{ij} \tau_{ij}
\]

subject to \( \lambda_{ij} (1 - F(l_{ij})) \leq x_{ij} \quad \forall i, j \in \mathcal{M} \),

\[
\sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{m} x_{ji} \quad \forall i \in \mathcal{M},
\]

\( x_{ij} \geq 0 \quad \forall i, j \in \mathcal{M} \).

The operational costs \( \beta_g = $2.5 \) (per 10 minutes, \( \text{[42]} \)) are different than those of electric vehicles. Because there is no “charging” (or refueling action, since it takes negligible time), \( \beta_g \) also includes fuel cost. The optimal static policy is used to compare and highlight the performance of the dynamic policy.

We divided Manhattan into 10 regions as in Figure 4 and using the yellow taxi data from the New York City Taxi and Limousine Commission dataset \( \text{[43]} \) for May 04, 2019, Saturday between 18.00-20.00, we extracted the average arrival rates for rides and average trip durations \( \tau_{ij} \) between the regions (we exclude the rides occurring in the same region). To create the potential arrival rate \( \lambda_{ij} \), we multiplied the average arrival rates by 1.5. We trained our model by creating new induced random arrivals with the same potential arrival rate \( \lambda_{ij} \). The rest of the parameters were left as default as specified by the Stable Baselines toolkit \( \text{[38]} \). We trained the model for 5 million iterations. The first 1 million iterations of the training phase is displayed in Figure 5a. Observe that, during the first phase of the iterations, the rewards go down rapidly (because the queues blow up). Hence, the policy moves towards higher prices to decrease the arrival rates. However, because the queues cannot be cleared with a single iteration of higher prices, the algorithm observes negative rewards are still there with higher prices, and hence decreases the prices again. This causes the queues to blow up if we leave the algorithm run as is. To overcome this issue, for the first five hundred thousand iterations only, the reward output of the environment was set to the difference between the current and the previous reward. This allows the algorithm to learn that decreasing the queue lengths is favorable. Furthermore, to reduce the variance and stabilize the algorithm, we subtracted a baseline value from the rewards \( \text{[4]} \) after stabilizing the queues.

Next, we compare different policies’ performance using the rewards and total queue length as metrics. The results are demonstrated in Figures 5. In Figure 6a, we compare the rewards generated and the total queue length by applying the static and the dynamic policies as defined in Sections \( \text{[IV] and [III]} \). We can observe that while the optimal static policy provides rate stability in a dynamic setting (since the queues do not blow up), it fails to generate profits as it is not able to clear the queues. On the other hand, the dynamic policy is able to keep the total length of the queues 50 times shorter than the static policy while generating higher profits.

The optimal static policy fails to generate profits and is not necessarily the best static policy to apply in a dynamic setting. As such, in Figure 6b, we demonstrate the performance of a sub-optimal static policy, where the prices are slightly higher to reduce the arrival rates and hence reduce the queue lengths. Observe that the profits generated are higher than the profits generated using optimal static policy for the static planning problem while the total queue length is less. This result indicates that under the stochasticity of the dynamic setting, a sub-optimal static policy can perform better than the optimal static policy. Nevertheless, this policy does still do worse in terms of rewards and total queue length compared to the dynamic policy.

\( ^5 \)The solution of the static problem yields vehicle flows. In order to make the policy compatible with our environment and to generate integer actions that can be applied in a dynamic setting, we randomized the actions by dividing each flow for OD pair \((i, j)\) (and energy level \(v\)) by the total number of vehicles in \(i\) (and energy level \(v\)) and used that fraction as the probability of sending a vehicle from \(i\) to \(j\) (with energy level \(v\)).

\( ^6 \)To get the baseline value, we tested the policy at every one million iterations and subtracted the average value of reward from the reward output during training.
Fig. 6: Comparison of different policies for Manhattan case study. The legends for all figures are the same as the top left figure, where red lines correspond to the dynamic and blue lines correspond to the static policies (We excluded the running averages for (d), because the static policy diverges). In all scenarios, we use the rewards generated and the total queue length as metrics. In (a), we demonstrate the results from applying the dynamic and the optimal static policy. In (b), we compare the dynamic policy with a sub-optimal static policy, where the prices are higher than the optimal static policy. In (c), we utilize a surge pricing policy along with the optimal static policy and compare with the dynamic policy. In (d), we employ the dynamic and static policies developed for May 4, 2019, Saturday for the arrivals on May 11, 2019, Saturday.

Fig. 7: San Francisco divided into $m = 7$ regions. We obtained the map from the San Francisco County Transportation Authority [44]. The map shows number of Transportation Network Company (TNC) pickups and dropoffs. Darker colors mean more trips to/from an area and we divided according to the number of trips rather than the geographical areas.

Next, we showcase the even some heuristic modifications that resemble what is done in practice can do better than the optimal static policy. We utilize the optimal static policy, but additionally utilize a surge-pricing policy. The surge-pricing policy aims to decrease the arrival rates for longer queues so that the queues will stay shorter and the rewards will increase. At each time period, the policy is to increase the prices of the queues longer than 2 people by $7.5 such that the arrival rates for those queues are decreased. The results are displayed in Figure 6c. New arrivals bring higher revenue per person and the total queue length is decreased, which stabilizes the network while generating more profits. The surge pricing policy results in stable short queues and higher rewards compared to the other static policies, yet our dynamic policy beats it.

Finally, we test how the static and the dynamic policies are robust to variations in input statistics. We compare the rewards generated and the total queue length applying the static and the dynamic policies for the arrival rates of May 11, 2019, Saturday between 18.00-20.00. The results are displayed in Figure 6d. Even though the arrival rates between May 11 and May 4 do not differ much, the static policy is not resilient and fails to stabilize when there is a slight change in the network. The dynamic policy, on the other hand, is still able to stabilize the network and generate profits. The neural-network based policy is able to determine the correct pricing and routing decisions by considering the current state of the network, even under different arrival rates.

B. Case Study in San Francisco

We conducted the case study in San Francisco by utilizing an EV fleet of 420 vehicles (according to the optimal fleet size for the static planning problem). We divided San Francisco into 7 regions as in Figure 7 and using the traceset of mobility of taxi cabs data from CRAWDAD [45], we obtained the average arrival rates and travel times between regions (we exclude the rides occurring in the same region).
Fig. 8: Comparison of different policies for San Francisco case study. The legends for all figures are the same as the top left figure, where red lines correspond to the dynamic and blue lines correspond to the static policies. In all scenarios, we use the rewards generated and the total queue length as metrics. In (a), we demonstrate the results from applying the dynamic and the optimal static policy. In (b), we compare the dynamic policy with a sub-optimal static policy, where the prices are higher than the optimal static policy. In (c), we utilize a surge pricing policy along with the optimal static policy and compare with the dynamic policy.

In Figure 8a, we compare the rewards and the total queue length resulting from the dynamic and the static policy. In Figure 8b, we again change the static policy such that the prices are slightly higher as detailed in Section V-A. In Figure 8c, we use the static policy but also utilize a surge pricing policy in order to keep the queues shorter (See Section V-A). Similar to the case study in Manhattan, the results demonstrate that the performance of the trained dynamic policy is superior to the other policies (we note that the performance can be further improved by longer training).

In Figure 9, we compare the charging costs paid under the dynamic and the static policies. The static policy is generated by using the average value of the electricity prices, whereas the dynamic policy takes into account the current electricity prices before executing an action. Therefore, the dynamic policy provides cheaper charging options by utilizing smart charging mechanisms.

VI. CONCLUSION

In this paper, we developed a dynamic control policy based on deep reinforcement learning for operating an AMoD fleet of EVs as well as pricing for rides. Our dynamic control policy jointly makes decisions for: 1) vehicle routing in order to serve passenger demand and to rebalance the empty vehicles, 2) vehicle charging in order to sustain energy for rides while exploiting geographical and temporal diversity in electricity prices for cheaper charging options, and 3) pricing for rides in order to adjust the potential demand so that the network is stable and the profits are maximized. Furthermore, we formulated the static planning problem associated with the dynamic problem in order to define the capacity region of the dynamic problem and the optimal static policy for the static planning problem. The static policy provides stability of the
Consider the fluid scaling of the queueing network, $Q_{ij}^t = \rho(t)(I_{ij})$ (see [46] for more discussion on the stability of fluid models), and let $Q_{ij}^t$ be the corresponding fluid limit. The fluid model dynamics is as follows:

$$Q_{ij}^t = Q_{ij}^0 + A_{ij}^t - X_{ij}^t,$$

where $A_{ij}^t$ is the total number of riders from node $i$ to node $j$ that have arrived to the network until time $t$ and $X_{ij}^t$ is the total number of vehicles routed from node $i$ to $j$ up to time $t$. Suppose that $\rho^* > 1$ and there exists a policy under which for all $t \geq 0$ and for all origin-destination pairs $(i,j)$, $Q_{ij}^t = 0$. Pick a point $t_1$, where $Q_{ij}^{t_1}$ is differentiable for all $(i,j)$. Then, for all $(i,j)$, $Q_{ij}^t = 0$. Since $A_{ij}^t = \Lambda_{ij}$, this implies $X_{ij}^{t_1} = \Lambda_{ij}$. On the other hand, $X_{ij}^{t_1}$ is the total number of vehicles routed from $i$ to $j$ at $t_1$. This implies $\Lambda_{ij} = \sum_{v=0}^{\max} x_{ij}^{v} \alpha_{ij}^v$ for all $(i,j)$ and there exists $\alpha_{ij}^v$ and $\alpha_{ij}^\infty$ at time $t_1$ such that the flow balance constraints hold and the allocation vector $[\alpha_{ij}^v \alpha_{ij}^\infty]$ is feasible, i.e. $\alpha_{ij}^v + \sum_{j=1}^{m} \alpha_{ij}^\infty \leq 1$.

Now suppose $\rho^* \leq 1$ and $\alpha^* = [\alpha_{ij}^v \alpha_{ij}^\infty]$ is an allocation vector that solves the static problem. The cumulative number of vehicles routed from node $i$ to $j$ up to time $t$ is $S_{ij}^t = \sum_{v=0}^{\max} x_{ij}^{v} t = \sum_{v=0}^{\max} x_{ij}^{v} t \geq \Lambda_{ij} t$. Suppose that for some origin-destination pair $(i,j)$, the queue $Q_{ij}^t \geq \epsilon > 0$ for some positive $t_1$ and $\epsilon$. By continuity of the fluid limit, there exists $t_0 \in (0, t_1)$ such that $Q_{ij}^{t_0} = \epsilon/2$ and $Q_{ij}^t > 0$ for $t \in [t_0, t_1]$. Then, $Q_{ij}^t > 0$ implies $\Lambda_{ij} > \sum_{v=0}^{\max} x_{ij}^{v} \alpha_{ij}^v$, which is a contradiction.

**B. Proof of Proposition 2**

For brevity of notation, let $\beta + P_i = P_i^*$. Let $\nu_{ij}$ be the dual variables corresponding to the demand satisfaction constraints and $\mu_i^v$ be the dual variables corresponding to the flow balance constraints. Since the optimization problem (7) is a convex quadratic maximization problem (given a with uniform $F(i)$) and Slater’s condition is satisfied, strong duality holds. We can write the dual problem as:

$$\min_{\nu_{ij}, \mu_i^v} \max_{\ell_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( \lambda_{ij} (1 - \ell_{ij}) / \ell_{ij} \right) \left( \ell_{ij} - \nu_{ij} \right)$$

subject to

$$\nu_{ij} \geq 0,$$

$$\nu_{ij} + \mu_i^v - \mu_j^{v-v_i} - \beta \tau_{ij} \leq 0,$$

$$\mu_i^v - \mu_j^{v+1} - P_i \leq 0 \quad \forall i, j, v.$$

For fixed $\nu_{ij}$ and $\mu_i^v$, the inner maximization results in the optimal prices:

$$\ell_{ij}^* = \frac{\ell_{\max} + \nu_{ij}}{2}.$$

By strong duality, the optimal primal solution satisfies the dual solution with optimal dual variables $\nu_{ij}^*$ and $\mu_i^{v*}$, which completes the first part of the proposition. The dual problem with optimal prices in (13) can be written as:

$$\min_{\nu_{ij}, \mu_i^v} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( \lambda_{ij} \left( \ell_{\max} - \nu_{ij} \right)^2 \right)$$

subject to

$$\nu_{ij} \geq 0,$$

$$\nu_{ij} + \mu_i^v - \mu_j^{v-v_i} - \beta \tau_{ij} \leq 0,$$

$$\mu_i^v - \mu_j^{v+1} - P_i \leq 0 \quad \forall i, j, v.$$

The objective function in (14a) with optimal dual variables, along with (13) suggests:

$$P = \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_{ij} \left( \ell_{\max} - \ell_{ij}^* \right)^2,$$

where profits $P$ is the value of the objective function of both optimal and dual problems. To get the upper bound on prices, we go through the following algebraic calculations using the constraints. The inequality (14d) gives:

$$\mu_i^{v-v_i} \leq \nu_{ij} P_i + \mu_i^v,$$

and equivalently:

$$\mu_i^{v-v_i} \leq \nu_{ij} P_i + \mu_i^v.$$

The inequalities (14c) and (14b) yield:

$$\mu_i^v - \mu_j^{v-v_i} - \beta \tau_{ij} \leq 0,$$

and equivalently:

$$\mu_i^v - \mu_j^{v-v_i} - \beta \tau_{ij} \leq 0.$$

Inequalities (15) and (17):  

$$\mu_i^v \leq \mu_i^v + \beta \tau_{ij} + v_j P_i.$$

And finally, the constraint (14c):  

$$\nu_{ij} \leq \beta \tau_{ij} + \mu_j^{v-v_i} - \mu_i^v,$$

$$\leq \beta \tau_{ij} + v_j P_j + \mu_j^v - \mu_i^v,$$

$$\leq \beta \tau_{ij} + v_j P_j + \beta \tau_{ij} + v_j P_i.$$

Replacing $P_i$ with $\beta$ and rearranging the terms:

$$\nu_{ij} \leq \beta (\tau_{ij} + v_j) + v_j P_j + v_j P_i.$$

Using the upper bound on the dual variables $\nu_{ij}$ and (13), we can upper bound the optimal prices.
BERKAY TURAN is pursuing the Ph.D. degree in Electrical and Computer Engineering at the University of California, Santa Barbara. He received the B.Sc. degree in Electrical and Electronics Engineering as well as the B.Sc. degree in Physics degree from Boğaziçi University, Istanbul, Turkey, in 2018. His research interests include optimization and reinforcement learning for the design, control, and analysis of smart infrastructure systems such as the power grid and transportation systems.

RAMTIN PEDARSANI is an Assistant Professor in ECE Department at the University of California, Santa Barbara. He received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2009, the M.Sc. degree in communication systems from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2011, and his Ph.D. from the University of California, Berkeley, in 2015. His research interests include machine learning, intelligent transportation systems, and information theory. Ramtin is a recipient of the IEEE international conference on communications (ICC) best paper award in 2014.

MAHNOOSH ALIZADEH is an assistant professor of Electrical and Computer Engineering at the University of California Santa Barbara. Dr. Alizadeh received the B.Sc. degree in Electrical Engineering from Sharif University of Technology in 2009 and the M.Sc. and Ph.D. degrees from the University of California Davis in 2013 and 2014 respectively, both in Electrical and Computer Engineering. From 2014 to 2016, she was a postdoctoral scholar at Stanford University. Her research interests are focused on designing scalable control and market mechanisms for enabling sustainability and resiliency in societal infrastructures, with a particular focus on demand response and electric transportation systems. Dr. Alizadeh is a recipient of the NSF CAREER award.