Transportation Research Record

Deep Reinforcement Learning Algorithm for Dynamic Pricing of Express Lanes with Multiple Access Locations

--Manuscript Draft--

Full Title:	Deep Reinforcement Learning Algorithm for Dynamic Pricing of Express Lanes with Multiple Access Locations
Abstract:	This article develops a deep reinforcement learning (Deep-RL) framework for dynamic pricing on managed lanes with multiple access locations and with heterogeneity in travelers' value of time, origin, and destination. The problem is formulated as a partially observable Markov decision process (POMDP) and policy gradient algorithms are used to determine tolls as a function of real-time observations. The method is compared against feedback control method for dynamic pricing. We show that Deep-RL is effective in learning toll policies for multiple objectives like maximizing revenue, minimizing total system travel time, and objectives with policy constraints, when tested on real-world transportation networks. The Deep-RL toll policies outperform the feedback control heuristic for the revenue maximization objective by generating revenue 8%-2406% higher than the heuristic. We also propose reward shaping methods for the POMDP to overcome undesired behavior of toll policies, like the jam-and-harvest behavior of revenue maximizing policies. Additionally, we test transferability of the algorithms trained on one set of inputs for new input distributions and offer recommendations on real-time implementations of Deep-RL algorithms.
Manuscript Classifications:	Data and Information Technology; Artificial Intelligence and Advanced Computing Applications ABJ70; Intelligent Agents; Congestion Pricing and User-Based Fees ABE25; Tolling
Manuscript Number:	20-01951
Article Type:	Presentation
Order of Authors:	Venktesh Pandey, M.S.
	Evana Wang
	Stephen D Boyles, Associate Professor

Deep Reinforcement Learning Algorithm for Dynamic Pricing of Express Lanes with Multiple Access Locations

Venktesh Pandey
Graduate Research Assistant
Department of Civil, Architectural and Environmental Engineering
The University of Texas at Austin
301 E. Dean Keeton St. Stop C1761
Austin, TX 78712-1172
Ph: 737-222-8473, FAX: 512-475-8744
Email: venktesh@utexas.edu (*Corresponding author.*)

Evana Wang
Undergraduate Researcher
Department of Civil, Architectural and Environmental Engineering
The University of Texas at Austin
301 E. Dean Keeton St. Stop C1761
Austin, TX 78712-1172
Email: evienctx@utexas.edu

Stephen D. Boyles
Associate Professor
Department of Civil, Architectural and Environmental Engineering
The University of Texas at Austin
301 E. Dean Keeton St. Stop C1761
Austin, TX 78712-1172
Ph: 512-471-3548, FAX: 512-475-8744
Email: sboyles@mail.utexas.edu

Word count:
6251 words text+
172 words abstract+
500 words references+
2 tables $\times$250 words (each) = 7424 words

Submission date: August 1, 2019

ABSTRACT

This article develops a deep reinforcement learning (Deep-RL) framework for dynamic pricing on managed lanes with multiple access locations and with heterogeneity in travelers' value of time, origin, and destination. The problem is formulated as a partially observable Markov decision process (POMDP) and policy gradient algorithms are used to determine tolls as a function of real-time observations. The method is compared against feedback control methods for dynamic pricing. We show that Deep-RL is effective in learning toll policies for multiple objectives like maximizing revenue, minimizing total system travel time, and objectives with policy constraints, when tested on real-world transportation networks. The Deep-RL toll policies outperform the feedback control heuristic for the revenue maximization objective by generating revenue 8%-2406% higher than the heuristic. We also propose reward shaping methods for the POMDP to overcome undesired behavior of toll policies, like the *jam-and-harvest* behavior of revenue maximizing policies. Additionally, we test transferability of the algorithms trained on one set of inputs for new input distributions and offer recommendations on real-time implementations of Deep-RL algorithms.

Keywords— Managed lanes, Express lanes, HOT lanes, Dynamic pricing, Deep reinforcement learning, Continuous control, Feedback control heuristics.

INTRODUCTION

Priced managed lanes (MLs), also referred to as express lanes or high-occupancy/toll (HOT) lanes, are increasingly being used by many cities to mitigate traffic congestion and provide reliable travel time. As of January 2019, there are 41 managed lane projects across the United States *(1)*. On these lanes, travelers pay a toll which changes either with the time of day, or dynamically based on the congestion pattern, to experience less congested travel time. In recent years, managed lane networks have become increasingly complex, spanning longer corridors and having multiple entrance and exit locations. For example, the LBJ TEXpress lanes in Dallas, TX have 17 entrance ramps and 18 exit ramps, and three tolling segments with different time-varying toll values *(2)*.

Dynamic pricing for express lanes with multiple access points is a complex control problem due to the heterogeneity in lane choice behavior of travelers belonging to different classes. Vehicles differ in their value of time and their destination of travel, both of which impact the pricing structure. Predicting driver behavior with certainty is a difficult process. A recent study showed that a binary logit model, commonly used for modeling lane choice, is inadequate in predicting heterogeneity in lane choice decisions *(3)*.

Several dynamic pricing algorithms have been explored that optimize tolls under varying assumptions on driver behavior. These include methods using stochastic dynamic programming *(4)*, hybrid model predictive control (MPC) *(5, 6)*, reinforcement learning (RL) *(7, 8)*, and approximate dynamic programming *(9)*. While the current algorithms do well against the existing heuristics, they make the following restricting assumptions, which we relax in this study.

1. Restricted access for travelers: travelers do not exit the managed lane once they enter till their exit is reached *(4, 7)* and that they only consider the first entry point as the decision point for the lane choice decision *(5)*
2. Fully observable system: toll operators have access to measurements of traffic density throughout the network for optimizing tolls *(4, 5, 7, 8, 9)*
3. Ignored traveler heterogeneity: a single vehicle class is considered with a single origin and destination *(4, 7, 9)*
4. Simplified traffic dynamics: traffic dynamics are simplified with assumptions like the flow dynamics on general-purpose lanes (GPLs) are independent of vehicles using the managed lane *(4)*; or that the proportion of flow split at diverge points is identical for all origins *(5)*

In addition, there are relatively few analysis on the conflict between optimization of multiple objectives with realistic constraints. Pandey and Boyles *(9)* showed that the revenue-maximizing tolls exhibit a *jam-and-harvest* (JAH) nature where the parallel GPLs are intentionally jammed to congestion in earlier time periods to harvest more revenue towards the end. Handling such undesirable behavior of optimal policies has not been studied in the literature.

Furthermore, practical applicability of these algorithms in real-world environments is a less explored question. Algorithms that optimize prices using a simulation model can be applied in real-time using lookup tables. However, the transferability analysis of such lookup tables to new input distributions is not considered *(4, 7, 9)*. The hybrid MPC algorithm in Tan and Gao *(5)* incorporates real-time measurements for optimizing tolls over a finite horizon with computation time in the range of 1.2–2.6 seconds for a 30 seconds optimization horizon. This is sufficient for a real-time implementation; however, the tests conducted are limited to one test network under

two scenarios of demand, assuming full observability of the system. Solving a MPC-based model with heterogeneous vehicle classes and partial observability of the system is complex and not fully studied. We thus require scalable algorithms for real-world networks that relax the assumptions on driver behavior and traffic flow, and transfer well from simulation settings to new input distributions.

In this research, we use deep reinforcement learning (Deep-RL) algorithms for optimizing tolls without making simplifying assumptions in the earlier literature. The algorithms rely on real-time density observations using sensors (such as loop detectors) located only at certain locations without access to information about the demand distribution or driver characteristics like the value of time (VOT) distribution. We formulate the dynamic pricing problem as a Deep-RL problem, use standard algorithms based on policy gradient to solve the problem, and compare their performance against the existing feedback control methods. Our framework considers multiple origins and destinations, multiple access points to the managed lane facility, *en route* diversion of vehicles at each diverge point, and partial observability of the systems. We investigate the usefulness of Deep-RL as a tool in our toolbox for dynamic pricing and explain its advantages and limitations by experiments on four different test networks.

The primary contributions of this article are:

- We demonstrate the usefulness of Deep-RL algorithms for solving the dynamic pricing problem under partial observability and show that it performs well against existing heuristics, without requiring restricting assumptions on driver behavior or traffic dynamics
- We apply multi-objective optimization methods to overcome undesirable JAH characteristics of revenue-maximizing optimal policies
- We conduct tests to verify the transferability of learned Deep-RL algorithms to new input values and make recommendations on real-time implementation of the algorithm

The rest of the article is organized as follows. The second section presents an overview of the related work. The third section introduces the notation, presents the details of the model, and discusses the solution algorithms. The fourth section presents the results from experimental analysis on four different test networks, and the final section concludes the paper and identifies directions for future work.

LITERATURE REVIEW

Control problems in the area of transportation are broadly solved using three methods: open-loop optimal control methods (that solve the optimal control problem without incorporating real-time measurements), closed-loop control methods like MPC (that incorporate the feedback of real-time measurements and optimize over a rolling horizon), and RL methods where the optimal control is learned with an iterative interaction with the environment. The applications of reinforcement learning methods for traffic control are a few, with adaptive traffic signal control (ATSC) being a prominent explored area of research. See Yan et al. *(10)* and Zhao et al. *(11)* for a survey of RL methods for signal control.

Managed lane (or HOT) pricing problem is also a traffic control problem where the chosen control directly impacts the driver behavior and thus the congestion pattern. There are three com-

ponent models to the HOT pricing problem *(12)*: a *lane choice model* that determines how travelers choose a lane given the tolls and travel times, a *traffic flow model* that models the interaction of vehicles in simulated environments, and a *toll pricing model* which determines the toll pricing objectives and how the optimization problem is solved to achieve the best value of the objective.

Toll pricing models for MLs with a single access point are commonly studied. Gardner et al. *(12)* argued that for ML with a single entrance and exit, the tolls minimizing the total system travel time (TSTT) also utilize the managed lanes to full capacity at all times. The authors developed an analytical formulation for tolls minimizing TSTT as a function of the VOT distribution. Lou et al. *(13)* used a self-learning approach for optimizing toll prices where the average VOT values were learned using real-time measurements. Toledo et al. *(6)* used a rolling horizon approach to optimize future tolls with predicted demand from traffic simulation; however, the method of exhaustive search to solve the non-convex control problem does not scale well for large managed lane networks.

For managed lanes with multiple access points, Tan and Gao *(5)* presented a formulation optimizing the proportion of vehicles entering the managed lane instead of directly optimizing toll prices. The authors showed a one-to-one mapping between optimal toll prices and the proportion values, and transformed the control problem into a mixed-integer linear program which can be solved efficiently for networks with multiple access points. Dorogush and Kurzhanskiy *(14)* used a similar method and optimized split ratios at each diverge, which are then used to determine toll prices; however, their analysis ignored the variation of incoming flow at each diverge. Apart from these optimal control based methods, Zhu and Ukkusuri *(7)* and Pandey and Boyles *(9)* used RL methods, where the control problem is formulated as a Markov decision process (MDP) and the value function (or its equivalent, Q-function) is learned by iterative interactions with the environment. However, the tests are conducted for discrete state and action spaces assuming full observability of the system. This article is guided by the advances in RL methods and improves these earlier RL-based approaches for dynamic pricing.

Deep-RL improves traditional RL by replacing function approximators with deep neural networks which has been effective in various control problems. See Arulkumaran et al. *(15)* for a survey of Deep-RL applications. Deep-RL algorithms have also been used for other traffic control problems. Belletti et al. *(16)* developed an "expert-level" control of coordinated ramp metering using Deep-RL methods with multiple agents and achieved precise adaptive metering without requiring model calibration that does better than the traditional benchmark algorithm named ALINEA. Wu et al. *(17)* used Deep-RL algorithms to solve the control problem of selecting the acceleration and brake of multiple autonomous vehicles (AVs) under conditions of mixed human vehicles and AVs to mitigate traffic congestion. When compared against classical approaches, their approach generated 10-20% lower TSTT. Other applications of Deep-RL algorithms are in the domain of ATSC including traditional one signal control *(18, 19)*, coordinated control of traffic signals *(20)*, and large-scale multiagent control using Deep-RL methods *(21)*. See Yau et al. *(10)* for a review of RL algorithms in the area of ATSC. With an exception of Belletti et al. *(16)*, all other Deep-RL models in transportation domain used microsimulation to capture the vehicle-to-vehicle interactions.

DEEP REINFORCEMENT LEARNING MODEL FOR DYNAMIC PRICING

Notation and Assumptions

Consider the directed network shown in Figure 1 which is an abstraction of a managed lane network. The upper set of links form MLs and the lower set of links form GPLs. As we describe the network, we label the assumptions made in our model as "$A\#$". We also label ideas for future work as "FW$\#$".

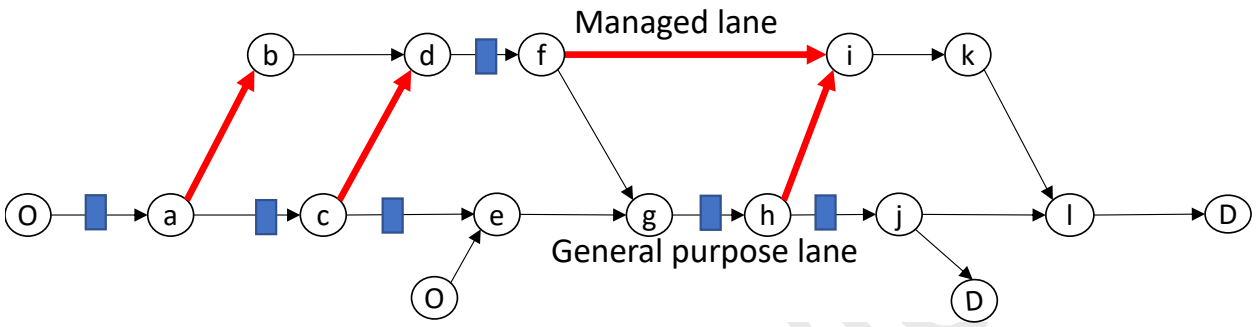


FIGURE 1 Managed lane network with multiple entrance and exit where links with higher thickness are tolled, and links with a box are observed by the toll operator

The time horizon is divided into equal time steps, each Δt units long. The set of all time periods is given by $\mathscr{T} = \{t_0, t_1, t_2, \ldots, t_{T/\Delta t}\}$, where T is the time horizon. Tolls are updated after every $\Delta\tau = m\Delta t$ time units, where m is a positive integer fixed by the tolling agency. Define $\mathscr{T}_\tau = \{k \mid t_{km} \in \mathscr{T}, \text{ where } k \in \{0, 1, 2, \ldots\}\}$ as the set of time periods where tolls are updated, indexed in increasing order of positive integers. Then, $|\mathscr{T}_\tau| = T/\Delta\tau$ represent the number of toll updates throughout the simulation.

Let N represent the set of all nodes and $A = \{(i, j) \mid i, j \in N\}$ represent the set of all links in the network. Let N_o and N_d denote the set of all origins and all destinations, respectively. We assume that origins and destinations connect to the network through nodes on the GPLs $(A\#1)$. The demand between an origin and a destination is a random variable. A toll operator does not know the demand distribution, but only relies on the observed realizations of demand. However, for simulation purposes, we model the demand of vehicles from origin $r \in N_o$ to destination $s \in N_d$ at time $t \in \mathscr{T}$ to be a rectified Gaussian random variable with mean $d_{rs}(t)$ and standard deviation σ_d, and ignore correlations of demand between different origin-destination (OD) pairs.

Let V denote the set of all values of VOT (assumed to a discrete distribution for the population, $A\#2$) and p_v for any $v \in V$ be the proportion of demand with VOT v. The p_v values are unknown to a toll operator. For simulation purposes, we choose the VOT distribution $(p_v \mid v \in V)$ and σ_d to be identical for all origin-destination pairs.

In contrast to the cell-based representation of managed lane network where MLs and GPLs are part of the same cell *(4, 5, 14)*, we divide each link on GPL or ML into individual cells. This choice lets us use the traditional cell-transmission model (CTM) equations from Daganzo *(22)* for modeling traffic flow. Let $\mathscr{C}_{(i,j)}$ represent the set of all cells for link $(i, j) \in A$ and $\mathscr{C} = \bigcup_{(i,j)\in A} \mathscr{C}_{(i,j)}$ denote the set of all cells in the network. The length of each cell $c \in \mathscr{C}$, denoted by l_c, is determined using the usual requirements of the CTM in *(22)* and is assumed

constant for all links in the network ($A\#3$). Let $l_{ij}, \nu_{ij}, q_{ij}, w_{ij}$, and $k_{\text{jam},ij}$ represent the length, free-flow speed, capacity, back-wave speed, and jam density, respectively, for link $(i,j) \in A$ as its fundamental diagram parameters.

A toll operator is assumed to manage the toll rate at each on-ramp and diverge point beyond a diverge on a ML ($A\#4$). We assume this toll structure because it inherently models the constraint that traveling longer distance on the ML levies a higher toll than traveling shorter distance. For a detailed discussion on various tolling options on managed lane networks with multiple access points, refer Pandey and Boyles *(23)*. Let A_{toll} represent the links where tolls are collected and Figure 1 highlights the link in bold. We denote the toll charged on link $(i,j) \in A_{\text{toll}}$ for any $t \in \mathscr{T}$ by $\beta_{ij}(t)$.

Travelers make routing decisions at each diverge using the received information about travel time and toll values while traveling towards their destination. Nodes a, c, f, and h are the diverge locations for the network in Figure 1. We assume that the information about the travel time is provided by measuring instantaneous travel time ($A\#5$), and that all travelers make lane choice decisions only using the instantaneous/real-time information and do not rely on historic information (obtained from prior experience) for making lane choices ($A\#6$). Assumptions $A\#5$ and $A\#6$ are only made for simulation purposes as the Deep-RL model only requires the realization of lane choice by each traveler in form of observed density measurements at loop detector locations. Though dynamic traffic assignment models have been used in the literature for optimization of toll prices for express lanes *(24)*, we focus on real-time optimization of toll prices and ignore route-choice equilibration of travelers ($A\#7$). Considering dynamic equilibrium while optimizing a dynamic stochastic control is a complex problem and will be studied as part of the future work (FW$\#1$).

Similar to earlier instances, the Deep-RL algorithm developed in this research is agnostic to the lane choice model. For simulation purposes, we focus our attention on two models: multiple VOT classes with two routes and stochastic choice (termed multiclass binary logit model) and multiple VOT classes with decision routes and deterministic choice (termed multiclass decision route model; refer Pandey and Boyles *(9)*). For simulation purposes, we evaluate the utility of a route as the linear combination of the toll and route's travel time, converted to the same units using the VOT value of the class ($A\#8$).

POMDP

Partially observable Markov decision processes (POMDPs) are MDPs where the states are not fully observable. This is suitable for cases where a toll operator does not have access to traffic information throughout the network but only at certain locations. We define the dynamic pricing problem as a POMDP with following components:

- **Timestep**: Tolls are to be optimized over a finite time horizon for each time $k \in \mathscr{T}_\tau$. A finite horizon can represent a morning or an evening peak period on a corridor, or an entire day.
- **State**: We define $x_c^z(t)$ as the number of vehicles in cell $c \in \mathscr{C}$ belonging to class $z \in Z$ at time $t \in \mathscr{T}$, where $Z = \{(v,d) \mid v \in V, d \in N_d\}$ is the set of all classes, disaggregated by the VOT value and the destination of the vehicle (the origin of a vehicle does not influence

the tolls once the vehicle is on the road and is thus ignored). For ML networks where high occupancy vehicles pay a different toll than single/low occupancy vehicles, we can extend Z to include the occupancy level of vehicles, but we leave that analysis for future work (FW#2). The dimensionality of Z impacts the computational performance of the multiclass cell transmission model. We denote the state of the POMDP by s which comprises of the current toll update step $k \in \mathscr{T}_\tau$ and the values $x_c^z(t_{k\Delta\tau})$ for all cells $c \in \mathscr{C}$ and class $z \in Z$. Thus, the state space S can be written as Equation (1).

$$S = \{(k, x_c^z(t_{k\Delta\tau})) \mid k \in \mathscr{T}_\tau, c \in \mathscr{C}, z \in Z\} \tag{1}$$

- **Observation**: In our model, the observation is done using loop detectors. The loop detectors measure the total number of vehicles going from one cell to the next and thus cannot distinguish between vehicles belonging to different classes, so the state is not fully observable. The observation space depends on the link location of loop detectors denoted by set $A_{\text{loop}} \subseteq A$. For the network in Figure 1, $A_{\text{loop}} = \{(o,a),(a,c),(c,e),(d,f),(g,h),(h,j)\}$. These locations are a variable in our model and we conduct sensitivity of results to changes in observation space later in the text. Let $\boldsymbol{o}(s)$ denote the observation vector for state s and comprise of the measurement of total number of vehicles on each link $(i,j) \in A_{\text{loop}}$. Mathematically, $\boldsymbol{o}(s) = \{\sum_{z\in Z}\sum_{c\in\mathscr{C}_{(i,j)}} x_c^z(t_{k\Delta\tau}) \mid (i,j) \in A_{\text{loop}}\}$. The actual observation is assumed to be Gaussian random variable with the mean as specified and the standard deviation σ_o which models the noise in loop detector measurements. We project negative values of observation, if any, to zero.

- **Action**: Action a in state s is the toll $\beta_{ij}(t_{k\Delta\tau})$ charged for a toll link $(i,j) \in A_{\text{toll}}$, where $\beta_{ij}(\cdot) \in (\beta_{\min}, \beta_{\max})$. The action is modeled as a continuous variable; the values can be rounded to nearest tenth of a cent or dollar if desired.

- **Transition function**: The transition of the POMDP from a state s to a new state s' given action a, is governed by the traffic flow equations from the CTM model which incorporates the lane choice behavior of travelers. For simulation purposes, we assume that traffic flow throughout the network is deterministic except at diverges where the lane choices of travelers may be stochastic $(A\#9)$. We use a multiclass version of the CTM model from the literature *(9)*.

- **Reward**: The reward obtained after taking action a in state s, denoted by $r(s,a)$, depends on the choice of tolling objective. We consider two objectives, revenue maximization and total system travel time (TSTT) minimization, with following definitions of reward:

  - Revenue maximization:

$$r^{\text{RevMax}}(s,a) = \sum_{x=k\Delta\tau}^{(k+1)\Delta\tau} \sum_{(i,j)\in A_{\text{toll}}} \left(\beta_{ij}(t_{k\Delta\tau}) \sum_{(h,i)\in A} y_{hij}(t_x) \right) \tag{2}$$

    where $y_{hij}(t)$ is the total flow going from link $(h,i) \in A$ to $(i,j) \in A$ from time step t to time step $t + \Delta t$

– Total system travel time minimization:

$$r^{\text{TSTTMin}}(s,a) = -\left(\sum_{x=k\Delta\tau}^{(k+1)\Delta\tau}\sum_{c\in\mathscr{C}}\sum_{z\in Z} x_c^z(t_x)\right) \tag{3}$$

where the negative sign is used to ensure that reward maximization is equivalent to TSTT minimization.

To overcome the undesired JAH nature, we use reward shaping methods that seek to find policies with less or no JAH behavior. We quantify the JAH behavior using a statistic defined as a numeric value at the end of simulation. The statistic, denoted by JAH_1, measures the maximum of difference between the number of vehicles in GPL to the number of vehicles in ML across all time steps. It is defined as in Equation (4), where $A_{\text{GPL}}(A_{\text{ML}})$ are links on the GPL (ML). The value of JAH_1 is dependent on network properties like number of lanes in GPL and ML.

$$\text{JAH}_1 = \max_{t\in\mathscr{T}}\left(\sum_{(i,j)\in A_{\text{GPL}}}\sum_{c\in\mathscr{C}_{(i,j)}}\sum_{z\in Z} x_c^z(t) - \sum_{(i,j)\in A_{\text{ML}}}\sum_{c\in\mathscr{C}_{(i,j)}}\sum_{z\in Z} x_c^z(t)\right) \tag{4}$$

For the given POMDP, a policy $\pi_\theta(a|s)$ denotes the probability of taking action a given state s. We consider stochastic policies parameterized by a vector of parameters θ. For example, for a policy replaced by a neural network, θ represents the flattened weights and biases for the nodes in the network. Since the action space for the POMDP is continuous, the neural network outputs the mean of the Gaussian distribution of tolls which is then used to sample continuous actions. We assume the covariance of the joint distribution of actions to be a diagonal matrix with constant diagonal terms $(A\#10)$.

Episodic RL

In an episodic RL problem, an agent's experience is broken into episodes, where an episode is a sequence with finite number of states, actions, and rewards. Since the POMDP introduced in the previous subsection is finite-horizon, the simulation terminates at time $T/\Delta t$. Thus, an episode is formed by a sequence of states, actions, and rewards for each time step $k \in \mathscr{T}_\tau$.

We first define a trajectory $\aleph$ as a sequence of states and actions visited in an episode, that is $\aleph = (s_0, a_0, s_1, a_1, \cdots, s_{|\mathscr{T}_\tau|-1})$, where s_k is same as the state defined earlier indexed by the time k in that state. Let $r(s_k, a_k)$ be denoted by r_k for all $k \in \mathscr{T}_\tau$. The goal of the reinforcement learning problem is to find a policy that maximizes the expected reward over the entire episode. The optimization problem can then be written as following:

$$\max_{\pi_\theta(\cdot)} J(\pi_\theta) = \mathbb{E}_\aleph[R(\aleph)|\pi] \tag{5}$$

$$R(\aleph) = \sum_{k\in\mathscr{T}_\tau} r_k \tag{6}$$

where, $\mathbb{E}_{\aleph}[R(\aleph)|\pi] = \int R(\aleph)p_{\pi}(\aleph)d\aleph$ is the expected reward over all possible trajectories obtained after executing policy π with $p_{\pi}(\aleph)$ as the probability distribution of trajectories obtained by executing policy π. Note that we do not consider discounting for future rewards.

The solution of this POMDP is a vector θ^* that determines the policy which optimizes the objective under certain constraints on the policy space. Commonly considered policy constraints for the dynamic pricing of express lanes include the following:

1. Tolls levied for a longer distance are higher than tolls levied for a shorter distance: with the choice of tolling structure (assumption $A\#4$) where tolls are charged at every diverge, this constraint is already satisfied.
2. The ML is always operated at a speed higher than the minimum speed limit: in our model, we allow violation of this constraint on the ML. We observe that, given the stochasticity in lane choice of travelers and demand, bottlenecks can occur at merges and diverges which can result in an inevitable spillover on managed lanes during congested cases. Thus, a hard constraint keeping ML congestion free throughout the learning period is not useful. However, as discussed later, we observe that optimal policies only violate this constraint less than 2% of the time throughout the simulation for all networks tested in the next section.
3. Toll variation from one time step to the next is restricted: we do not explicitly model this constraint. If tolling horizon is "sufficiently" large (say 5 minutes), bigger change in tolls from one toll update to the next can be less of a problem. Though, we observe that the optimal tolls have a structure and do not oscillate significantly.
4. Tolls are upper and lower bounded by a value: we model this by clipping the tolls predicted by the function approximator within the desired range $(\beta_{\min}, \beta_{\max})$.

Next, we discuss the solution methods used to solve the POMDP using Deep-RL methods and other heuristics.

Solution Methods

Deep RL Methods

For solving the POMDP, we use derivative-based policy gradient algorithms which learn the policy directly based on the observations. In this article, we choose two of the commonly used algorithms: the vanilla policy gradient (VPG) and the proximal policy optimization (PPO). Both these methods determine the derivative of the objective function $J(\pi_\theta)$ with respect to the policy parameters θ and improve the parameters using stochastic gradient descent from one iteration to the next. The methods differ in calculation of the derivatives, with PPO providing an improvement over VPG by restricting sudden policy updates. We refer the readers to Schulman *(25)* for a detailed explanation of these standard Deep-RL algorithms. For the experiments, we develop a new RL environment for macroscopic simulation of traffic and customize the open-source implementation of both algorithms in Python provided by OpenAI Spinningup *(26)*.

Feedback control heuristic

We compare the performance of Deep-RL algorithms against a feedback control (FC) heuristic based on the measurement of total number of vehicles in the links on ML. We customize the `Density` heuristic in Pandey and Boyles *(9)* to charge varying tolls for different toll links.

Define $\text{ML}(i,j)$ as the set of links on the ML used by a traveler who enters the ML using the toll link $(i,j) \in A_{\text{toll}}$. For the network in Figure 1, $\text{ML}(a,b) = \{(b,d)\}$, $\text{ML}(c,d) = \{(d,f)\}$, $\text{ML}(f,i) = \{(f,i)\}$, and $\text{ML}(h,i) = \{(i,k)\}$. The FC heuristic updates the tolls for each toll link $(i,j) \in A_{\text{toll}}$ based on the density observations on links in $\text{ML}(i,j)$. The toll value for an update time $(k+1) \in \mathscr{T}_\tau$ is based on the toll value in the previous update step tweaked by the difference in values of desired number of vehicles to current number of vehicles. The toll update is given by Equation (7),

$$\beta_{ij}(t_{(k+1)\Delta\tau}) = \beta_{ij}(t_{k\Delta\tau}) + P \times \left(X_{\text{ML}(i,j)}(k) - X^{\text{desired}}_{\text{ML}(i,j)}\right) \tag{7}$$

where $X_{\text{ML}(i,j)}(k)$ is the total number of vehicles on links in $\text{ML}(i,j)$ before updating tolls at time $k+1$ and $X^{\text{desired}}_{\text{ML}(i,j)}$ be the desired value of the number of vehicles on the links in $\text{ML}(i,j)$. P is the regulator parameter, with units $\$/\text{veh}$, controlling the influence of difference between the desired and current number of vehicles on the toll update. The commonly used desired value is the number of vehicles corresponding to the critical density on the ML link. We generalize the desired number of vehicles by defining $X^{\text{desired}}_{\text{ML}(i,j)}$ as:

$$X^{\text{desired}}_{\text{ML}(i,j)} = \sum_{(g,h)\in\text{ML}(i,j)} \eta k_{\text{critical},(g,h)} l_{gh} \tag{8}$$

where, $k_{\text{critical},(g,h)}$ is the critical density for link $(g,h) \in A$ and η is the scaling parameter varying between $(0,1]$ that sets the desired number of vehicles to a proportion value of the number of vehicles at critical density. We calibrate the FC heuristic for different values of desired density and regulator parameter. We do not include other algorithms for comparison because of lack of compatibility or details. The algorithms using discrete variables do not scale for continuous observation space and tolls *(7, 9)*. Comparing the performance of Deep-RL methods against the hybrid MPC method in Tan and Gao *(5)* requires extensive analysis and will be a part of the future work (FW#3).

EXPERIMENTAL ANALYSIS

Framework

We conduct our analysis on four different networks. The first is a network with single entrance and single exit (SESE) commonly used in the managed lane pricing literature. The next two are the double entrance single exit (DESE) network and the abstract network for the toll segment 2 of the LBJ TEXpress lanes in Dallas, TX (LBJ). The DESE network includes two toll locations for modeling *en route* lane changes. The LBJ network has four toll locations. Last is the network of the northbound Missouri Pacific (MoPac) Express lanes in Austin, TX. The MoPac network has three entry locations to the express lanes and two exit locations.

Figure 2a shows the abstract networks, where the thick lines denote the links where tolls are collected. Demand distribution for the first three networks is artificially generated and follows a two-peak pattern (refer the original demand curve in Figure 2b for the LBJ network) while the demand for the MoPaC network is derived from the dynamic traffic assignment model of the Travis county region. There are a total of 105 origin-destination pairs in the MoPaC network with a total demand of 49, 273 vehicles using the network in three hours of the evening peak.

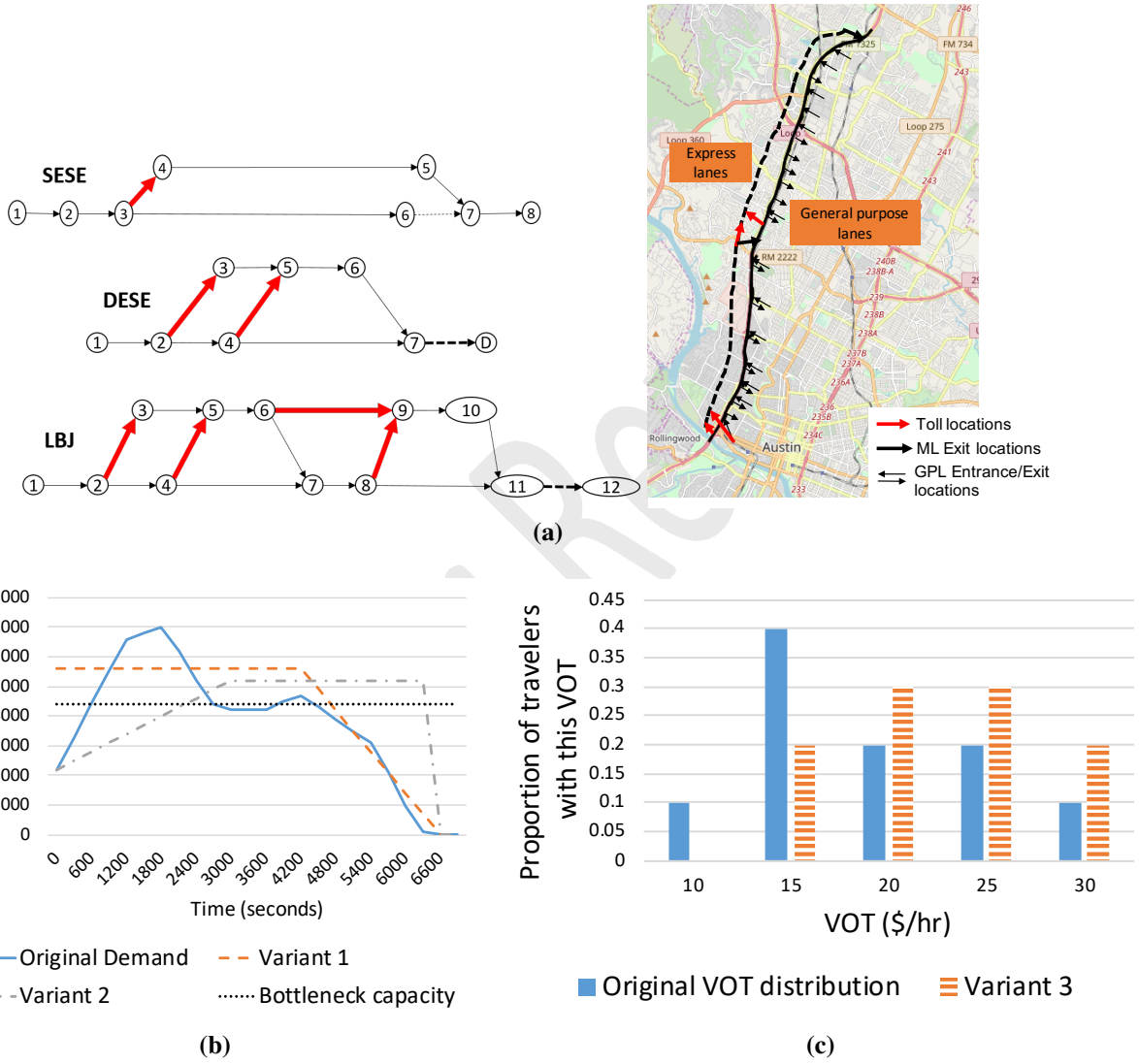


FIGURE 2 (a) Abstract networks for single entrance single exit (SESE) network, double entrance and double exit (DESE) network, LBJ network, and Northbound MoPaC express lane network (latitude-longitude locations of express lanes are shifted to the left to show the locations of toll points and exits from the managed lane), (b) demand distributions used for the SESE, DESE and LBJ networks, and (c) VOT distribution

Table 1 shows the values of parameters used for different networks. Five VOT classes were selected for each network and the same VOT distribution was used. Figure 2c shows the original VOT distribution.

TABLE 1 Values of parameters used in the simulation

	SESE	DESE	LBJ	MoPaC		Parameter	Value
Corridor length (miles)	7.3	1.59	2.91	11.1		β_{min}	\$0.1
Simulation duration (hour)	2	2	2	3		β_{max}	\$4.0
$\Delta\tau$ (seconds)	60	300	300	300		q_{ij} (vphpl)	2200
ν_{ij} (mph)	55	55	55	65		$k_{\text{jam},ij}$ (veh/mile)	265
σ_o (veh/hr)	50	50	50	50		ν_{ij}/w_{ij}	3
σ_d (veh/hr)	10	0	0	100		Δt (seconds)	6

A feedforward multilayer perceptron was selected as the neural network. Hyperparameter tuning was conducted and the architecture with two hidden layers and 64 nodes in each layer was selected. For the MoPaC network, three hidden layers with 128 nodes each were selected. Each network was simulated for a number of iterations ranging between 100 and 200 or until convergence of average reward values.

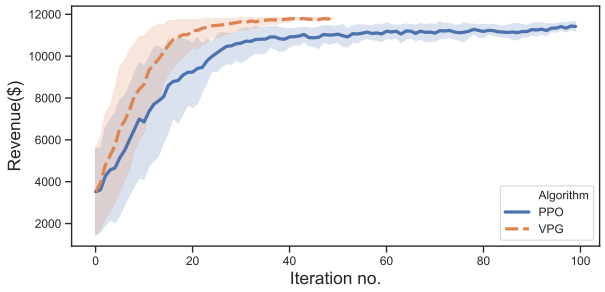
Deep-RL Learning

In this section, we compare the learning performance of the VPG and PPO Deep-RL algorithms for both revenue maximization and TSTT minimization objectives. Figure 3 show the plots of variation of learning for two objectives where the average in each iteration is reported over 10 random seeds.

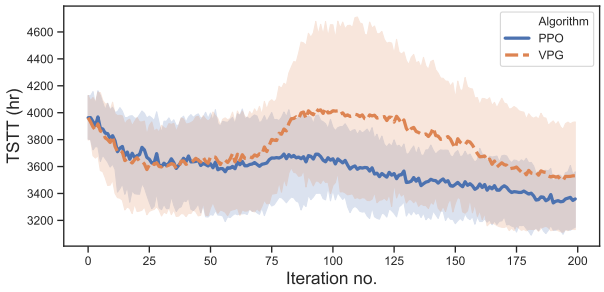
We make following observations. First, both Deep-RL algorithms are able to learn "good" objective values within 200 iterations evident in the increasing trend of the average revenue for the revenue-maximization objective and a decreasing trend of the average TSTT for the TSTT-minimization objective. For the revenue-maximization objective, the average revenue values converge to a high value for all networks. For the TSTT-minimization objective, the average TSTT values for SESE (Figure 3b) and DESE (Figure 3d) networks do not converge; however a decreasing trend is evident. The VPG algorithm for the DESE network in Figure 3d shows divergence towards the end. However, if the simulation is run long enough, the learning converges back to a lower TSTT value.

We argue that learning for the revenue maximization objective is easier than learning for the TSTT minimization objective. This is because the reward definition for revenue maximization in Equation (2) involves the action values (in terms of $\beta_{ij}(\cdot)$) and thus incorporates a direct feedback on the efficiency of current toll. On the other hand, reward for the TSTT minimization objective in Equation (3) does not incorporate the toll values directly. This is known as the *credit assignment problem* in the RL literature where it is unclear which actions over the entire episode were helpful. The *credit assignment problem* can be addressed by reframing the reward definition for the TSTT minimization objective, but this analysis is left as part of the future work (FW$\#4$).

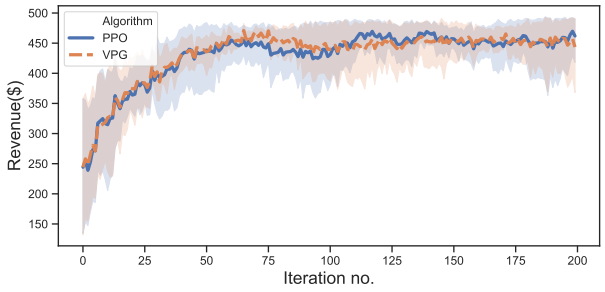
Second, we observe that there is no evident difference in the performance of VPG and PPO algorithms. For the revenue maximization objectives, the algorithms perform "almost identically" with values of average revenue of PPO within $\sim 5\%$ of the average revenue values of VPG algorithm at any iteration. For the TSTT minimization objective, we observe that PPO prevents high variation in average TSTT values from one iteration to the next, whereas the VPG algorithm shows



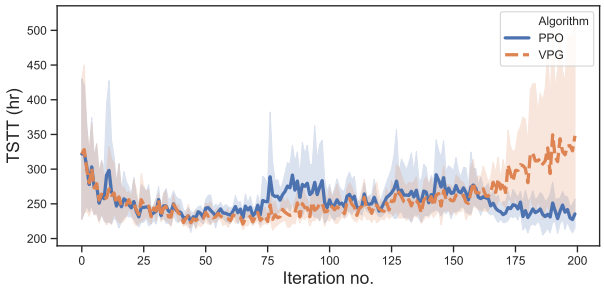
(a) SESE Revenue Maximization.



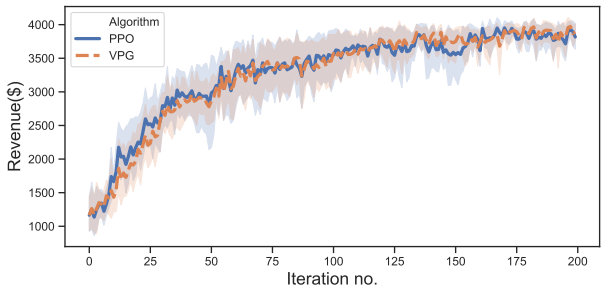
(b) SESE TSTT Minimization.



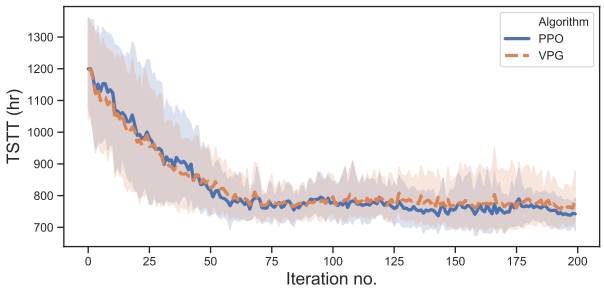
(c) DESE Revenue Maximization.



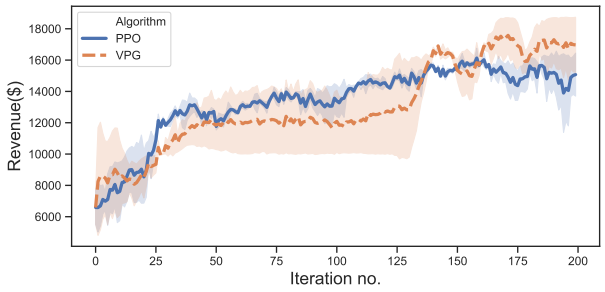
(d) DESE TSTT Minimization.



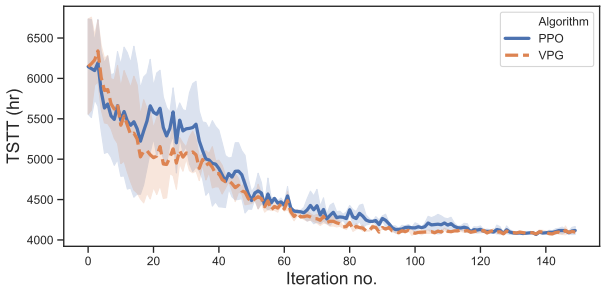
(e) LBJ Revenue Maximization.



(f) LBJ TSTT Minimization.



(g) MoPaC Revenue Maximization.



(h) MoPaC TSTT Minimization.

FIGURE 3 Plot of average objective value with iteration over 10 random seeds for the four networks

higher oscillations (evident in Figures 3b and 3d).

Last, in contrast to our expectation that a larger network with high dimensional action space might require large number of iterations to converge, we observe that for both LBJ and MoPaC networks, the average objectives converge within 200 iterations. This is equivalent to simulating 2000 episodes with $2000 * 2$ hours$/5$ minutes $= 48000$ action interactions with the environment. The computation time for these interactions when run on a Unix machine with 8 GB RAM is 25 min for the LBJ network and 23.39 hours for the MoPaC network. For the MoPaC network $|Z| = 65$ and $|\mathscr{C}| = 258$, and thus updating $65 * 258 = 16,770$ flow variables for every time step is time consuming. Efficient implementation of CTM model with parallel computations can help improve the efficiency of training. We note that the 23.39 hours spent for training are conducted offline on a simulation model. Once the model is trained, it can be transferred with less effort to real-world settings. Thus, we argue that learning is possible within a reasonable number of interactions with the environment even for real-world networks. The amount of data required for training Deep-RL models is often considered its major limitation *(15)*; however, for the dynamic pricing problem it is not a constraining factor.

Impact of Observation Space

We also test the impact of observation space on the learning of Deep-RL algorithms. For the SESE network, the results in Figures 3a and 3b assumed that flows are only observed on certain links: $(3, 6)$, $(6, 7)$, and $(4, 5)$ (call it, Medium observation). We consider two additional observation cases: (a) observing all links in the network (High observation), and (b) only observing link $(3, 6)$ in the network (Low observation). Figure 4 shows the learning results for revenue maximization objectives for the two algorithms for three levels of observation space.

We observe that changing observation space has no significant impact on learning rate, which indicates that we might be better off just relying on sensors on the main GPL for determining the optimal tolls. This behavior is counter-intuitive in a sense that having more sensors installed and collecting more data does not impact the learning rate compared to the case of less number of sensors. We argue that this happens due to the spatial correlation of the congestion pattern on a corridor (where observing additional links does not add a new information for setting the tolls). Our tests for other networks also yielded similar results where changing observation space did not impact the learning.

Multi-objective Optimization

In this section, we focus our attention on multiple optimization objectives together on the LBJ network. We consider how different objective vary with respect to each other for 1000 randomized toll profiles simulated for all four networks. Figure 5 shows the plots of variation of TSTT, the percent of time speed limit constraint is violated on the ML (% violation), and JAH_1 against the revenue obtained from the toll policies for the LBJ network. The figure also shows the values of objectives from the toll profiles generated by Deep-RL algorithms where "DRLRevMax" indicates toll profiles from the revenue maximization objectives and "DRLTSTTMin" indicates toll profiles from the TSTT minimization objective.

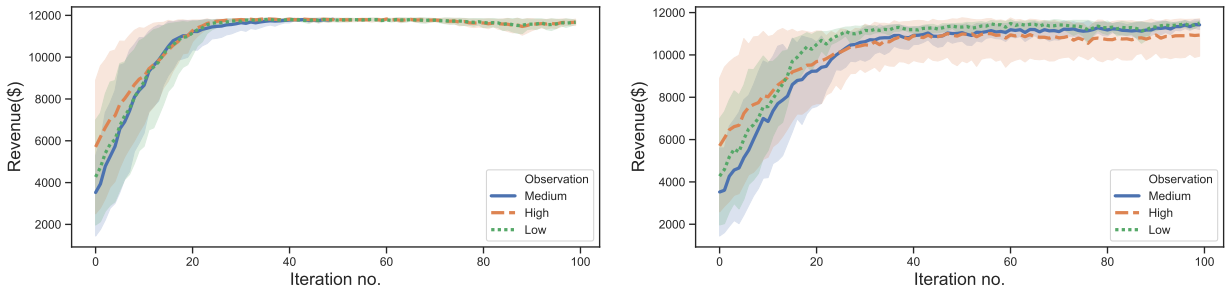


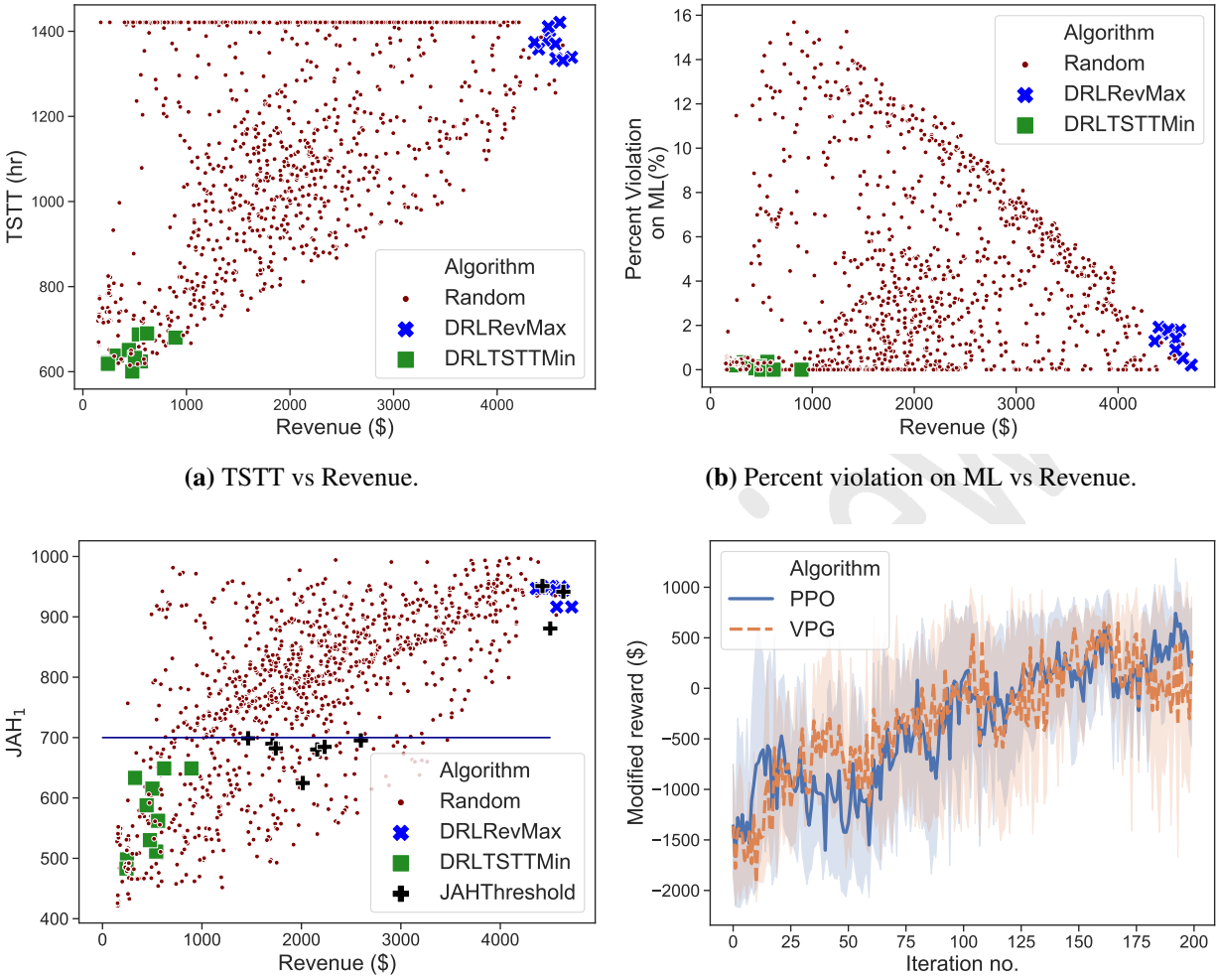
FIGURE 4 Plot of the average revenue with iteration over 5 random seeds for the three levels of observation for (a) VPG algorithm, and (b) PPO algorithm

We make following observations. First, the best toll profiles generated from Deep-RL algorithm are the best found among the other randomly generated profiles for the respective objectives. For the revenue maximization objective, toll profiles generated from Deep-RL algorithms have the highest revenue for all networks. For the TSTT minimization objective, toll profiles from Deep-RL algorithm have the lowest TSTT. This indicates that Deep-RL is able to learn best policies.

Second, we observe that for the LBJ network with multiple access points to the ML, several toll profiles can cause violation of the speed limit constraint. However, the toll profiles optimizing the revenue or TSTT generate % violation less than 2%. This is intuitive: for the revenue maximization objectives, a higher revenue is obtained only when more travelers use ML and the lane is kept congestion free. Similarly, for the TSTT minimization objective, low TSTT occurs when travelers spend less time in the network and exit the system sooner which is achieved when ML is ensured to be flowing at its capacity and does not become congested.

Last, similar to the trends in the literature, toll profiles generating high revenue also generate high values of TSTT. Similarly, tolls generating high revenue also have higher values of JAH_1 statistics, indicating the *jam-and-harvest* nature of revenue-maximizing tolls. To reduce the undesired JAH nature, we modify the reward definition. We simulate a policy and if at the end of an episode the JAH statistic is higher than a threshold, a high negative value is added to the reward to penalize such update. We test this technique, referred as JAHThreshold, to find tolls that maximize revenue such that JAH_1 statistic is less than a threshold value.

For the LBJ network, we apply the JAHThreshold technique with a threshold JAH_1 of 700 vehicles and add a reward value of $-\$3000$ to the final reward if at the end of simulation the JAH_1 statistic is higher than the threshold. Figure 5d shows the learning curve plotting the variation of modified reward with iterations. We observe that both VPG and PPO algorithms improve the modified reward with iterations, though it is hard to argue that they have converged. Learning is difficult in this case due to the same *credit assignment problem* where it is unclear will toll over an episode resulted in constraint violation. Figure 5c shows the plot for tolls obtained from JAH-threshold technique on the JAH_1-Revenue space. As observed, the reward penalization method is able to learn toll profiles with desired JAH value for 7 out of 10 random seeds. However, the learned toll profile is not the best found (that is there are toll profiles with JAH less than 700 but generating revenue higher than $\$2800$, which is the best found revenue), which is because



(a) TSTT vs Revenue.

(b) Percent violation on ML vs Revenue.

(c) JAH_1 vs Revenue.

(d) JAH-Threshold: variation of modified reward with iterations.

FIGURE 5 Plot of multiobjective for the LBJ network

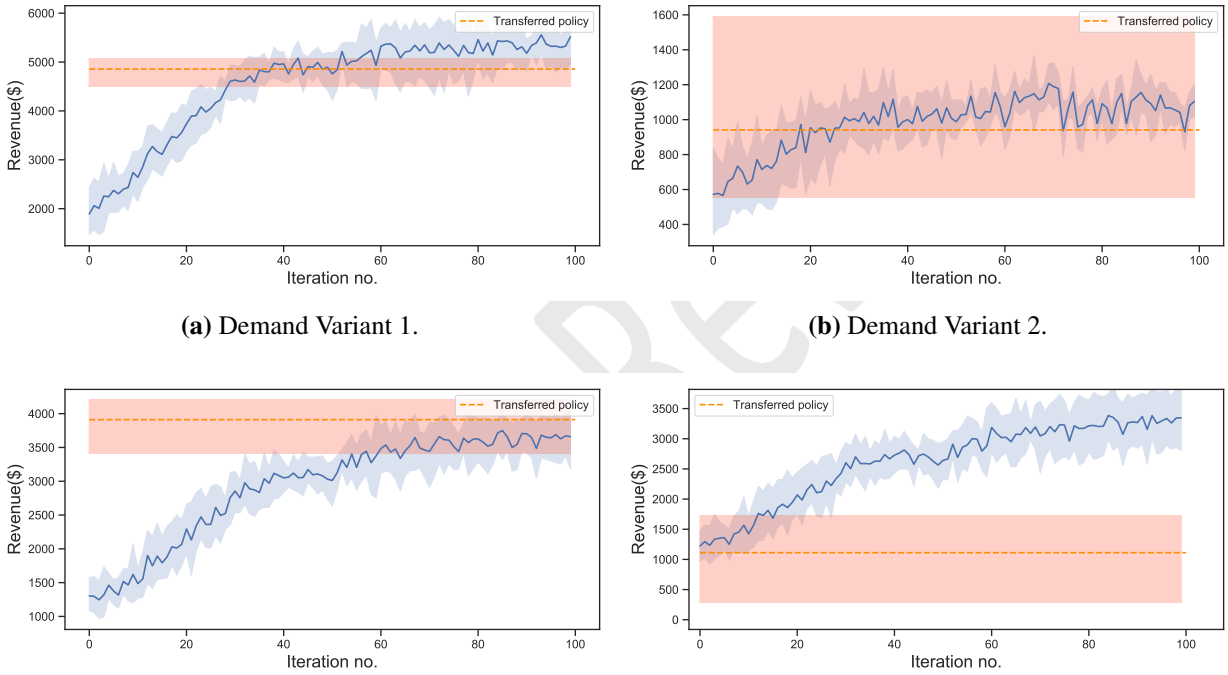
the modified reward did not converge (yet) after 200 iterations. Despite the lack of convergence, we conclude that the penalization method is a useful tool to model constraints on toll profiles. The success of penalization method depends on the random seed as that determines which local minimum the algorithm will converge to, so it is encouraged to simulate several random seeds to capture randomness in the model.

Transferability and Comparison against FC Heuristic

In this section we test how the policies trained on one set of inputs perform when transferred to new inputs without retraining for the new inputs. This analysis is useful for a toll operator who trains the algorithm in a simulation environment for certain assumptions of input. We consider the revenue-maximizing policy for the LBJ network and report results of transferability analysis for four cases. The first two cases consider new demand distributions (Variant 1 and Variant 2) shown

in Figure 2b. The third case considers a new VOT distribution (Variant 3) shown in Figure 2c. And, the last case transfers the policy trained using multiclass decision route model to a setting where driver lane choice is governed using a multiclass binary Logit model with scaling parameter value of 6 *(23)*. The observation space was kept the same to ensure that the learnt policy can be applied.

Figure 6 show the plots of variation of revenue with iterations while learning from scratch and the average revenue (and its full range of variation) obtained from the transferred policy for the new inputs. The average is reported over 100 runs of the transferred policy in new environments without retraining.



(a) Demand Variant 1.

(b) Demand Variant 2.

(c) VOT Variant 3.

(d) Stochastic lane choice.

FIGURE 6 Comparing learning from scratch with transfering the policy learnt on a different distribution

First, we observe that learning for the new input configurations "converges" within 100 iterations for all four cases with less than 10% variation in average revenue over last 50 iterations. Second, the average revenue of the transferred policy is found to be "close to optimal" for the first three cases: the average revenue from transferred policy is within $5 - 12\%$ of the average revenue at termination while learning from scratch. For the case 3, the transferred policy does even better than the policy learnt from scratch after 100 iterations of training. The observations from the first three cases suggest that even though the Deep-RL algorithms were not trained for the new inputs, they are able to learn characteristics of the congestion in the network and perform well (on an average) on the new inputs. However, for case 2 there is a lot of variance in the generated revenue from the transferred policy; this is because small changes in input tolls have a higher impact on generated revenue for demand Variant 2.

Third, contrary to the first three cases, the transfer of policy in case 4 did not work well: the

average revenue of transferred policy is 40% of the maximum revenue obtained. This is because the Logit model predicts different proportion of splits of travelers at a diverge and thus has a significant impact on the evolution of congestion. This finding suggests that the driver lane choice model should be appropriately selected for Deep-RL training.

Next, we compare the performance of the FC heuristic against Deep-RL algorithms. Table 2 shows the values of different statistics reported as four-tuple: (revenue, TSTT, JAH_1, %-violation) for both the revenue maximization and the TSTT minimization objectives for Deep-RL algorithms (we report the better objective value between VPG and PPO) and the FC heuristic. We highlight the value of the optimization objective in bold.

TABLE 2 Comparison of Deep-RL against the FC heuristic

Revenue maximization objective		
	Deep-RL	**Feedback Control**
SESE	(**$11999.80**, 2895.27 hr, 1109.00 veh, 0%)	(**$478.82**, 4693.09 hr, 583 veh, 0%)
DESE	(**$503.71**, 213.94 hr, 159.74 veh, 0%)	(**$467.44**, 283.19 hr, 162.99 veh, 0%)
LBJ	(**$4718.43**, 1338.86 hr, 916.42 veh, 0.21%))	(**$3767.04**, 1328.18 hr, 859.12 veh, 4.78%)
MoPaC	(**$18903.82**, 9658.53 hr, 3094.58 veh, 1.41%)	(**$3956.72**, 5151.57 hr, 1490.99 veh, 0%)
TSTT minimization objective		
	Deep-RL	**Feedback Control**
SESE	($11622.51, **2907.50 hr**, 1166.37 veh, 0%)	($240.91, **4071.62 hr**, 544.86 veh, 0%)
DESE	($258.15, **173.34 hr**, 120.69 veh, 0%)	($233.05, **172.11 hr**, 108.91 veh, 0%)
LBJ	($477.02, **600.36 hr**, 530.04 veh, 0.14%)	($463.49, **642.35 hr**, 557.70 veh, 0%)
MoPaC	($620.17, **4021.73 hr**, 1157.40 veh, 0.07%)	($603.89, **4008.97 hr**, 1132.52 veh, 0.09%)

We observe that Deep-RL does consistently well in generating toll profiles with higher revenues than the FC heuristic. The generated revenues from Deep-RL are 8%–2406% higher than the FC heuristic for different cases. For the TSTT minimization objective, no algorithm is clearly superior to the other, though Deep-RL algorithms perform relatively well. For DESE and MoPaC networks, the FC heuristic generates tolls with 0.32–0.71% lower TSTT than Deep-RL algorithms; however, the trend is reversed for the SESE and LBJ network, where Deep-RL algorithms generate tolls with 6.5–28.6% lower TSTT than the FC heuristic. Similar to the observations made earlier, the tolls maximizing the revenue also generate a high value of JAH_2 statistic and the tolls generating high revenue generate low TSTT (with an exception of SESE network). We note that the value of %-violation on the ML is less than 2% for the best-found toll profiles from Deep-RL algorithms.

We conclude that the FC heuristic is well suited for generating toll profiles with lower TSTT values and can serve as a good initial toll for training using Deep-RL algorithms for TSTT-

minimization objective. The feedback control heuristic has a computational advantage as it only requires one shot calculation which is easier to implement in real-time. Future work can be devoted in devising other heuristics that combine the optimization efficiency of Deep-RL algorithms and the computational efficiency of the FC heuristics (FW#5).

CONCLUSIONS

In this research, we developed a Deep-RL framework for dynamic pricing of express lanes with multiple access points. We showed that the Deep-RL algorithms can learn the best-found toll profiles for multiple objectives and for objectives with constraints. The average objective value converged within 200 iterations for the four networks tests. The number of sensors and sensor locations were found to have a little impact on the learning due to the spatial correlation of congestion pattern. We also conducted transferability tests and showed that policies trained using Deep-RL algorithm can be transferred to settings with new demand and VOT distributions without losing performance; however, if the lane choice model is changed the transferred policy performs poorly. This indicates that calibrating a lane choice model is critical for dynamic pricing. We also compared the performance of Deep-RL algorithms against the FC heuristic and found that it outperformed the heuristic for the revenue-maximization objective generating average revenue 8%-2406% higher than the heuristic. For the TSTT-minimization objective, differences were less significant. We recommend the use of VPG and PPO algorithms for finding optimal tolls. Even if the input distributions are unknown to a toll operator, training can be performed using an approximate distribution with large variance. The training can then be improved by using data from a given day and the modified policy can be applied from next day onwards.

In addition to the future work ideas discussed earlier, this research highlights other new research directions. Since the current macroscopic model for traffic flow does not capture lane choice interactions, efficient Deep-RL algorithms can be developed using microscopic simulation models. It is also relevant to test the transferability of algorithms trained on macroscopic scale to microscopic levels and vice versa. Next, in addition to loop detectors, other types of observations like speeds, toll-tag readings, and measurements using Lagrangian sensors like GPS devices on vehicles can be used to train optimal toll profiles. Last, the research can benefit from constrained policy optimization methods like in Achiam et al. *(27)* that enforce that the minimum speed limit constraint on ML is satisfied throughout the learning phase.

ACKNOWLEDGMENTS

Partial support for this research was provided by the North Central Texas Council of Governments, the Data-Supported Transportation Operations and Planning University Transportation Center, and the National Science Foundation Grants No. 1254921, 1562291, and 1826230. The authors are grateful for this support. The authors would also like to thank Natalia Ruiz-Juri and Tianxin Li at the Center for Transportation Research, Austin for their help in providing us the data for the MoPaC Express lanes.

AUTHOR CONTRIBUTION STATEMENT

The authors confirm contribution to the paper as follows: study conception and design: V. Pandey and S. D. Boyles; data collection: V. Pandey and E. Wang; analysis and interpretation of results: V. Pandey and S. D. Boyles; draft manuscript preparation: V. Pandey, E. Wang, and S.D. Boyles. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] TRB Managed Lanes Committee. Managed Lanes Project Database. https://managedlanes.wordpress.com/category/projects/, 2019. Last Accessed: June 20, 2019.

[2] LBJ. LBJ express FAQs. http://www.lbjtexpress.com/faq-page/t74n1302, 2016. Last Accessed: June 20, 2019.

[3] Mark W Burris and John F Brady. Unrevealed preferences: Unexpected traveler response to pricing on managed lanes. In *Proceedings of the 98th Annual meeting of Transportation Research Board*, pages 18–04294. TRB, 2018.

[4] Li Yang, Romesh Saigal, and Hao Zhou. Distance-based dynamic pricing strategy for managed toll lanes. *Transportation Research Record: Journal of the Transportation Research Board*, (2283):90–99, 2012.

[5] Zhen Tan and H Oliver Gao. Hybrid model predictive control based dynamic pricing of managed lanes with multiple accesses. *Transportation Research Part B: Methodological*, 112:113–131, 2018.

[6] Tomer Toledo, Omar Mansour, and Jack Haddad. Simulation-based optimization of HOT lane tolls. *Transportation Research Procedia*, 6:189–197, 2015.

[7] Feng Zhu and Satish V Ukkusuri. A reinforcement learning approach for distance-based dynamic tolling in the stochastic network environment. *Journal of Advanced Transportation*, 49(2):247–266, 2015.

[8] Venktesh Pandey and Stephen D Boyles. Multiagent reinforcement learning algorithm for distributed dynamic pricing of managed lanes. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2346–2351. IEEE, 2018.

[9] Venktesh Pandey and Stephen D Boyles. Dynamic pricing for managed lanes with multiple entrances and exits. *Transportation Research Part C: Emerging Technologies*, 96:304–320, 2018.

[10] Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys (CSUR)*, 50(3):34, 2017.

[11] Dongbin Zhao, Yujie Dai, and Zhen Zhang. Computational intelligence in urban traffic signal control: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):485–494, 2011.

[12] Lauren M Gardner, Hillel Bar-Gera, and Stephen D Boyles. Development and comparison of choice models and tolling schemes for high-occupancy/toll (HOT) facilities. *Transportation Research Part B: Methodological*, 55:142–153, 2013.

[13] Yingyan Lou, Yafeng Yin, and Jorge A Laval. Optimal dynamic pricing strategies for high-occupancy/toll lanes. *Transportation Research Part C: Emerging Technologies*, 19(1):64–74, 2011.

[14] Elena G Dorogush and Alex A Kurzhanskiy. Modeling toll lanes and dynamic pricing control. *arXiv:1505.00506*, 2015.

[15] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[16] Francois Belletti, Daniel Haziza, Gabriel Gomes, and Alexandre M Bayen. Expert level control of ramp metering based on multi-task deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[17] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465*, 2017.

[18] Wade Genders and Saiedeh Razavi. Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142*, 2016.

[19] Soheil Mohamad Alizadeh Shabestary and Baher Abdulhai. Deep learning vs. discrete reinforcement learning for adaptive traffic signal control. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 286–293. IEEE, 2018.

[20] Elise van der Pol. Deep reinforcement learning for coordination in traffic light control. *Master's thesis, University of Amsterdam*, 2016.

[21] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[22] Carlos F Daganzo. The cell transmission model, part II: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

[23] Venktesh Pandey and Stephen D Boyles. Comparing route choice models for managed lane networks with multiple entrances and exits. *Transportation Research Record*, 2019. URL https://doi.org/10.1177/0361198119848706.

[24] Yundi Zhang, Bilge Atasoy, and Moshe Ben-Akiva. Calibration and optimization for adaptive toll pricing. In *2018 97th Annual Meeting of Transportation Research Board*, pages 18–05863. TRB, 2018.

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[26] OpenAI. Welcome to Spinning Up in Deep RL– Spinning Up documentation. https://spinningup.openai.com/en/latest/index.html, 2019. Last Accessed: June 20, 2019.

[27] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.