In *Proceedings of the 32nd Conference on Artificial Intelligence* (AAAI 2018),
New Orleans, Lousiana, USA. February 2018

DyETC: Dynamic Electronic Toll Collection for Traffic Congestion Alleviation

Haipeng Chen[1,†], Bo An[1,†], Guni Sharon[2,‡], Josiah P. Hanna[2,§], Peter Stone[2,§], Chunyan Miao[1,†], Yeng Chai Soh[1,†]
[1] Nanyang Technological University
[2] University of Texas at Austin
[†] {chen0939,boan,ascymiao,ecsoh}@ntu.edu.sg
[‡] gunisharon@gmail.com
[§] {jphanna,pstone}@cs.utexas.edu

Abstract

To alleviate traffic congestion in urban areas, electronic toll collection (ETC) systems are deployed all over the world. Despite the merits, tolls are usually pre-determined and fixed from day to day, which fail to consider traffic dynamics and thus have limited regulation effect when traffic conditions are abnormal. In this paper, we propose a novel dynamic ETC (DyETC) scheme which adjusts tolls to traffic conditions in realtime. The DyETC problem is formulated as a Markov decision process (MDP), the solution of which is very challenging due to its 1) multi-dimensional state space, 2) multi-dimensional, continuous and bounded action space, and 3) time-dependent state and action values. Due to the complexity of the formulated MDP, existing methods cannot be applied to our problem. Therefore, we develop a novel algorithm, PG-β, which makes three improvements to traditional policy gradient method by proposing 1) time-dependent value and policy functions, 2) Beta distribution policy function and 3) state abstraction. Experimental results show that, compared with existing ETC schemes, DyETC increases traffic volume by around 8%, and reduces travel time by around 14.6% during rush hour. Considering the total traffic volume in a traffic network, this contributes to a substantial increase to social welfare.

Introduction

Nowadays, governments face a worsening problem of traffic congestion in urban areas. To alleviate road congestion, a number of approaches have been proposed, among which ETC has been reported to be effective in many countries and areas (e.g., Singapore (LTA 2017), Norway (AutoPASS 2017)). The tolls of different roads and time periods are different, so that the vehicles are indirectly regulated to travel through less congested roads with lower tolls. However, although current ETC schemes vary tolls at different time periods throughout a day, they are *pre-determined* and *fixed* at the same periods from day to day. A few dynamic road pricing schemes (Joksimovic et al. 2005; Lu, Mahmassani, and Zhou 2008; Zhang, Mahmassani, and Lu 2013) have been proposed in the transportation research community which consider the variations of traffic demands over time. However, these tolling schemes still assume that traffic demands are fixed and are known a priori, and thus are *static* in essence. In practice, traffic demands fluctuate and cannot be predicted accurately, especially when the traffic is abnormal (e.g., in case of traffic accidents, or city events). As a result, these static tolling schemes usually have limited regulation effect. One recent work of Sharon et al. (2017) proposes a dynamic tolling scheme called Δ-tolling, which assigns a toll to each road proportional to the difference between its current travel time and its free-flow travel time. However, Δ-tolling does not take a proactive approach towards changes in the demand side. Instead, D-tolling only reacts to such changes once they are detected. In contrast, we propose a novel dynamic tolling scheme which optimizes traffic over the long run, with the following three major contributions.

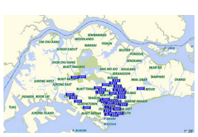
The first key contribution of this paper is a formal model of the DyETC problem. Since MDPs have various advantages in modelling long term planning problems with uncertainty, we formulate the DyETC problem as a discrete-time MDP. Though several existing methods have been proposed to solve the traffic assignment problem with MDP (Akamatsu 1996; Baillon and Cominetti 2008), our method is notably distinct from these works, in the sense that they consider the uncertainty of drivers' route choice behavior, while this work considers the uncertainty of traffic demand as well. The state of the formulated MDP is the number of vehicles on a road that are heading to certain destination, the action is the toll on each road, and the formulated MDP has 1) a multi-dimensional state space, 2) multi-dimensional, continuous and bounded action space, and 3) time dependent state and action values.

Due to the huge size of the MDP, it is very challenging to find its optimal policy. Traditional reinforcement learning algorithms based on tabular representations of the value and policy functions (e.g., *tabular Q-learning* (Watkins 1989), *prioritized sweeping* (Moore and Atkeson 1993), Monte Carlo Tree Search (MCTS) (Coulom 2006) and UCT (UCB applied to trees) (Kocsis and Szepesvári 2006)) cannot be applied to our problem due to the large scale state and action spaces. Value-based methods with function approximation (Precup, Sutton, and Dasgupta 2001; Maei et al. 2010; Nichols and Dracopoulos 2014; Mnih et al. 2015), which represent the state-action values (often referred to as "Q-values") with function approximators are also inefficient due to the complexity in selecting the optimal action in a continuous action space. While *policy gradient methods* (Williams 1992; Sutton et al. 1999; Peters and Schaal 2008; Schulman et al. 2015) work well in solving large scale MDPs with continuous action space, current policy gradient approaches usually focus on MDPs with unbounded action space. To handle bounded action spaces, Hausknecht and Stone (2016) propose three approaches (i.e., zeroing, squashing and inverting) on the gradients which force the parameters to preserve in their intended ranges. These manually enforced constraints may deteriorate the solution's optimality.

To solve the DyETC problem, we make our second key contribution by proposing an efficient solution algorithm, PG-β (Policy Gradient method with Beta distribution based policy functions). In the DyETC problem, the value of a state is time-dependent. We make our first improvement by extending the formulated MDP as time-dependent, adapting the traditional policy gradient method to maintain a separate value and policy function for each time period, and derive the update rule for the parameters. Moreover, to balance the tradeoff between "exploration" and "exploitation" for continuous action space, traditional methods either perform poorly in unbounded action space MDPs (Sutton et al. 1999), or manually enforce the parameters to guarantee the bounded action space (Hausknecht and Stone 2016). To overcome this deficiency, we make the second improvement by proposing a novel form of policy function, which is based on the Beta probability distribution, and deriving its corresponding update rules. Last, to further improve scalability of PG-β, we exploit the structure of the formulated DyETC problem, and propose an abstraction over the state space, which significantly reduces the scale of the state space, while maintaining near-optimality.

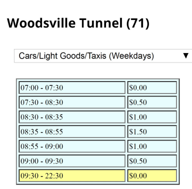
Third, we conduct extensive experimental evaluations to compare our proposed method with existing policy gradient methods as well as current tolling schemes. The results demonstrate that PG-β performs significantly better than state-of-the-art policy gradient methods and current tolling schemes. Performed in a road network of Singapore Central Region, DyETC increases the traffic volume by around 8%, and reduces the total travel time by around 14.6%.

Motivation Scenario



(a) ETC system

(b) ETC gantry



Woodsville Tunnel (71)

Cars/Light Goods/Taxis (Weekdays)

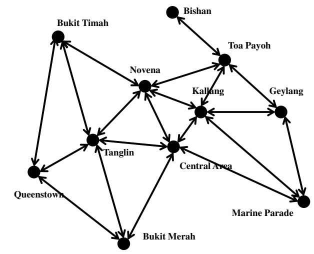
07:00 - 07:30	$0.00
07:30 - 08:30	$0.50
08:30 - 08:35	$1.00
08:35 - 08:55	$1.50
08:55 - 09:00	$1.00
09:00 - 09:30	$0.50
09:30 - 22:30	$0.00

(c) ETC rates

Figure 1: ETC in Singapore

Since 1998, an ETC system has been deployed in Singapore to alleviate its traffic congestion, especially in the central region (Figure 1(a)). In an ETC system, vehicles are

(a) Singapore Central Region



(b) Abstracted road network

Figure 2: Road network of Singapore Central Region

charged when they pass through ETC gantries (Figure 1(b)) and use priced roads during peak hours. Typically, ETC rates vary from different roads and time periods, which encourages vehicles to change their travel routes in order to alleviate traffic congestion. As shown in Figure 1(c), while an ETC gantry charges different ETC rates at different time periods, the rates are fixed during certain time periods from Monday to Friday. This framework, which is demonstrated to have significantly improved Singapore's traffic, still has two major shortcomings. *First, the current ETC system is determined based on the historical traffic flows, which fails to adjust to the uncertainty of traffic demand.* In reality, the real-time traffic demand could fluctuate. For example, during morning rush hour, we may know the average traffic demand based on history, but the exact demand cannot be precisely predicted. As a result, if we set tolls merely according to average traffic demand but the real demand is not as severe as expected, few vehicles will go through this road even if the congestion is not severe since they are scared off by the high tolls. Under such cases, the current ETC system will have less positive or even negative effect on the traffic condition. *Second, current ETC rating systems fail to precisely respond to the extent to which the traffic is congested.* As we can see from Figure 1(c), the interval of the current tolling scheme is 0.5. For example, the tolls are either 0 or 0.5, while the optimal toll might be somewhere between these two rates. In such cases, the current tolling scheme can hardly reveal and react to the accurate congestion level. To further alleviate traffic congestion in urban areas, we propose a novel dynamic ETC scheme which is 1) *fully dynamic* and 2) *finer-grained*.[1]

Formulation of the DyETC Problem

In this section, we first introduce the dynamic ETC system, and then formulate the DyETC problem as an MDP.

Dynamic ETC System

The urban city area can be abstracted as a directed *road network* $G = (E, Z, O)$, where E is the set of roads, Z is the

[1]Such dynamic toll information can be displayed on telescreens along roads which can be easily accessed by drivers. Moreover, with the introduction of intelligent agent-based vehicles and even autonomous vehicles (e.g., in Singapore (LTA 2016)), autonomous agents are able to aid the drivers in deciding the optimal travel routes under the proposed DyETC scheme.

set of zones and O is the set of origin-destination (OD) pairs. Take Singapore Central Region as an example (Figure 2), there are altogether 11 zones and 40 abstracted roads.[2] The decision time horizon H (usually the length of rush hour) is discretized into several intervals, with a length of τ (e.g., 10 minutes) for each interval. For time period $t = 0, 1, \ldots, H$, we denote an OD pair as a tuple $\langle z_i, z_j, q_{i,j}^t, P_{i,j} \rangle$, where zone z_i is the origin, zone z_j is the destination, $q_{i,j}^t$ denotes traffic demand during time t, and $P_{i,j}$ denotes the set of all possible paths from z_i to z_j which do not contain a cycle. Different from most previous works which assume that the OD traffic demand $q_{i,j}^t$ for a certain time period t is fixed and known *a priori*, we consider dynamic OD travel demand, where the travel demand of an OD pair follows a probability distribution function (PDF) $q_{i,j}^t \sim f(q_{i,j}^t)$.

We follow the commonly used *travel time model* (BPR 1964) to define the travel time on a road e at t:

$$T_e^t = T_e^0[1 + A(s_e^t/C_e)^B]. \tag{1}$$

s_e^t is the number of vehicles on road e. C_e and T_e^0 are road-specific constants, where T_e^0 is interpreted as the free-flow travel time, and C_e is the capacity of the road. A and B are constants which quantify the extent to which congestion influences travel time. Consequently, average travel speed is $\frac{L_e}{T_e^t} = \frac{L_e}{T_e^0[1+A(s_e^t/C_e)^B]}$, where L_e is the length of road e. At time t, the travel cost of a path $p \in P_{i,j}$, which we denote as $c_{i,j,p}$, consists of both time and monetary costs:

$$c_{i,j,p}^t = \sum\nolimits_{e \in p} (a_e^t + \omega T_e^t), \tag{2}$$

where a_e^t is the toll imposed on road e, and ω is a constant which reveals the value of time. To make the analysis tractable, we assume that all vehicles have the same value of time. Given the current traffic condition (i.e., number of vehicles on each road) and tolls, each vehicle will select a path $p \in P_{i,j}$ leading to its destination, which aggregately forms a *traffic equilibrium*. To describe this traffic equilibrium, we adopt a widely-used *stochastic user equilibrium* (SUE) model (Lo and Szeto 2002; Lo, Yip, and Wan 2003; Huang and Li 2007), where the portion of traffic demand $x_{i,j,p}^t$ travelling with path $p \in P_{i,j}$ is

$$x_{i,j,p}^t = \frac{\exp\{-\omega' c_{i,j,p}^t\}}{\sum_{p' \in P_{i,j}} \exp\{-\omega' c_{i,j,p'}^t\}}. \tag{3}$$

ω' is a constant measuring vehicles' sensitivity to travel cost.

An MDP Formulation

In general, the government sets the tolls for the current time period and announces them to the vehicles, while each vehicle individually selects paths according to the total travel cost, and the aggregate choice of all the vehicles is assumed to follow the discrete choice model in Eq.(3). We formulate the DyETC problem as a discrete time MDP, due

[2]Each pair of two adjacent zones has two directed roads. If there are multiple roads from one zone to another, they could be treated as one abstracted road, where the capacity of the abstracted road is a sum of these roads, and the length of the abstracted road is an average of these roads.

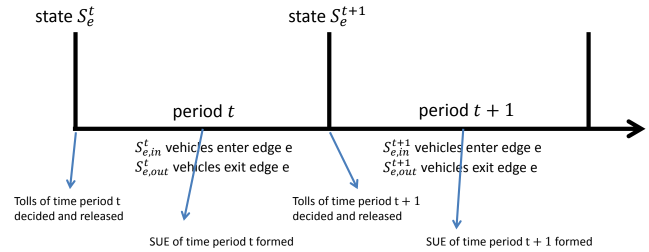


Figure 3: Event timeline of two subsequent time periods

to its advantages in modeling sequential planning problems.

State & action. At the beginning of time period t, the *state* is defined as the number of vehicles $s_{e,j}^t$ on each road e that are going to destination z_j. $\mathbf{s}_e^t = \langle s_{e,j}^t \rangle$ is the state vector of a road e, and $\mathbf{s}^t = \langle \mathbf{s}_e^t \rangle$ is the state matrix of the road network G. At time t, the government's *action* is to set the tolls $\mathbf{a}^t = \langle a_e^t \rangle, e \in E'$, where $E' \subseteq E$ is the subset of roads which have ETC gantries.

Since both the traffic condition and tolls change over time, a vehicle has an incentive to change its path once it reaches the end of a road. The path readjustment does not depend on the past decisions of the vehicle, but only depends on its destination. Therefore, for a vehicle that arrives at the end z_i of a road e, we treat it as a vehicle from the new origin z_i, while maintaining its destination z_j. To distinguish these vehicles with those that really use z_i as the origin, we define:

Definition 1. *At time period t, the* ***primary OD demand*** $q_{i,j}^t$ *from z_i to z_j is the number of vehicles that originate from z_i at time period t; while the* ***secondary OD demand*** $\bar{q}_{i,j}^t$ *is the number of vehicles that come from z_i's neighbouring roads during time period $t-1$ that are heading for destination z_j.*

We refer to Figure 3 as an illustration of the event timeline for two subsequent time periods. At the beginning of time period t, tolls are decided based on the state $\mathbf{s}^t$ of the current time period. After the tolls are announced to the vehicles, the vehicles will react to the tolls and traffic conditions and a SUE will gradually form during time period t. In practice, it usually takes time to form an SUE and it keeps evolving over time. Before the SUE is formed, the number of vehicles on a road is normally larger (or smaller) than that in the SUE, while after the SUE, the number is usually smaller (or larger). To make the analysis tractable, we use the number of vehicles when the SUE is formed to approximate the average number of vehicles on a road during time period t.

State transition. After the SUE is formed, the state of the next time period can be derived. At the beginning of time $t+1$, the number of vehicles on road e is determined by the number of vehicles that 1) stay on road e, 2) exit road e, and 3) enter road e during time t. Formally,

$$s_{e,j}^{t+1} = s_{e,j}^t - s_{e,j,out}^t + s_{e,j,in}^t. \tag{4}$$

We make a mild and intuitive assumption that the number of vehicles $s_{e,j,out}^t$ that exit a road e is proportional to the average travel speed during time period t. Thus,

$$s_{e,j,out}^t = s_{e,j}^t \cdot \frac{v_e^t \cdot \tau}{L_e} = \frac{s_{e,j}^t \tau}{T_e^0[1 + A(s_e^t/C_e)^B]}, \tag{5}$$

where L_e is the length of road e.

We now derive the last term $s^t_{e,j,in}$ in Eq.(4). Recall that the total demand of an OD pair is the sum of primary and secondary OD demand. The secondary demand of an OD pair from z_i to z_j is

$$\bar{q}^t_{i,j} = \sum\nolimits_{e'^+=z_i} s^t_{e',j,out}, \tag{6}$$

where e^+ is the ending point of road e (correspondingly, we use e^- to denote the starting point of e). Note that during time t, for an OD traffic demand from z_i to z_j to be counted as a component of $s^t_{e,j,in}$, two conditions must be satisfied. First, z_i should be the starting point of e, i.e., $z_i = e^-$. Second, at least one of the paths from z_i to z_j should contain road e, i.e., $e \in p \in P_{i,j}$. Thus, we have:

$$s^t_{e,j,in} = \sum\nolimits_{z_i=e^- \cap e \in p \in P_{i,j}} (q^t_{i,j} + \bar{q}^t_{i,j}) \cdot x^t_{i,j,p}. \tag{7}$$

Combining Eqs.(4)-(7), there is:

$$\begin{aligned} s^{t+1}_{e,j} = s^t_{e,j} - \frac{s^t_{e,j}\tau}{T^0_e[1 + A(s^t_e/C_e)^B]} + \\ \sum\nolimits_{z_i=e^- \cap e \in p \in P_{i,j}} (q^t_{i,j} + \sum\nolimits_{e'^+=v_i} s^t_{e',j,out}) \cdot x^t_{i,j,p}. \end{aligned} \tag{8}$$

Value function & Policy. From the perspective of the government, a good traffic condition means that, during the planning time horizon H, the total traffic volume (i.e., the number of vehicles that reach their destinations) is maximized.[3] Formally, we define the *immediate reward function* $R^t(\mathbf{s}^t, \mathbf{a}^t)$ as the number of vehicles that arrive at their destinations during time t:

$$R^t(\mathbf{s}^t) = \sum\nolimits_{e \in E} \sum\nolimits_{z_j=e^+} \frac{s^t_{e,j}\tau}{T^0_e[1 + A(s^t_e/C_e)^B]}. \tag{9}$$

Note that we simplify the notation as $R^t(\mathbf{s}^t)$ since it does not depend on $\mathbf{a}^t$. The long term value function $v^t(\mathbf{s}^t)$ is the sum of rewards from t to $t + H$:

$$v^t(\mathbf{s}^t) = \sum\nolimits_{t'=t}^{t+H} \gamma^{t'-t} R^{t'}(\mathbf{s}^{t'}), \tag{10}$$

where γ is a discount factor. At time t, a *policy* $\pi^t(\mathbf{a}^t|\mathbf{s}^t)$ is a function which specifies the conditional probability of taking an action $\mathbf{a}^t$, given a certain state $\mathbf{s}^t$. The optimal policy maximizes the value function in Eq.(10):

$$\pi^{t,*}(\mathbf{a}^t|\mathbf{s}^t) = \arg\max\nolimits_{\pi^t} v^t(\mathbf{s}^t). \tag{11}$$

Solution Algorithm: PG-β

It is very challenging to find the optimal policy function for the formulated DyETC problem due to three reasons. First, the state space is multi-dimensional w.r.t. the number of roads and destinations. Second, the action space is also multi-dimensional w.r.t. the number of ETC gantries. Moreover, the action space is bounded and continuous. Last, both the state and action values are dependent on the specific time periods. As a result, it is intractable to find the optimal policy function by simply going through all combinations of state-action pairs, which would be on an astronomical order.

Although numerous reinforcement learning algorithms have been proposed to solve MDPs, they cannot be directly applied to our problem due to the complexities presented above. While the policy gradient methods (Williams 1992; Sutton et al. 1999) have shown promise in solving large scale MDPs with continuous action spaces, current policy gradient methods usually focus on MDPs with unbounded action spaces. In the following, we present our solution algorithm, PG-β (Policy Gradient method with Beta distribution based and time-dependent policy functions), with novel improvements to typical applications of policy gradient methods.

General Framework of PG-β

Algorithm 1: PG-β

1 Initialize $\boldsymbol{\vartheta}^t \leftarrow \boldsymbol{\vartheta}_0, \boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}_0, \forall t = 0, 1, \ldots, H$;
2 **repeat**
3   Generate an episode $\mathbf{s}^0, \mathbf{a}^0, R^0, \ldots, \mathbf{s}^H, \mathbf{a}^H, R^H$;
4   **for** $t = 0, \ldots, H$ **do**
5     $Q^t \leftarrow \sum_{t'=t}^{H} R^\tau$;
6     $\delta \leftarrow Q^t - \hat{v}(\mathbf{s}^t, \mathbf{a}^t, \boldsymbol{\vartheta}^t)$
7     $\boldsymbol{\vartheta}^t \leftarrow \boldsymbol{\vartheta}^t + \beta\delta\nabla_{\boldsymbol{\vartheta}^t}\hat{v}(\mathbf{s}^t, \boldsymbol{\vartheta}^t)$;
8     $\boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}^t + \beta'\delta\nabla_{\boldsymbol{\theta}^t} \log \pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)$
9 **until** *#episodes*$= M$
10 **return** $\boldsymbol{\theta}^t$, $\forall t = 0, 1, \ldots, H$;

The idea of PG-β is to approximate the policy function with a parameterized function, and update the parameter with stochastic gradient descent method. More specifically, PG-β incorporates an *actor-critic* architecture, where the "actor" is the learned policy function which performs action selection, and the "critic" refers to the learned value function measuring the performance of the current policy function.

In Algorithm 1, the input of PG-β includes the planning horizon H, the state transition function Eq.(8), a set of parameterized value functions $v^t(\mathbf{s}, \boldsymbol{\vartheta}^t)$ w.r.t. parameter $\boldsymbol{\vartheta}^t$ ($\forall t = 0, \ldots, H$), and a set of parameterized policy functions $\pi^t(\mathbf{a}|\mathbf{s}, \boldsymbol{\theta}^t)$ w.r.t. parameter $\boldsymbol{\theta}^t$ ($\forall t = 0, \ldots, H$). The algorithm starts with an initialization of the parameters $\boldsymbol{\vartheta}^t$ and $\boldsymbol{\theta}^t$. It then enters the repeat loop (Lines 2-9). In the repeat loop, it first simulates an *episode*, which contains a series of state-action pairs in the planning horizon H. The states are generated following the state transition function Eq.(8), the actions are selected following the policy function $\pi(\mathbf{a}|\mathbf{s}, \boldsymbol{\theta})$, while the immediate rewards are computed with Eq.(9). After an episode is simulated, the algorithm then updates the parameters at each time period $t = 0, \ldots, H$ (Lines 4-9) with stochastic gradient descent method. Q^t denotes the sum of rewards from time t to H obtained from the simulated episode, which reveals the "real" value obtained by the current policy, while $\hat{v}^t(\mathbf{s}^t, \boldsymbol{\vartheta}^t)$ is the "estimated" sum of rewards approximated by the parameterized value function $v^t(\mathbf{s}^t, \boldsymbol{\vartheta}^t)$. Consequently, δ

[3]DyETC can be extended to optimize other objective functions, total travel time among them, by changing the reward function appropriately. We leave such extensions to future work.

denotes the difference of these two terms. In Lines 7-8, PG-β updates the parameters by our derived update rule which is to be introduced in the two subsequent subsections. This process iterates until the number of episodes reaches a predefined large number M (e.g., 100,000).

Policy Gradient for Time-Dependent MDP

A fundamental property of traditional MDPs is that the value of a state does not change over time. However, such property does not exist in our problem. Intuitively, with the same number of vehicles on the road network, the number of vehicles that reach the destinations (i.e., the objective) still depends on the OD demand of a specific time period as well as future periods. Consequently, the value of an action is also dependent on the specific time period. This class of MDPs is called *finite horizon MDPs* (FHMDPs). For FHMDPs, we need to maintain and update a value function $v^t(\mathbf{s}, \boldsymbol{\vartheta}^t)$ and a policy function $\pi^t(\mathbf{a}|\mathbf{s}, \boldsymbol{\theta}^t)$ for each time step $t = 0, 1, \ldots, H$, as shown in Algorithm 1. The following theorem ensures that the update of $\boldsymbol{\theta}^t$ improves action selection in the FHMDP.[4]

Theorem 1. *The gradient function of policy gradient method on FHMDPs is*

$$\nabla_{\boldsymbol{\theta}^t} v_\pi(\mathbf{s}) = Q^t(\mathbf{s}^t, \mathbf{a}^t)\nabla_{\boldsymbol{\theta}^t} \log \pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t), \tag{12}$$

where $Q^t(\mathbf{s}^t, \mathbf{a}^t)$ is the action value of $\mathbf{a}^t$ given state $\mathbf{s}^t$.

Proof. We first write the state value as the expected sum of action values, i.e., $v^t(\mathbf{s}^t) = \sum_{\mathbf{a}^t} \pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)Q^t(\mathbf{s}^t, \mathbf{a}^t)$, where $Q^t(\mathbf{s}^t, \mathbf{a}^t) = \sum_{\mathbf{s}^{t+1}} P(\mathbf{s}^{t+1}, \mathbf{a}^t, \mathbf{s}^t)v^{t+1}(\mathbf{s}^{t+1})$. Since both the transition probability function $P(\mathbf{s}^{t+1}, \mathbf{a}^t, \mathbf{s}^t)$ and the value function $v^{t+1}(\mathbf{s}^{t+1})$ of time $t+1$ do not depend on $\boldsymbol{\theta}^t$, $Q^t(\mathbf{s}^t, \mathbf{a}^t)$ is also independent on $\boldsymbol{\theta}^t$. Thus, we have

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}^t} v^t(\mathbf{s}^t) &= \sum\nolimits_{\mathbf{a}^t} Q^t(\mathbf{s}^t, \mathbf{a}^t)\nabla_{\boldsymbol{\theta}^t}\pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)\\
&= \sum\nolimits_{\mathbf{a}^t} \pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)Q^t(\mathbf{s}^t, \mathbf{a}^t)\frac{\nabla_{\boldsymbol{\theta}^t}\pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)}{\pi^t(\mathbf{s}^t, \mathbf{a}^t)}\\
&\quad \text{(multiplying and dividing by } \pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t))\\
&= E[Q^t_\pi(\mathbf{s}^t, \mathbf{a}^t)\frac{\nabla_{\boldsymbol{\theta}^t}\pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)}{\pi^t(\mathbf{a}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)}].
\end{aligned}$$

For stochastic gradient, a sampled action $\mathbf{A}^t$ is used to replace the expectation, i.e.,

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}^t} v^t(\mathbf{s}^t) &= Q^t_\pi(\mathbf{s}^t, \mathbf{A}^t)\frac{\nabla_{\boldsymbol{\theta}^t}\pi^t(\mathbf{A}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)}{\pi^t(\mathbf{A}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)}\\
&= Q^t(\mathbf{s}^t, \mathbf{A}^t)\nabla_{\boldsymbol{\theta}^t}\log\pi^t(\mathbf{A}^t|\mathbf{s}^t, \boldsymbol{\theta}^t)
\end{aligned}$$

□

[4]For FHMDPs, a natural alternative is to incorporate the time as one extra dimension of state. However, experimental results (see Figure 4(a)) show that this alternative does not work well using polynomial basis functions where states are not interrelated. Other alternatives might be to utilize an intrinsic basis function, or use deep neural networks to represent the policy functions. However, such intrinsic functions are very hard to find (perhaps it does not exist), while the time consumed in training an effective neural network is usually way longer than training linear policy functions.

A Novel Policy Function Based on the Beta PDF

An important challenge of the policy gradient method is to balance the "exploitation" of the optimal action generated from the policy function and the "exploration" of the action space to ensure that the policy function is not biased. For MDPs with *continuous* action space, a Normal PDF has been used in recent works (Sutton and Barto 2011) as the policy function. While Normal PDF works well in cases where the action space is unbounded, it is not suitable for MDPs which have bounded action spaces, since the action generated by the Normal PDF policy function would possibly become infeasible. In practice, tolls are restricted within a certain interval $[0, a_{max}]$ (e.g., in Singapore, tolls are within 6 Singapore dollars). A straightforward adaptation is to project the generated action to the feasible action space whenever it is infeasible. However, experimental results (will be presented later in Figure 4(a)) show that, even with such adaptation, Normal PDF policy function performs poorly in solving MDPs with bounded action spaces.

To adapt policy gradient methods to bounded action space, we propose a new form of policy function, which is derived from Beta PDF $f(x)$:

$$f(x, \lambda, \nu) = \frac{x^{\lambda-1}(1-x)^{\xi-1}}{B(\lambda, \xi)}, \tag{13}$$

where the *Beta function*, $B(\lambda, \xi) = \int_0^1 t^{\lambda-1}(1-t)^{\xi-1}dt$ is a normalization constant, and the variable $x \in [0, 1]$ is bounded and continuous. For each road e, let $x_e = a_e/a_{max}$, then the policy function is denoted as

$$\pi_e(a_e = x_e a_{max}|\mathbf{s}, \lambda_e, \xi_e) = \frac{x_e^{\lambda_e-1}(1-x_e)^{\xi_e-1}}{B(\lambda_e, \xi_e)}, \tag{14}$$

where $\lambda_e = \lambda_e(\boldsymbol{\phi}(\mathbf{s}), \boldsymbol{\theta}^\lambda)$ and $\xi_e = \xi_e(\boldsymbol{\phi}(\mathbf{s}), \boldsymbol{\theta}^\xi)$ are parameterized functions, and $\boldsymbol{\theta}^\lambda$ and $\boldsymbol{\theta}^\xi$ are parameters to be approximated. For example, in the commonly utilized *linear function approximation*, each λ_e and ξ_e is represented by a linear function of the parameters $\boldsymbol{\theta}^\lambda$ and $\boldsymbol{\theta}^\xi$: $\lambda_e = \boldsymbol{\theta}^\lambda \cdot \boldsymbol{\phi}(\mathbf{s}), \xi_e = \boldsymbol{\theta}^\xi \cdot \boldsymbol{\phi}(\mathbf{s})$.

Proposition 1. *Using Beta PDF policy function, the update rule for a parameter $\theta^\lambda_{e,e',j,i}$ (associated with road e) is*

$$\theta^\lambda_{e,e',j,i} \leftarrow \theta^\lambda_{e,e',j,i} + \beta'\delta[\ln(x_e) - \Psi(\lambda_e) + \Psi(\lambda_e + \xi_e)]\frac{\partial\lambda_e}{\partial\theta^\lambda_{e,e',j,i}},$$

$$\theta^\xi_{e,e',j,i} \leftarrow \theta^\xi_{e,e',j,i} + \beta''\delta[\ln(1-x_e) - \Psi(\xi_e) + \Psi(\lambda_e + \xi_e)]\frac{\partial\xi_e}{\partial\theta^\xi_{e,e',j,i}},$$

where β' and β'' are learning rates for θ^λ and θ^ξ, respectively, $\Psi(\cdot)$ is the diagamma function. e', j and i respectively correspond to a road, a destination and a dimension in the basis functions.

Proof. For ease of notation, we discard the superscript t. In Eq.(12), for edge e and dimension i of $\boldsymbol{\theta}^\lambda$, by substituting

$\pi(\mathbf{a}|\mathbf{s},\boldsymbol{\theta})$ with Eq.(14), we have

$$\frac{\partial v(\mathbf{s})}{\partial \theta^{\lambda}_{e,e',j,i}} = \frac{Q(\mathbf{s},\mathbf{a})\partial \pi_e(a_e|\mathbf{s},\theta^{\lambda}_{e,e',j,i})}{\pi_e(a_e|\mathbf{s},\theta^{\lambda}_{e,e',j,i})\partial \theta^{\lambda}_{e,e',j,i}}$$

$$= \frac{Q(\mathbf{s},\mathbf{a})}{\pi_e(a_e|\mathbf{s},\theta^{\lambda}_{e,e',j,i})} \frac{\partial \pi_e(a_e|\mathbf{s},\theta^{\lambda}_{e,e',j,i})}{\partial \lambda_e} \frac{\partial \lambda_e}{\partial \theta^{\lambda}_{e,e',j,i}}$$

$$= \frac{Q(\mathbf{s},\mathbf{a})}{\pi_e(a_e|\mathbf{s},\theta^{\lambda}_{e,e',j,i})} \frac{\partial \xi_e}{\partial \theta^{\xi}_{e,e',j,i}} [\frac{x_e^{\lambda_e-1}\ln x_e(1-x_e)^{\xi_e-1}}{B(\lambda_e,\xi_e)} - \frac{x_e^{\lambda_e-1}(1-x_e)^{\xi_e-1}\partial B(\lambda_e,\xi_e)/\partial\lambda_e}{B^2(\lambda_e,\xi_e)}]$$

$$= \frac{Q(\mathbf{s},\mathbf{a})}{\pi_e(a_e|\mathbf{s},\theta^{\lambda}_{e,e',j,i})} \frac{\partial \xi_e}{\partial \theta^{\xi}_{e,e',j,i}} \frac{x_e^{\lambda_e-1}(1-x_e)^{\xi_e-1}[\ln x_e - \Psi(\lambda_e) + \Psi(\lambda_e+\xi_e)]}{B(\lambda_e,\xi_e)}$$

$$= Q(\mathbf{s},\mathbf{a})[\ln x_e - \Psi(\lambda_e) + \Psi(\lambda_e+\xi_e)]\frac{\partial \xi_e}{\partial \theta^{\xi}_{e,e',j,i}}$$

By replacing $Q(\mathbf{s},\mathbf{a})$ with δ, we obtain the update rule for $\theta^{\lambda}_{e,e',j,i}$. Similarly, we derive the update rule for $\theta^{\xi}_{e,e',j,i}$. $\square$

State Abstraction

As discussed above, there are two corresponding parameters $\theta^{\lambda}_{e,e',j,i}$ and $\theta^{\xi}_{e,e',j,i}$ for each edge e, each edge e', each destination j and each dimension i in the basis functions. As a result, the total number of parameters in the policy function is of order $\propto |E|^2|Z|Hd$, where $|E|$, $|Z|$, H and d are respectively the number of edges, number of vertices (destinations), length of time horizon and number of dimensions in the basis functions. In this case, when a road network grows too large, it becomes rather time consuming for PG-β to learn an effective tolling policy. Moreover, higher dimension states may lead to over-fitting. To handle these two issues, we conjecture:

Conjecture 1. *The vehicles on a same edge that are going to different destinations have almost equal effects on tolls.*

This conjecture means that, for the tolls on road e, the parameters (weights) associated to $s_{e',j}$ and $s_{e',i}$ are equal:

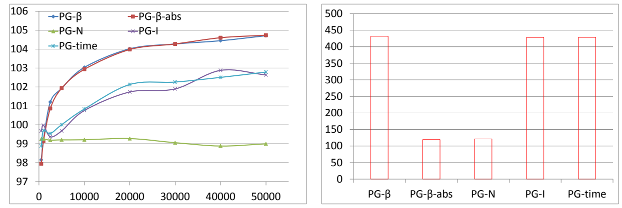
$$\theta^{\lambda}_{e,e',j,i} = \theta^{\lambda}_{e,e',j',i} = \theta^{\lambda}_{e,e',i}, \forall j,j' \tag{15}$$

$$\theta^{\xi}_{e,e',j,i} = \theta^{\xi}_{e,e',j',i} = \theta^{\xi}_{e,e',i}, \forall j,j' \tag{16}$$

The supporting evidence of this conjecture will be shown through experimental evaluations.

Experimental Evaluation

In this section, we conduct experiments on both simulated settings and a real-world road network in Singapore Central Region to evaluate our proposed DyETC scheme and its solution algorithm PG-β. All algorithms are implemented using Java, all computations are performed on a 64-bit machine with 16 GB RAM and a quad-core Intel i7-4770 3.4 GHz processor.



(a) Learning curve (b) Per episode runtime

Figure 4: Performance of different policy gradient methods

Evaluation on Synthetic Data

We first conduct experiments on synthetic data. For policy gradient methods, we first obtain the policy function with offline training, and then use the trained policy to evaluate their performance. Unless otherwise specified, the parameters of this subsection are set as follows.

Learning-related parameters. The learning rates of the value function and policy function are hand-tuned as 10^{-7} and 10^{-10}, respectively. The discount factor γ is set as 1, which assigns same weights to rewards of different time periods in the finite time horizon H. The number of episodes for training is 50,000 and the number of episodes for validation is 10,000.

Road network-related parameters. The number of zones in the simulation is set as $|Z| = 5$, and all zones can be destinations, i.e., $|Z'| = |Z|$. The number of roads is set to 14, and all roads have ETC gantries, i.e., $|E'| = |E| = 14$. The lengths of roads are randomized within $[4, 10]$ km, which is the usual length range between zones of a city. Capacity of a road is set as 50 vehicles per kilometer per lane. This amount is obtained from an empirical study of Singapore roads (Olszewski 2000). Free flow travel speed $\frac{T_e^0}{L_e} = 0.5$ km/min. The parameters in the travel time model Eq.(1) is set as $A = 0.15, B = 4$ according to (BPR 1964).

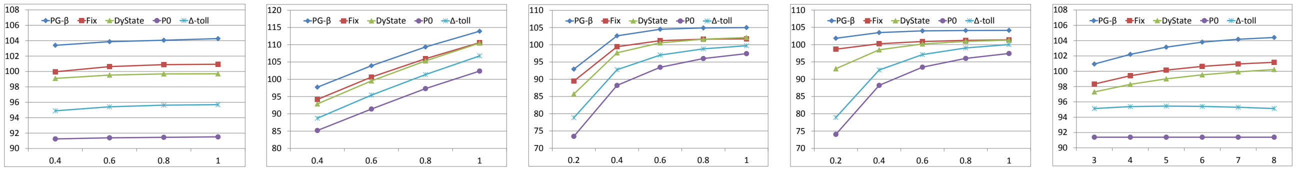
Demand-related parameters. OD demand is simulated as a step function of time, where demand at time $t = 0$ is the lowest, and gradually grows to peak demand in the middle of the planning time horizon, and decreases again to a lower level. The peak demand for each OD pair is randomized within $[8, 12]$ vehicles per minute, which is a reasonable amount. The OD demand at $t = 0$ (which is usually the beginning of rush hour) is set as 60% of the peak demand. Initial state is randomized within $[0.5, 0.7]$ of the capacity of a road.

Toll-related parameters. Maximum toll $a_{max} = 6$. This value is obtained from the current toll scheme in Singapore. Planning horizon $H = 6$. Length of a time period $\tau = 10$ $mins$. Passenger cost-sensitivity level $\omega' = 0.5$, and value of time $\omega = 0.5$. We will evaluate other values for these two terms.

Comparison of different policy gradient methods.

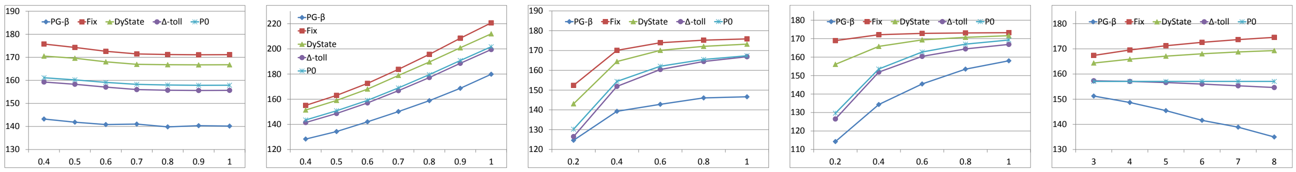
We compare the learning curve and solution quality of PG-β with the following policy gradient algorithms.

1. PG-N: policy gradient (PG) with Normal distribution based policy function.



(a) Initial state (b) Initial demand (c) Cost sensitivity (d) Value of time (e) Maximum price

Figure 5: Traffic volume (in thousands) of existing tolling schemes under various traffic conditions



(a) Initial state (b) Initial demand (c) Cost sensitivity (d) Value of time (e) Maximum price

Figure 6: Total travel time (in thousands) of existing tolling schemes under various traffic conditions

2. PG-I: PG with time-independent policy function.
3. PG-time: PG-β where time is incorporated into state.
4. PG-β-abs: PG-β with state abstraction.

Figure 4(a) shows the learning curve of different policy gradient methods, where x-axis is the number of training episodes, y-axis is the traffic volume (in thousands). It shows that PG-β and PG-β-abs converge faster (in terms of number of training episodes) than other policy gradient methods, and achieve higher traffic volume after $50,000$ episodes. It is worth mentioning that the classical policy gradient method PG-N cannot learn an effective policy in our problem. Moreover, the learning curves of PG-β and PG-β-abs almost overlap, which gives supporting evidence to Conjecture 1. Figure 4(b) presents per episode runtime (in milliseconds) of different policy gradient methods, where PG-β-abs has the shortest per episode runtime. Combining the two figures, we conclude that PG-β-abs is the best approach in terms of both runtime efficiency and optimality. In the following, we implement PG-β-abs to compare with other existing tolling schemes, while using PG-β to denote PG-β-abs for neatness of notation.

Comparison of PG-β with existing tolling schemes under different settings. We now compare PG-β with the following baseline tolling schemes:

1. Fix: fixed toll proportional to average OD demand.
2. DyState: dynamic toll proportional to the state scale.
3. Δ-toll (Sharon et al. 2017).
4. P0: no tolls.

To compare tolling schemes under different settings, we vary the traffic parameter which under evaluation, and keep all the other parameters as stated above. Figure 5 shows the traffic volume obtained by different tolling schemes under different settings, where the y-axis is the traffic volume, and the x-axis is the value of the parameter that is under evaluation. Figures 5a-5b show that, the traffic volume increases linearly w.r.t. the increasing initial state and OD demand ratio, and PG-β works well under different initial state and OD demand scales. Similarly in Figures 5c-5d, with a higher cost-sensitivity level and value of time, the traffic volume of all tolling schemes increases. According to Eq.(3), this is intuitive in the sense that when the travelling cost of a congested road grows (faster than a less congested road), vehicles diverge towards less congested roads with less travelling cost. When these two parameters grow too large, the regulation effect of all tolling schemes becomes saturated and traffic volume converges to a maximum amount. In this case, PG-β has a larger maximum limit than other tolling schemes. Figure 5e shows that PG-β works well with different maximum toll amounts, and it works even better when the maximum toll amount grows. In general, PG-β outperforms existing tolling schemes under all settings. It is worth mentioning that, the state-of-the-art Δ-tolling approach, which does not take a pro-active approach towards changes in the demand side, does not work well in our setting.

To demonstrate that our DyETC framework can be adapted to other objectives, we also evaluate the total travel time of different tolling schemes under the above settings (Figure 6, where the y-axis is the total travel time). We can see that PG-β still significantly outperforms all the other tolling schemes (the less total travel time, the better).

Evaluation on a Real-World Road Network of Singapore Central Region

In this subsection, we evaluate the performance of PG-β for its regulation effect on the morning rush hour traffic of Singapore Central Region. The number of episodes for training PG-β is $500,000$, and the learning rates for the value and policy functions are fine-tuned as 10^{-8} and 10^{-12}, respectively. Figure 7 shows the abstracted road network of Singapore Central Region, where the zones are labelled from 1 to 11. The numbers along the roads denote the travel distance of the adjacent zones, which is

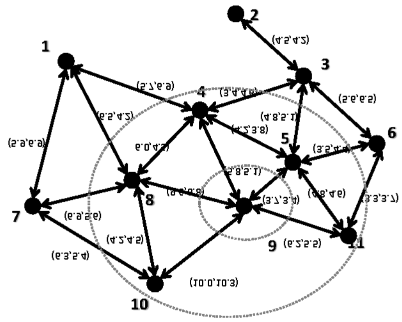
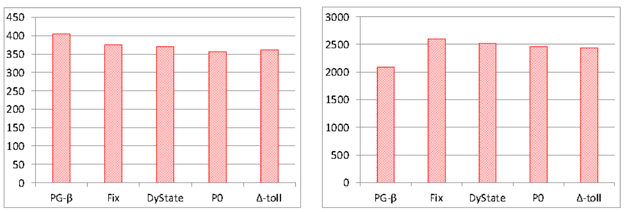


Figure 7: Singapore Central Region road network



(a) Total traffic volume (b) Total travel time

Figure 8: Performance of existing tolling schemes evaluated in Singapore Central Region

obtained from Google Map. Since the OD demand is not revealed by Singapore government, we use the population of different zones to estimate it. The data is obtained from the Department of Statistics (2017) of Singapore in 2016. We first obtain the total number of vehicles as $957,246$ and the population of Singapore as $5,535,002$. The per person vehicle ownership is $956,430/5,696,506 = 0.173$. We then obtain the population of each zone in the Central Region and thus are able to estimate the number of vehicles in each zone (the origin side during morning rush hour). For the destination side, we categorize the 11 zones into 3 types: type 1 is the Downtown Core which is the center of the Central Region, type 2 is the zones that are adjacent to Downtown Core, type 3 is the other zones. We assume a $1 : 0.8 : 0.6$ of demand ratio for these three types of zones. We assume 40% of the vehicles in each zone will go to the Central Region. All the other parameters are estimated as those in the above subsection.

As shown in Figure 8, when applied to Singapore Central Region, DyETC significantly outperforms the other tolling schemes, in terms of both total traffic volume and total travel time. Compared with the second best tolling scheme (Fix), DyETC is able to increase the traffic volume by around 8% (compared with Fix), and decrease the total travel time by around 14.6% (compared with Δ-tolling).

Conclusion & Future Research

In this paper, we propose the DyETC scheme for optimal and dynamic road tolling in urban road network. We make three key contributions. First, we propose a formal model of the DyETC problem, which is formulated as a discrete-time MDP. Second, we develop a novel solution algorithm, PG-β to solve the formulated large scale MDP. Third, we conduct extensive experimental evaluations to compare our proposed method with existing tolling schemes. The results show that on a real world traffic network in Singapore, PG-β increases the traffic volume by around 8%, and reduces the travel time by around 14.6% during rush hour.

Our DyETC scheme can be adapted to various dynamic network pricing domains such as the taxi system (Gan et al. 2013; Gan, An, and Miao 2015) and electric vehicle charging stations network (Xiong et al. 2015; 2016). While our current work focused on a single domain with a relatively small scale traffic network, we will extend DyETC to larger scale networks. Potential approaches include stochastic optimization methods such as CMA-ES (Hansen 2006), continuous control variants of DQN (Mnih et al. 2015) such as DDPG (Lillicrap et al. 2015) and NAF (Gu et al. 2016), parallel reinforcement learning such as A3C (Mnih et al. 2016) and variance reduction gradient methods such as Averaged-DQN (Anschel, Baram, and Shimkin 2017).

Acknowledgments

We would like to thank the anonymous reviewers for their suggestions. This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative. A portion of this work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, IIS-1651089, IIS-1724157), Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

Akamatsu, T. 1996. Cyclic flows, markov process and stochastic traffic assignment. *Transportation Research Part B: Methodological* 30(5):369–386.

Anschel, O.; Baram, N.; and Shimkin, N. 2017. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *ICML*, 176–185.

AutoPASS. 2017. Find a toll station. `http://www.autopass.no/en/autopass`.

Baillon, J.-B., and Cominetti, R. 2008. Markovian traffic equilibrium. *Mathematical Programming* 111(1-2):33–56.

BPR. 1964. Traffic assignment manual. *US Department of Commerce*.

Coulom, R. 2006. Efficient selectivity and backup operators in monte-carlo tree search. In *ICCG*, 72–83.

Gan, J.; An, B.; and Miao, C. 2015. Optimizing efficiency of taxi systems: Scaling-up and handling arbitrary constraints. In *AAMAS*, 523–531.

Gan, J.; An, B.; Wang, H.; Sun, X.; and Shi, Z. 2013. Optimal pricing for improving efficiency of taxi systems. In *IJCAI*, 2811–2818.

Gu, S.; Lillicrap, T.; Sutskever, I.; and Levine, S. 2016. Continuous deep q-learning with model-based acceleration. In *ICML*, 2829–2838.

Hansen, N. 2006. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation* 75–102.

Hausknecht, M., and Stone, P. 2016. Deep reinforcement learning in parameterized action space. In *ICLR*.

Huang, H.-J., and Li, Z.-C. 2007. A multiclass, multicriteria logit-based traffic equilibrium assignment model under atis. *European Journal of Operational Research* 176(3):1464–1477.

Joksimovic, D.; Bliemer, M. C.; Bovy, P. H.; and Verwater-Lukszo, Z. 2005. Dynamic road pricing for optimizing network performance with heterogeneous users. In *ICNSC*, 407–412.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *ECML*, 282–293.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lo, H. K., and Szeto, W. Y. 2002. A methodology for sustainable traveler information services. *Transportation Research Part B: Methodological* 36(2):113–130.

Lo, H. K.; Yip, C.; and Wan, K. 2003. Modeling transfer and non-linear fare structure in multi-modal network. *Transportation Research Part B: Methodological* 37(2):149–170.

LTA, S. 2016. Lta to launch autonomous mobility-on-demand trials. `https://www.lta.gov.sg/apps/news/page.aspx?c=2&id=73057d63-d07a-4229-87af-f957c7f89a27`.

LTA, S. 2017. Electronic road pricing (ERP). `https://www.lta.gov.sg/content/ltaweb/en/roads-and-motoring/managing-traffic-and-congestion/electronic-road-pricing-erp.html`.

Lu, C.-C.; Mahmassani, H. S.; and Zhou, X. 2008. A bi-criterion dynamic user equilibrium traffic assignment model and solution algorithm for evaluating dynamic road pricing strategies. *Transportation Research Part C: Emerging Technologies* 16(4):371–389.

Maei, H. R.; Szepesvári, C.; Bhatnagar, S.; and Sutton, R. S. 2010. Toward off-policy learning control with function approximation. In *ICML*, 719–726.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*, 1928–1937.

Moore, A. W., and Atkeson, C. G. 1993. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning* 13(1):103–130.

Nichols, B. D., and Dracopoulos, D. C. 2014. Application of newton's method to action selection in continuous state-and action-space reinforcement learning. ESANN.

of Singapore, G. 2017. Department of statistics, Singapore. `http://www.singstat.gov.sg/`.

Olszewski, P. 2000. Comparison of the hcm and singapore models of arterial capacity. In *TRB Highway Capacity Committee Summer Meeting*. Citeseer.

Peters, J., and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural networks* 21(4):682–697.

Precup, D.; Sutton, R. S.; and Dasgupta, S. 2001. Off-policy temporal-difference learning with function approximation. In *ICML*, 417–424.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *ICML*, 1889–1897.

Sharon, G.; Hanna, J. P.; Rambha, T.; Levin, M. W.; Albert, M.; Boyles, S. D.; and Stone, P. 2017. Real-time adaptive tolling scheme for optimized social welfare in traffic networks. In *AAMAS*, 828–836.

Sutton, R. S., and Barto, A. G. 2011. Reinforcement Learning: An Introduction.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; Mansour, Y.; et al. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, volume 99, 1057–1063.

Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, University of Cambridge England.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Xiong, Y.; Gan, J.; An, B.; Miao, C.; and Bazzan, A. L. 2015. Optimal electric vehicle charging station placement. In *IJCAI*, 2662–2668.

Xiong, Y.; Gan, J.; An, B.; Miao, C.; and Soh, Y. C. 2016. Optimal pricing for efficient electric vehicle charging station management. In *AAMAS*, 749–757.

Zhang, K.; Mahmassani, H. S.; and Lu, C.-C. 2013. Dynamic pricing, heterogeneous users and perception error: Probit-based bi-criterion dynamic stochastic user equilibrium assignment. *Transportation Research Part C: Emerging Technologies* 27:189–204.