Dynamic Electronic Toll Collection via Multi-Agent Deep Reinforcement Learning with Edge-Based Graph Convolutional Networks

Wei Qiu[1] , **Haipeng Chen**[2] and **Bo An**[1]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Department of Computer Science, Dartmouth College, USA
{wqiu, boan}@ntu.edu.sg, haipeng.chen@dartmouth.edu

Abstract

Over the past decades, Electronic Toll Collection (ETC) systems have been proved the capability of alleviating traffic congestion in urban areas. Dynamic Electronic Toll Collection (DETC) was recently proposed to further improve the efficiency of ETC, where tolls are dynamically set based on traffic dynamics. However, computing the optimal DETC scheme is computationally difficult and existing approaches are limited to small scale or partial road networks, which significantly restricts the adoption of DETC. To this end, we propose a novel multi-agent reinforcement learning (RL) approach for DETC. We make several key contributions: i) an enhancement over the state-of-the-art RL-based method with a deep neural network representation of the policy and value functions and a temporal difference learning framework to accelerate the update of target values, ii) a novel edge-based graph convolutional neural network (eGCN) to extract the spatio-temporal correlations of the road network state features, iii) a novel cooperative multi-agent reinforcement learning (MARL) which divides the whole road network into partitions according to their geographic and economic characteristics and trains a tolling agent for each partition. Experimental results show that our approach can scale up to realistic-sized problems with robust performance and significantly outperform the state-of-the-art method.

1 Introduction

All over the world, traffic congestion is becoming a severe issue in urban areas. ETC systems are one of the most effective approaches introduced by transportation authorities to mitigate traffic congestion. By setting different tolls on different roads, vehicles are regulated to travel on less congested roads with lower tolls. However, traditional ETC prices are pre-defined based on historical data and do not adapt to real-time traffic conditions. The true merit of ETC has not been fully leveraged. To this end, many dynamic pricing schemes have been proposed to tackle these issues [Zhang *et al.*, 2013; Sharon *et al.*, 2017; Chen *et al.*, 2018; Mirzaei *et al.*, 2018] where tolls are computed based on traffic dynamics.

Nonetheless, devising a DETC scheme is still challenging and non-trivial due to its complex problem structure including a large scale road network, dynamic traffic flow, uncertainty in traffic demand, and its essence of a sequential decision-making problem. Conventionally, this sequential decision-making problem can be formulated as a Markov Decision Problem (MDP) and solved by Reinforcement Learning (RL) methods. Recently, Chen et al. [2018] make the first attempt of solving the DETC problem via reinforcement learning. Despite its good performance in mitigating traffic congestion over other methods, a major limitation of the proposed method, PG-β, is that it is limited to a small scale road network and cannot work under realistic-sized road networks. To this end, this paper aims to close the gap by scaling up current state-of-the-art DETC method to large scale real-world road networks with multi-agent deep reinforcement learning and various other novel techniques exploiting the problem characteristics.

The key contributions are summarized as follows: i) First, in order to improve the performance of PG-β, we propose DPG-β, which employs deep neural networks to better represent the policy and value functions of the tolling agent, together with a temporal difference learning to accelerate the update of target values; ii) Second, naive feature representation with deep neural network has only limited improvements over the state-of-the-art methods as it fails to exploit the graph nature of the road network. We apply a novel edge-based graph convolutional neural network (eGCN) to extract the spatio-temporal correlations of the state features for the road network; iii) Moreover, the dimension of the state space is equal to the number of roads, while the action space is the number of tolled roads. For large scale road networks, training of RL models become extremely challenging due to huge state and action spaces. Hence, we propose a novel cooperative multi-agent reinforcement learning (MARL) algorithm which decomposes the state and action spaces into sub-spaces and solves DETC in a divide-and-conquer fashion; iv) Finally, we conduct extensive experimental evaluations in the real world road network of Singapore. Experimental results show that our approach can scale up to realistic-sized problems with robust performance and significantly outperform the state-of-the-art method.

2 Background and Related Works

2.1 Electronic Toll Collection

Originally aimed to alleviate the delay on toll roads and having the merit of collecting tolls without cash and without requiring cars to stop, ETC systems have been widely deployed in highways and arterial roads in urban areas all over the world. For example, ETC has been implemented and deployed in Norway's three major cities: Bergen (1986), Oslo (1990), and Trondheim (1991), and in Singapore (1998) as a congestion pricing system to alleviate traffic conjestion in central areas and restricted zones for expressways and arterial roads. Based on a pay-as-you-use principle, motorists are charged when they pass through ETC gantries during tolling hours. To alleviate traffic congestion, the toll rate varies based on traffic conditions and time periods, which motivates drivers to travel on less congested roads.

2.2 Related Works

ETC has received consistent and considerable attention in recent years. DETC was recently proposed to further improve the efficiency of ETC. Existing DETC schemes [Joksimovic *et al.*, 2005; Lu *et al.*, 2008; Zhang *et al.*, 2013] have been proposed by taking into traffic dynamics into consideration. One major limitation of these schemes is that they unrealistically assume a fixed traffic demand.

Recently, Sharon et al. [2017] propose a dynamic tolling scheme named Δ-tolling. Although Δ-tolling computes tolls based on real-time traffic flows, it still does not consider the traffic demand in the model in a proactive way and may lead to sub-optimal performance. Chen et al. [2018] propose a dynamic model called DyETC which formulates the DETC problem as an MDP by considering the traffic dynamics and gains significant improvement over Δ-tolling. Their proposed algorithm, PG-β, is built on a policy gradient algorithm called *actor-critic* [Konda and Borkar, 1999; Konda and Tsitsiklis, 2000] with Beta distribution as the actor function instead of Gaussian distribution to model a bounded action range. Despite its good performance in traffic congestion alleviation, one major limitation of PG-β is that it only works on partial road networks with 11 zones and cannot scale up to large scale road networks.

Recent advances in function approximation with deep learning have shown promise in learning under high-dimension inputs in reinforcement learning [Mnih *et al.*, 2015; Silver *et al.*, 2016; Schulman *et al.*, 2015; Lillicrap *et al.*, 2015]. Many multi-agent reinforcement learning methods [Sukhbaatar *et al.*, 2016; Gupta *et al.*, 2017; Lowe *et al.*, 2017; Foerster *et al.*, 2017] have been proposed to tackle simple cooperative and competitive problems. However, single agent-based reinforcement learning methods like DQN [Mnih *et al.*, 2015], DDPG [Lillicrap *et al.*, 2015] and multi-agent reinforcement learning like MADDPG [Lowe *et al.*, 2017] cannot solve our DETC because of the large, discrete state as well as large, continuous and bounded action space. By addressing the above issues, a novel multi-agent reinforcement learning method for the DETC problem will be introduced in the following sections.

3 Problem Formulation

3.1 Problem Setup

Formally, the city road network can be abstracted as a directed road network $G = (Z, E, D)$ in which Z is the set of zones, E is the set of roads that connect the zones and D is the set of origin-destination (OD) pairs which define optional paths and traffic demand of both zones.

We denote an OD pair as a tuple $\langle z_k, z_j, q^t_{k,j}, P_{k,j}\rangle$, where zone z_k is the origin, zone z_j is the destination, $q^t_{k,j}$ is the traffic demand at time step t between z_k and z_j, and the $P_{k,j}$ denotes all the acyclic paths from z_k to z_j. There are H time steps for daily-basis DETC planning. Following the *travel time model* [BPR, 1964], the travel time on road e at time step t is $T^t_e = T^0_e[1 + \gamma\,(s^t_e/C_e)^{\xi}]$, where T^0_e is the free-flow travel time, s^t_e is the number of vehicles on road e, and C_e is the capacity of road e. γ and ξ are constants that measure to which extent congestion affects travel time. The travel cost of a path $p \in P_{k,j}$ can be defined as $c^t_{k,j,p} = \sum_{e\in P}(a^t_e + \omega T^t_e)$ where a^t_e is the toll imposed on road e and ω is a constant for the value of time. We use the widely-adopted *stochastic user equilibrium* (SUE) model [Lo and Szeto, 2002; Huang and Li, 2007] as our traffic equilibrium, where the portion of traffic demand from zone k to zone j via path p at time step t is defined as $x^t_{k,j,p} = \frac{\exp\{-\omega' c^t_{k,j,p}\}}{\sum_{p'\in P_{k,j}} \exp\{-\omega' c^t_{k,j,p'}\}}$ where ω' is a constant which reveals vehicles' sensitivity to travel cost. We formulate the problem setup as an MDP in section 3.2.

3.2 A Finite Horizon MDP Formulation

Conventionally, an MDP is defined as a tuple $\langle S, A, r, T\rangle$, which consists of a set of states S, a set of actions A, a reward function $r : S \times A \to r$ and a transition function $T : S\times A \to S$. The goal of the agent for each time step t is to maximize the expected accumulated reward $\sum_{t'=t}^{\infty} \lambda^{t'-t} r^{t'}$ where $\lambda \in [0, 1]$ is the discount factor. It can be achieved by learning an optimal policy $\pi : S \to A$ which outputs an action a given a state s. However, the number of vehicles that reach the destinations depends on the OD demands at each time step due to traffic dynamics. Consequently, the value of a state changes over time and the value of an action is also time-dependent. Due to its advantages of formulating sequential decision making problems, we use finite horizon MDP to model DETC, where we maintain and update a value function and policy function for each time step $t = 0, 1, ..., H$. We define the corresponding elements for our MDP formulation.

State & action. We define the state at time step t, as $\mathbf{s}^t_e = \langle s^t_{e,j}\rangle$ where $s^t_{e,j}$ denotes the number of vehicles that travel to zone j at time step t on road e, and $\mathbf{s}^t = \langle \mathbf{s}^t_e\rangle$ is the state matrix of G at time step t. The action at time step t is defined as $\mathbf{a}^t = \langle a^t_e\rangle$, where a^t_e is the toll set by the transit authority on e that has an ETC gantry.

State transition. The state of the next time step $t+1$ can be formed as $s^{t+1}_{e,j} = s^t_{e,j} - s^t_{e,j,out} + s^t_{e,j,in}$, where $s^{t+1}_{e,j}$ and $s^t_{e,j}$ denote the number of vehicles on road e that travel to zone j at time step $t + 1$ and t respectively. $s^t_{e,j,out}$ is the number of vehicles that go to zone j and exit road e at time step t.

It can be defined as $s^t_{e,j,out} = s^t_{e,j} \cdot \frac{v^t_e \cdot \tau}{L_e} = \frac{s^t_{e,j} \cdot \tau}{T^0_e[1+\gamma(s^t_e/C_e)^\xi]}$, where L_e is the length of road e, $v^t_e = \frac{L_e}{T^t_e} = \frac{L_e}{T^0_e[1+\gamma(s^t_e/C_e)^\xi]}$ is average travel speed of road e, and τ is the time interval between time steps. Similarly, $s^t_{e,j,in}$ is the number of vehicles that go to zone j and enter road e at time step t. It can be defined as $s^t_{e,j,in} = \sum_{D_k=e^- \bigcap e \in p \in P_{k,j}} (q^t_{k,j} + \bar{q}^t_{k,j}) \cdot x^t_{k,j,p}$, where $q^t_{k,j}$ is the traffic demand which comes from zone k to zone j as introduced in Section 3.1. $\bar{q}^t_{k,j}$ is traffic demand which denotes the number of vehicles that come from zone k's neighbouring roads and head for zone j during the period of $t-1$. It can be defined as $\bar{q}^t_{k,j} = \sum_{e^+=k} s^t_{e,j,out}$. Thus,

$$\begin{aligned} s^{t+1}_{e,j} =& s^t_{e,j} - \frac{s^t_{e,j} \cdot \tau}{T^0_e[1+\gamma(s^t_e/C_e)^\xi]} \\ &+ \sum_{D_k=e^- \bigcap e \in p \in P_{k,j}} (q^t_{k,j} + \bar{q}^t_{k,j}) \cdot x^t_{k,j,p} \end{aligned} \quad (1)$$

Reward function. The target of the MDP is to maximize the accumulated reward. We define the reward $r(s^t) = \sum_{e \in E} \sum_{z_j = e^+} \frac{s^t_{e,j}\tau}{T^0_e[1+\gamma(s^t_{e,j}/C_e)^\xi]}$ as the number of vehicles which arrive at destinations at time step t, where e^+ is the ending point of road e and τ is the length of time interval.

Policy & value function. At time step t, a policy $\pi^t(\mathbf{a}^t|\mathbf{s}^t)$ is the conditional probability of taking action $\mathbf{a}^t$ provided state $\mathbf{s}^t$. The value function at time step t can be defined as $v^t(\mathbf{s}^t) = \sum^H_{t'=t} \lambda^{t'} r^t$.

Challenges. There are three key challenges in our formulation: 1) the state space is discrete and high dimensional (w.r.t. the number of roads), 2) the action space is also continuous and high dimensional (w.r.t. the number of tolled roads), and 3) the action space is bounded. Dynamic programming (DP) method is a classic method to solve MDPs. However, solving our formulated MDP with DP can be computationally difficult. Vanilla reinforcement learning methods, such as Q-learning and SARSA [Watkins and Dayan, 1992; Rummery and Niranjan, 1994], also fail under our problem setup because of the large scale and continuous state-action spaces. While policy gradient methods [Williams, 1992; Konda and Borkar, 1999; Lillicrap *et al.*, 2015] work well under large scale MDPs with continuous action space, these methods do not perform well under the bounded action space. Although PG-β [Chen *et al.*, 2018] can solve DETC, it is limited to partial road networks.

4 Solution Algorithm: MARL-eGCN

In this section, we introduce the improved PG-β with deep neural network to better represent the policy and value networks of the tolling agents, together with a temporal difference learning to boost the update of target values. To extract the spatio-temporal correlations of the state features for the road network, we propose a novel edge-based graph convolutional neural network, which is adapted to our problem structure. To tackle the large state-action space in DETC, we decompose the state and action space and solve DETC with our MARL model.

4.1 Deep PG-β with TD-learning

The linear representation of PG-β fails to capture the inner correlation of state and the value function updating method, (i.e., Monte Carlo Method), of PG-β is not efficient because it updates the value function until the end of an episode. We propose DPG-β, which employs deep neural networks to replace the linear representation of PG-β to better represent the policy and value functions of the tolling agent, together with a temporal difference (TD) learning in place of the Monte Carlo method to accelerate the update of target values. In DPG-β, we use two fully connected layers with tanh nonlinearity by adopting the idea of target network [Mnih *et al.*, 2015], we define the TD-error loss function at time step t as

$$\mathcal{L}^t(\phi^t) = \mathbb{E}_{\mathbf{s}^{t+1} \sim T}[(Q^{t+1}_{dpg}(\mathbf{s}^t, \mathbf{a}^t) - y)^2], \quad (2)$$

where ϕ^t is the parameters of the network at time step t and T is the state transition function. $y = r + \lambda \cdot Q^{t+1}_{dpg}(\mathbf{s}^{t+1}, \mathbf{a}^{t+1})$ is the TD-target value. Q^{t+1}_{dpg} is the value function at time step t for policy π^{t+1}. The TD-error is applied to update the policy function, TD-target is applied to update the value function, and the policy distribution is the Beta distribution.

4.2 Edge-Based Graph Convolutional Networks Representation

Naive feature representation with deep neural network has only limited improvements over the state-of-the-art methods as it fails to exploit the graph nature of the road network. As a result, we adopt a graph convolutional networks (GCNs) framework to represent the value and policy functions in the DETC problem. Moreover, the states define the traffic volume of roads (edges) and vehicles transit from roads to roads in road network and impact price of tolled roads in DETC. Unlike ordinary GCNs [Henaff *et al.*, 2015; Niepert *et al.*, 2016; Kipf and Welling, 2016; Defferrard *et al.*, 2016] which take nodes as input, we design a novel edge-based graph convolutional neural network (eGCN) on DETC road network to extract the spatio-temporal correlations of the state (edge) features. Both the feature matrix and adjacency matrix in our problem are defined on edges instead of nodes.

We define eGCN as $\mathbf{f}(\mathbf{s}, \mathbf{V})$ where $\mathbf{s} \in \mathbb{R}^{|E|\times|Z|}$ is the state matrix of a given road network, and $\mathbf{s}^t$ is state matrix at time step t which is defined earlier at section 3.2. V is the adjacency matrix of the edges. We first apply an embedding layer to each layer to obtain a latent matrix $\mathbf{Y}$ and apply a non-linear activation function to $\mathbf{Y}$. We feed a state matrix $\mathbf{s}^t$ at time step t to eGCN. Formally, we adopt the formulation proposed by Kipf and Welling [2016] to model our eGCN with the following layer-wise formulation:

$$\mathbf{Y}^{(l+1)} = \sigma(\tilde{\mathbf{M}}^{-\frac{1}{2}}\tilde{\mathbf{V}}\tilde{\mathbf{M}}^{-\frac{1}{2}}\mathbf{Y}^{(l)}\mathbf{\Phi}^{(l)}), \quad (3)$$

where $\tilde{\mathbf{V}} = \mathbf{V} + I$ and $\mathbf{V}$ is the adjacency matrix defined on the edges. $\mathbf{I}$ is an identity matrix to account for the effect of a node itself. $\mathbf{M}$ is a diagonal node-degree matrix utilized to normalize $\tilde{\mathbf{V}}$. $\mathbf{Y}^l$ and $\mathbf{Y}^{(l+1)}$ are the outputs of the l-th and the $(l+1)$-th layers, respectively. $\mathbf{\Phi}^l$ is a trainable weight matrix for the l-th layer, and $\sigma(\cdot)$ is the activation function. The final output of eGCN is put into the neural network of policy and value function as demonstrated in Figure 1.

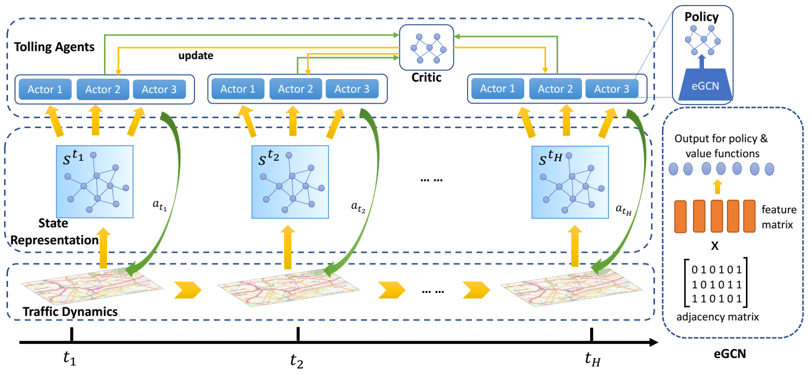


Figure 1: The architecture of the proposed MARL-eGCN with three modules, i.e., the *traffic dynamics*, the *state representation*, and the *tolling agent*. At each time step, state matrix is extracted from the *traffic dynamics* module and fed into the *state representation* (eGCN) module together with adjacency matrix of the road network. Then, the output of eGCN is fed into the *tolling agent* module. The policies generate tolling actions, which are further input into traffic dynamics to change the traffic flow. The *actor* and *critic* of the *tolling agent* are updated according to the reward emitted by the *traffic dynamics*.

4.3 MARL with eGCN

It is very challenging to train one single agent under large state space and large and bounded-continuous action space or training such an agent may lead to sub-optimal performance due to the huge amount of parameters to be optimized. A key observation is that traffic and road networks in metropolises can be partitioned by geographical distance, demographic distribution and electoral divisions.

As shown in Figure 2, there are 4 partitions and each partition has some nodes (zones). The magenta lines in the figure are tolled roads. Therefore, the pricing scheme in each partition is largely based on the traffic condition within the partition and, correspondingly, the pricing scheme between partitions is dominated by traffic conditions between neighbouring partitions. Inspired by this urban characteristic, we propose our multi-agent reinforcement learning (MARL) solution by dividing the planning areas into N partitions. Each partition is treated as one agent, and all the agents cooperate by sharing states with each other to solve the formulated MDP introduced in section 3.2. We adopt the framework of centralized training with decentralized execution. Figure 1 illustrates the architecture of MARL-eGCN.

More concretely, we employ a general and fast policy gradient algorithm, *actor critic* where *actor* is the policy function and critic is the value function, to build our MARL model with eGCN. We use $\theta^t = \{\theta_t^i, ..., \theta_t^N\}$ to denote policy parameters where $t \in [1, ..., H]$ and use ϕ to denote the parameter matrix of the value networks. We define the state of agent i of partition i at time step t as $\hat{\mathbf{s}}^{i,t} = \langle \mathbf{s}^{i,t}, s_{e,j'}^{i,t} \rangle$, where

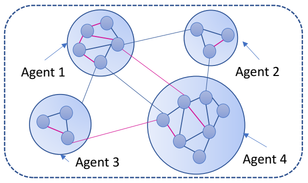


Figure 2: An example of zone partitions.

$\mathbf{s}^{i,t} = \langle \mathbf{s}_e^{i,t} \rangle$. $s_{e,j}^{i,t}$ denotes the number of vehicles on road e of partition i, which travel to zone j of partition i at time step t. $s_{e,j'}^{i,t}$ denotes the number of vehicles on road e of partition i, which head for zone j' of other partitions at time step t. The global state $\mathbf{s}^t$ at time step t is a global status of the traffic condition. The action for agent i at time step t is $\mathbf{a}^{i,t} = \langle a_e^{i,t} \rangle$ and the global action at time step t is $\mathbf{a}^t$, which is a concatenated vector of actions of all agents. The reward r_i^t for each agent i given $\hat{\mathbf{s}}^{i,t}$ at time step t is the number of vehicles that arrive at their destinations.

Algorithm 1: MARL-eGCN

```
   input: $\theta_t^i \in \mathbb{R}, j \in [1, ..., N]$ and $\phi$
1  for $e \leftarrow 0$ to MAX-EPISODE do
2      while $t = 1 <$ MAX-EPISODE-LENGTH do
3          for agent $i$ in $N$ do
4              $\rho_t^i = \mathbf{V}_i \times \mathbf{f}(\mathbf{s}^t, \mathbf{V})$;
5              $\mathbf{a}^{i,t} = \pi(\rho_t^i, \theta_t^i)$;
6          Concatenate $\mathbf{a}^{i,t}, i \in [1, .., N]$ into $\mathbf{a}_t$;
7          Take $\mathbf{a}^t$ into traffic road graph and get $\hat{\mathbf{s}}^{i,t+1}$;
8          foreach $i \leftarrow 0$ to $N$ do get $r_t^i$;
9          for $i \leftarrow 0$ to $N$ do
10             $y_t^i = \sum_{t'=t}^{H} r_{t'}^i$;
11             Update critic by minimizing the loss:
                 $\mathcal{L}(\phi) = y_t^i - Q(\rho_t^i, \mathbf{a}_t)$;
12             Update actor using the policy gradient:
                 $\nabla_{\theta_t^i} \mathbf{J} = \nabla_{\theta_t^i} \pi_i(\mathbf{a}^{i,t} | \rho_t^i) \cdot Q(\rho_t^i, \mathbf{a}^{i,t})$;
13         $\hat{\mathbf{s}}^{i,t} = \hat{\mathbf{s}}^{i,t+1}$;
14 return $\theta_t^i, i \in [1, ..., N]$;
```

Then we build a local adjacency matrix $\mathbf{V}_i$ with size $K \times N$ for edges in partition i with its K neighbours where each row is a binary vector indicating neighbours of agent i. Finally the individual input for agent i is $\rho_t^i = \mathbf{V}_i \times \mathbf{f}(\mathbf{s}^t, \mathbf{V})$. We use Beta distribution as the policy distribution for the bounded and continuous action space, then we define the set of policies for each agent as $\pi = \{\pi_i\}_{i=1,..,N}$. Hence, we can derive the gradient of the performance measure $\mathbf{J}$ for policy function of agent i as:

$$\nabla_{\theta_t^i} \mathbf{J}(\theta_t^i) = \mathop{\mathbb{E}}_{\mathbf{s}_{t+1} \sim T} [\nabla_{\theta_t^i} \log \pi_i(\mathbf{a}^{i,t} | \rho_t^i) \cdot Q^t(\rho_t^i, \mathbf{a}^{i,t})], \quad (4)$$

where $Q^t(\rho_t^i, \mathbf{a}^{i,t})$ is the centralized action-value function that inputs ρ_t^i that output by eGCN at time step t as shown in line 4 in Algorithm 1. ρ_t^i is the spatio-temporal state for agent i related to the partition of agent i, which is different from the input for vanilla actor critic. We define loss function of value function as follows:

$$\mathcal{L}^t(\phi^t) = \mathbb{E}_{\mathbf{s}_{t+1} \sim T} \left[(Q^t(\rho_t^i, \mathbf{a}^{i,t}) - y)^2 \right], \quad (5)$$

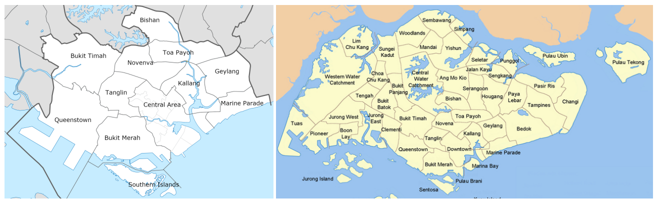
where ϕ is the parameter matrix of $\mathcal{L}$ and $y = r_i + \lambda \cdot Q^t(\rho_t^i, \mathbf{a}^{i,t})$ is the TD-target value of agent i, which is estimated by the value network and Q^t is the value function at time step t. We call our MARL model MARL-eGCN. Algorithm 1 illustrates the detailed description of our MARL-eGCN where embedded features are output by eGCN in line 4, the policy outputs action vector for each agent in line 5 and the parameters of *actor* and *critic* are updated in line 11 and 12.

5 Experimental Evaluation

In this section, we evaluate our proposed approach on real-world road networks of Singapore.

5.1 Experiment Setup

We first introduce our problem scenarios, compared methods, parameters of model and the training settings.



(a) Singapore central region (b) Entire Singapore planning areas

Figure 3: Map of Singapore

Problem scenarios. As shown in Figure 3, we choose the road network of Singapore central region (Central Net) and the road network of Singapore all planning areas (Whole Net) as our experimental scenarios. There are 11 zones (planning areas) in Central Net and over 33 zones in Whole Net. We create an abstract road network for both the Central Net and Whole Net based on the distribution of ETC gantries, the arterial roads and highways network of Singapore. We merge some zones which have low population and small acreage into their neighbouring zones, and thus we get 11 zones and 40 edges (roads) for Central Net and 33 zones and 124 edges for Whole Net. We estimate the OD demand for the two road network by using population data of each zone and Annual Vehicle Statistics 2017 published by Singapore government [LTA, 2017]. We first obtain the total number of vehicles as 961,842 and the population of Singapore as 5,612,300, and thus we get the per person vehicle ownership rate as 0.171. Then we obtain the population of each zone and finally estimate the number of vehicles of each zone. We set $\gamma = 0.15$ and $\xi = 4$ according to [BPR, 1964], and we set the toll range $[0, 6]$ for each arterial road based on the current ETC scheme in Singapore. The time horizon H is set as 6 and each time step is 2 hours.

Compared methods. The methods that we evaluated include (i) PG-β, (ii) DPG-β, (iii) DPG-β-eGCN, which combines DPG-β with eGCN, (iv) MARL-eGCN and (v) MARL (MARL-eGCN without eGCN). The first 3 methods are single-agnet methods, while the last two are multi-agent methods. For multi-agent methods, we train 2 agents on Central Net and 4 agents on Whole Net, respectively.

Parameters of model & training settings. The discount factor λ of PG-β is 1 (as that in [Chen *et al.*, 2018]) and set as 0.9 for the other models. There are 2 dense layers for all neural networks each with 128 neurons for single-agent models and each with 64 neurons for all multi-agent models. The learning rates of policy and value function for single agent models are 0.0001 and 0.0005 respectively. The learning rates of policy and value function for multi-agent models are 0.001 and 0.0005 respectively. We use tanh activation function for all neural networks. We use 2 layers for eGCN, with a shape of $(|Z|, 16)$ and $(16, 16)$, respectively. We use *Adam* to optimize deep neural networks. We train PG-β for 500,000 episodes and the other models for 100,000 episodes. We implement the traffic dynamics with C++ and create extension file for Python with Boost.Python[1] to speedup the Python code. All models are implemented by Python and run on a 64-bit server with 500 GB RAM and 80 Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz processors.

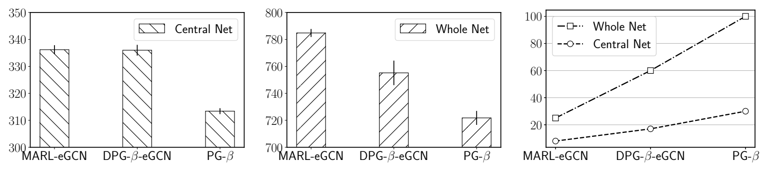
5.2 Result Analysis

We use the number of vehicles that arrive at their destinations (i.e., traffic throughput) and the number of episodes for convergence as the evaluation criteria.

Central net. We first compare MARL-eGCN, DPG-β-eGCN and PG-β on the road network of Singapore central region. As illustrated in Figure 4, under the scenario of Central Net, both DPG-β-eGCN and MARL-eGCN significantly outperform PG-β. We notice that for a small scale road network like Central Net, the multi-agent (MARL-eGCN) and single agent (DPG-β-eGCN) models have similar traffic throughput. In terms of training efficiency, we can see that compared with PG-β, DPG-β-eGCN uses around 50% of episodes to converge, while MARL-eGCN uses only 27.7%.

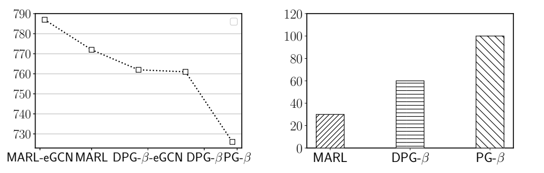
Whole net. To further demonstrate the advantage of MARL-eGCN in large scale road networks, we conduct experiments on Whole Network. As shown in Figure 4, MARL-eGCN and DPG-β-eGCN are superior to PG-β. Compared with PG-β, MARL-eGCN gains 8.4% larger traffic throughput and uses only 25% of episodes to get stable results. The

[1]https://www.boost.org/doc/libs/1_68_0/libs/python/doc/html/index.html



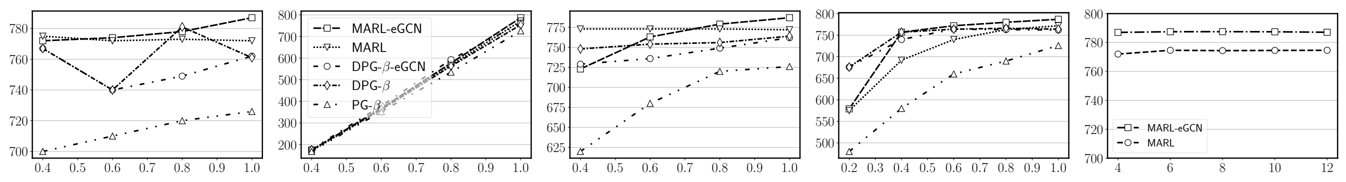
(a) Traffic throughput (b) Traffic throughput (c) Compare convergence

Figure 4: Traffic throughput (95% confidence interval) and convergence comparison (values in thousands)



(a) Traffic throughput (b) Compare convergence

Figure 5: Ablation study (values in thousands)



(a) Initial state (b) Initial demand (c) Cost rate (d) Price rate (e) Number of agents

Figure 6: Traffic throughput (in thousands) under various traffic conditions (Fig. a, c, d share the same legend in Fig. b)

results demonstrate that MARL-eGCN can get better traffic throughput with fewer training episodes.

5.3 Ablation Study and Robustness Test

As MARL and eGCN are the two key components of MARL-eGCN, we conduct an ablation study on Whole Net to investigate the contribution of these components. The traffic throughput results are shown in Fig 5(a), and the convergence comparison results are shown in Fig 5(b).

First, by comparing MARL-eGCN with MARL, we can see that after removing eGCN, MARL gets 2% lower traffic throughput and 20% larger number of episodes for convergence, compared with MARL-eGCN. Then, we remove both MARL and eGCN, and compare MARL-eGCN with DPG-β. Results show that MARL-eGCN outperforms DPG-β with 3.5% improvement in terms of traffic throughput and only half number of training episodes for convergence.

To compare tolling schemes under different traffic settings, we vary one parameter and keep other parameters fixed for further evaluation. Figure 6 depicts traffic throughput obtained from different pricing schemes under different parameter settings where x-aixs is the value of parameter under evaluation and y-axis is the traffic throughput (in thousands). As shown in Figure 6(a), the traffic throughput increases linearly w.r.t the increasing initial state except for PG-β and its variants that are more unstable under various initial states. In Figure 6(b), all models are sensitive to the changing initial demand, which is intuitive because OD demand is the main factor that impacts the traffic throughput. Similarly in Figure 6(c), with higher cost rate, traffic throughput increases nearly for all tolling schemes among which PG-β is more sensitive to the cost rate. Under different maximum price rates, in Figure 6(d), MARL-eGCN outperforms other methods. By adding more agents, as illustrated in Figure 6(e), we found that MARL-eGCN gets the largest traffic throughput under 8 agents and traffic throughput decreases with over 8 agents, while for MARL, the optimal number of agents is 6. This is because more agents makes learning harder due to feature redundancy and larger coordination overhead, while fewer agents cannot leverage the power of multi-agent learning. In general, MARL-eGCN outperforms existing tolling methods under all settings, and is consistently better than the state-of-the-art method PG-β. Compared with a single super agent which may lead to sub-optimal behaviours in some environments, our MARL approach can be more general and adaptive to various scenarios.

6 Conclusion

To scale up the state-of-the-art RL-based approaches to the DETC problem, we propose a novel learning architecture called MARL-eGCN. The two intrinsic ideas of MARL-eGCN are: (i) we design a problem-specific cooperative multi-agent RL framework to decompose the huge state and action spaces into sub-spaces and solve the DETC problem in a divide-and-conquer manner, and (ii) we devise an edge-based GCNs representation of the value and policy functions for the tolling agents, so as to exploit the inter-correlations among edges in the road network. By performing extensive experimental evaluations on a real-world traffic network in Singapore, we show that MARL-eGCN significantly outperforms the state-of-the-art approaches in terms of both traffic throughput and training efficiency, and is able to handle real-world sized DETC problems.

Acknowledgments

We would like to thank the anonymous reviewers for their suggestions. This research is supported by MoE, NRF, NTU, ONR grants N00014-18-1-2670 and N00014-16-1-2896.

References

[BPR, 1964] BPR. Traffic assignment manual. *US Department of Commerce*, 1964.

[Chen *et al.*, 2018] Haipeng Chen, Bo An, Guni Sharon, Josiah P Hanna, Peter Stone, Chunyan Miao, and Yeng Chai Soh. Dyetc: Dynamic electronic toll collection for traffic congestion alleviation. *AAAI*, pages 757–765, 2018.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering, 2016.

[Foerster *et al.*, 2017] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*, 2017.

[Gupta *et al.*, 2017] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *AAMAS*, pages 66–83, 2017.

[Henaff *et al.*, 2015] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[Huang and Li, 2007] Hai-Jun Huang and Zhi-Chun Li. A multiclass, multicriteria logit-based traffic equilibrium assignment model under atis. *European Journal of Operational Research*, 176(3):1464–1477, 2007.

[Joksimovic *et al.*, 2005] Dusica Joksimovic, Michiel CJ Bliemer, Piet HL Bovy, and Zofia Verwater-Lukszo. Dynamic road pricing for optimizing network performance with heterogeneous users. In *Networking, Sensing and Control, 2005. IEEE*, pages 407–412, 2005.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Konda and Borkar, 1999] Vijaymohan R Konda and Vivek S Borkar. Actor-critic–type learning algorithms for markov decision processes. *SIAM Journal on Control and Optimization*, 38(1):94–123, 1999.

[Konda and Tsitsiklis, 2000] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *NIPS*, pages 1008–1014, 2000.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

[Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NIPS*, pages 6379–6390, 2017.

[LTA, 2017] LTA. Annual motor vehicle population by vehicle quota categories. https://data.gov.sg/dataset/annual-motor-vehicle-population-by-vehicle-quota-category, 2017. Accessed: 2016-04-12.

[Lu *et al.*, 2008] Chung-Cheng Lu, Hani S Mahmassani, and Xuesong Zhou. A bi-criterion dynamic user equilibrium traffic assignment model and solution algorithm for evaluating dynamic road pricing strategies. *TR Part C: Emerging Technologies*, 16(4):371–389, 2008.

[Mirzaei *et al.*, 2018] Hamid Mirzaei, Guni Sharon, Stephen Boyles, Tony Givargis, and Peter Stone. Link-based parameterized micro-tolling scheme for optimal traffic management. In *AAMAS*, pages 2013–2015, 2018.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Niepert *et al.*, 2016] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016.

[Rummery and Niranjan, 1994] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using Connectionist Systems*, volume 37. University of Cambridge, 1994.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, pages 1889–1897, 2015.

[Sharon *et al.*, 2017] Guni Sharon, Josiah P Hanna, Tarun Rambha, Michael W Levin, Michael Albert, Stephen D Boyles, and Peter Stone. Real-time adaptive tolling scheme for optimized social welfare in traffic networks. In *AAMAS*, pages 828–836, 2017.

[Silver *et al.*, 2016] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[Sukhbaatar *et al.*, 2016] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. In *NIPS*, pages 2244–2252, 2016.

[Watkins and Dayan, 1992] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

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

[Zhang *et al.*, 2013] Kuilin Zhang, Hani S Mahmassani, and Chung-Cheng Lu. 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, 2013.