2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Multiagent Reinforcement Learning Algorithm for Distributed Dynamic Pricing of Managed Lanes

Venktesh Pandey[1] and Stephen D. Boyles[2]

Abstract— **Priced managed lanes are increasingly being used to manage congestion on urban freeways. This research looks at a distributed model of dynamic pricing for managed lanes with multiple entrances and exits with a toll agent controlling the toll at each diverge point. We formulate the problem as a Markov decision process for each agent and use a multiagent reinforcement learning algorithm to find toll policies that maximize revenue over a finite time horizon. We also propose a local policy search method which explores the continuous action space without the need to discretize tolls. We compare the performance of the distributed control against other heuristics used in practice. Experiments conducted on the test networks show promising results. The proposed algorithm generates $70-86\%$ more revenue than the other heuristics which assume no coordination and produces toll policies which reduce the violation of free flow travel on managed lanes to only 5% of the times. Despite showing lack of convergence within a reasonable computation time, the proposed algorithm generates toll policies which perform better than the existing heuristics and provides a viable alternative for dynamic tolling of managed lanes with multiple toll locations.**

I. Introduction

A. Background

Priced managed lanes (ML), also referred 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 using the existing capacity of the roadway. Travelers pay toll which changes with the time of day or dynamically based on the congestion pattern to experience free flow travel time from their origin to their destination.

Dynamic tolls paid by a traveler may differ based on the entrance location. Optimizing these toll prices given the proximity of toll segments is a complex optimization problem with multiple decision variables. The existing literature formulates the problem as a Markov decision process (MDP) and uses Q-learning [1], stochastic dynamic programming [2], or value function approximation methods [3] to solve the toll pricing problem. However, these methods make limiting assumptions about driver behavior like once a traveler enters the managed lane, they do not exit it until the destination is reached, or rely on optimizing one toll variable that applies uniformly for all entrance points.

This research article relaxes the assumptions in the literature and solves the dynamic pricing problem as a distributed multiagent control problem. The problem is formulated as a cooperative MDP and solved using a multiagent reinforcement learning (MARL) algorithm based on coordination graphs [4].

[1]Venktesh Pandey is a Ph.D. student at the Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Austin, TX 78712, USA venktesh@utexas.edu

[2]Stephen D. Boyles is an associate professor at the Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Austin, TX 78712, USA sboyles@mail.utexas.edu

B. Related work

Here we present a review of the literature in the field of managed lane pricing and the use of MARL algorithms in the traffic control literature.

Optimizing pricing for managed lanes with multiple entrance and exit points has been a recent area of research. Yang et al. [2] use stochastic dynamic programming and Zhu and Ukkusuri [1] use the R-Markov average reward technique to solve the optimal toll rate (single variable) at each time step. In both models, the authors make the assumption that travelers once onto the managed lane will not change their routes. Pandey and Boyles [3] relax this assumption by proposing a decision route framework and propose a value function approximation based algorithm to optimize a single variable toll. In this article, we extend this previous work to a multiagent setting using a model-free reinforcement learning algorithm.

Reinforcement learning (RL) algorithms have been used for optimal decision making in several traffic control problems involving active traffic management including adaptive traffic signal control [5], ramp metering [6], [7] and variable speed limits [8].

Applied in a multiagent setting, MARL algorithms use the MDP architecture to determine coordinated actions across all agents. Rezaee [6] and El-Tantawy et al. [9] present a coordination graph based approach to determine optimal coordinated action for multiple agents for the ramp metering and traffic signal control applications, respectively. When applied to traffic settings, the MARL problem is assumed to have the following characteristics: full cooperation between different agents; fix locations of the agents in space; and fixed shared objective across all agents (like minimize total delay or maximize revenue etc.) [6]. In this article, we build on a sparse cooperative Q-learning approach proposed in [10].

C. Contributions and outline

The primary contributions of the article are the following:

1) We develop a distributed model for dynamic pricing of managed lanes with multiple entrances and exits that scales for networks with multiple toll segments.
2) With an appropriate selection of the lane choice model, we develop a method to explore continuous toll action space for all agents which obviates the need to generate

and evaluate discrete toll values like previously done in [1] and [3].

In addition to these primary contributions, we provide another application of MARL algorithms in the area of dynamic tolling of managed lanes that shows improved performance against existing heuristics.

The rest of the article is organized as follows. Section II presents the model assumptions and details. Section III describes the solution methods used to solve the problem. Section IV presents results from the simulations conducted on two test networks. Section V concludes the paper and identifies the directions for future work.

II. Optimization model notations, assumptions, and formulation

A. Assumptions

Consider a managed lane network shown in Fig. 1(a). The upper set of links form managed lanes (ML) and the lower set of links form GP lanes. As we describe the network, we label the assumptions made in the our model as "$A\#$".

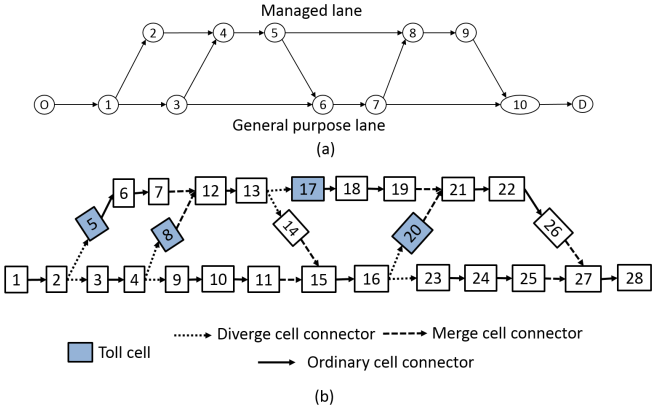

Fig. 1. (a) Managed lane network with multiple entrances and exits, (b) and the same network represented using cells from the cell transmission model (Source: [3])

We assume that there is only one origin and destination point for all travelers in the network ($A\#1$). This assumption can be relaxed by disaggregating the traffic flow based on its origin and destination for additional origins and destinations.

As travelers continue to travel towards the destination, they make routing decisions at diverge locations. Nodes $1, 3, 5$, and 7 are the diverge locations for the network in Fig. 1(a). At each diverge node, travelers use the information about the current travel time and toll values to make a lane choice decision. We assume that the information about the current travel time is provided by measuring instantaneous travel time ($A\#2$) and that all travelers have complete information about the current network state ($A\#3$).

We model lane choice using a value of time (VOT) distribution where each traveler is assumed to have a certain value of time and they choose a path that minimizes the linear combination of toll and travel time, converted to the same units using their VOT value ($A\#4$). We further assume that at each decision point, travelers compare the current utility along a certain set of routes associated with each diverge location, which are called decision routes at each diverge location ($A\#5$). Decision routes are defined as the set of routes connecting the current diverge node to the first merge node located immediately downstream of the first exit from the managed lane if a traveler enters the lane at the current diverge node. We borrow this definition from [3].

Each decision point is monitored by a toll agent which controls the toll for travelers entering the managed lane at that location. We assume that the toll rate is distance-based with units \$/mile ($A\#6$). We also consider that travelers pay the toll rate they see while entering the managed lane and continue to pay that rate until the next decision point or the end is reached ($A\#7$). We focus on the revenue maximization objective since it is one of the primary objectives for operating managed lanes. The model can be easily extended for other objectives.

The problem is formulated as a finite horizon MDP. The demand distribution at the origin is assumed to be known ($A\#8$). The entire problem is modeled as a deterministic process ($A\#9$). This assumption is made to simplify the understanding of the results; making the problem stochastic will be a part of the future work.

B. Notation

We divide the time horizon into equal time steps with $\mathscr{T} = \{1, 2, \ldots, T\}$ denoting the set of all time intervals, where T is the final time step. Set $\mathscr{C}$ represents the set of all cells in the network and the set of all diverging cells is denoted by $\mathscr{C}_D \subset \mathscr{C}$. We follow the Godunov scheme for discretizing the links into cells [11]. Fig. 1(b) shows the discretized representation of the network in Fig. 1(a).

We define a toll cell as the cell immediately after the diverge point which leads a traveler towards the managed lane. The toll is charged only for vehicles entering this cell. Set $\mathscr{C}_{\text{toll}}$ represents the set of toll cells. In the distributed pricing model, each toll agent regulates the toll at each toll cell. A toll agent $n \in N$ manages the toll rate at the toll cell immediately following a diverge cell i. We denote the toll rate per mile set by an agent n at time step t as $\beta_n(t)$, which is bounded by its minimum (β_{min}) and maximum (β_{max}) values.

A discrete VOT distribution is used to model lane choice. It is denoted by a set of possible VOT values $v \in V$, where the proportion of each class in the population is denoted by p_v ($\sum_v p_v = 1$). We define $x_i^v(t)$ as the number of vehicles of VOT class v in cell i at time step t. Throughout the rest of the article, variables i, n, and v are used to index variables of the set of all cells, agents, and VOT values respectively.

C. Multiagent Markov Decision Process Model

In this section, we explain the formulation of the toll pricing model as a cooperative MDP and its relaxed version under certain assumptions.

1) Complete MDP: The complete MDP problem has following parameters:

- Finite number of time steps $t \in \mathscr{T}$ and finite number of agents $n \in N$
- State vector time step t, denoted by $s(t)$ and defined as a vector containing the number of vehicles of each VOT class in each cell in the network. Mathematically, $s(t) = \{x_i^v(t) \mid i \in \mathscr{C}, v \in V\}$
- Action vector at time step t, denoted by $a(t)$ and defined as $a(t) = \{\beta_n(t) \mid \forall n \in N\}$
- Deterministic transition function f determines the state at the next time step given current state and action vectors, that is $s(t+1) = f(s(t), a(t))$. The f function is governed by the traffic flow update equations in [11] with updated dynamic for lane choice at a diverge like explained in [3]
- Reward function $R(s(t), a(t))$ determines the one step reward obtained from taking action $a(t)$ in state $s(t)$. For our problem, the reward is the product of the number of vehicles choosing the managed lane times the toll rate per mile times the length of travel on the managed lane. Additionally, since the managed lane is to kept uncongested at all times, we penalize tolls which push more vehicles towards managed lanes than required with a reward of -100

The objective of the model is to find a policy $\pi : s(t) \rightarrow a(t)$ which maximizes the total sum of one step reward across all time steps and agents, given the initial state $s(0)$. Since the one step reward depends on the joint action of all agents, each agent has to collaborate with the others to obtain an optimal policy. We define the optimal value of being in a state $s(t)$ by value function $V^*(s(t))$ which represents the total reward obtained from starting in state $s(t)$ at time t and choosing optimal actions thereafter. At optimality, the value functions satisfy the Bellman equation (1):

$$V^*(s(t)) = \max_{a(t)} \{R(s(t), a(t)) + V^*(s(t+1))\} \quad (1)$$

2) Relaxed MDP: Solving optimal policies in a multi-agent setting where the actions are a continuous function of time is a challenging task. The regular Q-learning or value function approximation methods fail due to the curse of dimensionality. If toll agents collaborate and need to coordinate their actions with few "neighboring" agents only, we can approximate the value function of a state as the sum of value functions defined for each agent as shown in (2):

$$V^*(s(t)) = \sum_{n \in N} V_n^*(s_n(t)) \quad (2)$$

,where, $V_n^*(s_n(t))$ is the value function associated with agent n defined at a local state vector $s_n(t)$. This value function denotes the total future reward obtained by agent n starting from local state $s_n(t)$ at time step t and assuming all agents take joint optimal actions thereafter. The local state vector for agent n is defined as the number of vehicles of each VOT class in each cell located along the decision route for the diverge cell associated with agent n. Substituting (2) in (1) for both $s(t)$ and $s(t+1)$, and decomposing the reward function as the sum of reward function for each agent $(R_n(s(t), a(t)))$, we can write a new form for the Bellman equation decomposed for each agent, similar to the Q function decomposition in [12]:

$$V_n^*(s_n(t)) = \max_{a(t)} \{R_n(s(t), a(t)) + V_n^*(s_n(t+1))\} \ \ \forall n \in N \quad (3)$$

III. Solution methods

To solve the relaxed MDP model, we use a variant of the sparse cooperative Q-learning algorithm from [10], where we replace learning Q-functions for each agent and state with learning value functions for each local state for each agent. We call this algorithm `SparseV`. The algorithm estimates the value function, $V_n(s_n(t))$ for each agent and at each time step. The basic structure of the algorithm is presented in Algorithm 1 where the policies are explored using an ϵ-greedy approach and the estimates for values functions are updated using a step size which decreases harmonically with time. A superscript m on $V_n^m(s_n(t))$ indicates the iteration number.

Algorithm 1 `SparseV` using look-up table representation

Step 0: Initialization
Initialize $V_n^0(s_n(t)) \quad \forall n \in N, t \in \mathscr{T}, s_n(t)$
Choose initial state $s(0)$ and set $m = 0$
Step 1: Simulating a policy
Set $\widehat{V}_n^m(s_n(t))$ =null $\forall s_n(t)$
for $t \in \{1, 2, \ldots, n(\mathscr{T})\}$ **do**
  if random number between 0 and 1 less than ϵ **then**
    Select $a(t)$ randomly between $\max\{\beta_{\min}, a(t-1) - \$0.25\}$ and $\min\{\beta_{\max}, a(t-1) + \$0.25\}$
  else
    $a(t) =$ `localPolicySearch`$(a(t-1), V_n^m(s_n(t)))$
  Determine $s(t+1) = f(s(t), a(t))$ and $s_n(t+1)$
  Determine the updated value function estimate for state $s_n(t)$:
  $\widehat{V}_n^m(s_n(t)) = R_n(s(t), a(t)) + V_n^m(s_n(t+1)))$
Step 2: Update the V values for the visited states
Step size update: $\alpha_m = 20000/(20000 + m)$
for $t \in \{T, \ldots, 3, 2, 1\}$ in the reverse order of time and **for each** agent $n \in N$ **do**
  if $\widehat{V}_n^m(s_n(t))$ is NOT null **then**
    $V_n^{m+1}(s_n(t)) = (1 - \alpha_m) V_n^m(s_n(t)) + \alpha_m \widehat{V}_n^m(s_n(t))$
if $m >$ max number of iterations or if V values converged **then**
  Stop. Report $V_n^m(s_n(t))$ as the final value estimates
else
  $m \leftarrow m + 1$ and go back to Step 1

To find an optimal joint action given the current value function estimates, we assume that the action of an agent

is influenced only by its "downstream neighboring agents" ($A\#10$). We define downstream neighboring agents as agents located downstream of the current agent which lie on the decision routes associated with the current agent's diverge cell. This assumption is reasonable since the toll values set by all downstream neighboring agents immediately impacts the decisions made by the travelers at current agent's toll gantry.

We show this influence relationship using a directed coordination graph (CG).The nodes of a CG represent the agents and edges connect agents which are assumed to influence actions of each other. If an edge is directed from agent n_1 towards agent n_2, then the action of agent n_1 influences the action of agent n_2. Given the managed lane network is acyclic, the CG is also acyclic and thus has a topological order. For the managed lane network in Fig. 1, the CG is shown in Fig. 2.

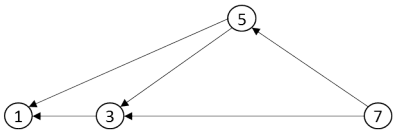

Fig. 2. Coordination graph for the network in Fig. 1 with agents as nodes and edges connecting agents representing interdependencies

The `localPolicySearch()` method solves the optimal action of each agent given the current value function estimates (shown in detail in Algorithm 2). It visits agents in the topological order of the CG and determines the optimal action assuming the action of all downstream neighboring agents is fixed. It first finds the threshold toll values for each agent corresponding to each VOT class (β_n^v). The calculation of these threshold values is explained later. Next, it evaluates the gain g_n^v for agent n for each threshold toll β_n^v and sets the action of the agent to the threshold toll that results in the maximum gain. We define gain as the sum of the total one step revenue for that agent and the value of the resulting next state from the joint action. Unlike the approach in [6] and [9] where the action of agents are continuously changed till no agent can cause any gain by changing its action (having no guaranteed stopping point), our approach terminates after one sweep of the CG and thus the computation time is linear in the number of agents.

Algorithm 2 `localPolicySearch`$(a(t-1), V_n^m(s_n(t)))$

Set $a(t) = a(t-1)$
for agent n in the topological order of the CG **do**
  Determine β_n^v for all $v \in V$
  for $v \in V$ **do**
    Set $\beta_n(t) = \beta_n^v$ and determine gain:
    $g_n^v = R_n(s(t), a(t)) + V_n^m(s_n(t+1))$
  Let $\bar{v} = \text{argmax}_v\ g_n^v$. Set $\beta_n(t) = \beta_n^{\bar{v}}$
Return $a(t)$

The toll threshold values for each agent enable the search on a continuous action space. We demonstrate the evaluation of threshold tolls using an example. Consider the diverge node 5 on the network in Fig. 1. There are three decision routes over which a traveler compares the utility using instantaneous travel time and toll values. Table I shows these paths, and the instantaneous travel time, the toll, and the total disutility for a vehicle with VOT value v for each path. The route $\{5, 8, 9, 10\}$ leads a traveler towards the managed lane.

TABLE I
DISUTILITY COMPARISON OVER DECISION ROUTES FOR AGENT 5 FOR A VEHICLE OF VOT CLASS v

Decision route	Inst. travel time	Inst. toll	Total disutility
$\{5, 8, 9, 10\}$	τ_1	β_1	$\beta_1 + v\tau_1$
$\{5, 6, 7, 8, 9, 10\}$	τ_2	β_2	$\beta_2 + v\tau_2$
$\{5, 8, 7, 10\}$	τ_3	0	$v\tau_3$

The goal is to determine β_1 given the instantaneous value of travel times τ_1, τ_2, and τ_3, and the assumed fixed value of β_2. Assuming discrete VOT distribution, the value of β_1 which lets vehicles of VOT class v onto the managed lane is the one which causes the route $\{5, 8, 9, 10\}$ to have the minimum disutility. That is, if we define the threshold toll value corresponding to VOT class v (β_1^v) as in (4) and (5), then for all $\beta_1 < \beta_1^v$, all vehicles of VOT class v will choose the managed lane. This method reduces the search on a continuous action space to a search over finite β_1^v values.

$$\beta_1^v = \min\{\beta_2 + v(\tau_2 - \tau_1), v(\tau_2 - \tau_1)\} \tag{4}$$

$$\beta_1^v = \min\{\beta_{\max}, \max\{\beta_{\min}, \beta_1^v\}\} \tag{5}$$

We compare the performance of the `SparseV` algorithm against following heuristics. The first two are feedback control based heuristics which seek to maintain the traffic flow operation on the managed lane at the desired level and are commonly used in the field implementations. The last heuristic generates the toll profiles randomly.

1) Density based heuristic (`Density`): Using this heuristic, each toll agent monitors the density on the managed lane cells downstream of the current diverge cell. If the density is different from the desired density, the toll is increased or decreased using a regulator parameter.
2) Ratio based heuristic (`Ratio`): Similar to the `Density` heuristic, each toll agent monitors the ratio of the density on the managed lane cells to the density on the GP cells downstream of the current diverge cell. If the ratio is different from the desired ratio, the toll is increased or decreased using a regulator parameter.
3) Random search (`Random`): We simulate 100,000 random policies where the action of each agent is chosen randomly and select the policy which generates highest revenue.

IV. RESULTS

We test the performance of the algorithms on two networks shown in Fig. 3. The first network has double entrances and a single exit (DESE) with two agents, while the second

network (LBJ) is an approximation of the 3.5-mile long toll segment 2 of the LBJ TEXpress lanes in Dallas, TX with four agents.

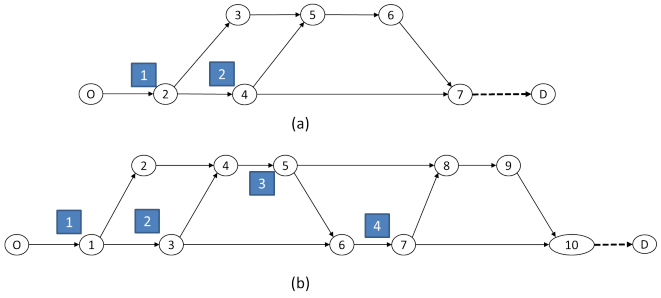

Fig. 3. Two test networks: (a) DESE network; (b) LBJ TEXpress toll segment 2 abstract network. The dashed link indicates a bottleneck

We consider five VOT classes with VOT values as \$10/hr, \$15/hr, \$20/hr, \$25/hr, and \$30/hr, with assumed known proportions of demand as $0.1, 0.4, 0.2, 0.2$, and 0.1 respectively. The traffic flow follows a trapezoidal fundamental diagram with free flow speed as 60 mph, back wave speed as 20 mph, link capacity as 2200 veh/hr/lane, and jam density as 265 veh/mile. Each time step is assumed 6 seconds long. The minimum and maximum values of toll rate are set as \$0.1/mile and \$3/mile respectively. The network is simulated for 30 minutes with no initial congestion. Buildup of congestion is modeled using a downstream bottleneck located at the end of each network. The value functions in the `SparseV` algorithm are initialized to an upper bound revenue.

Fig. 4(a)-(d) show the results for the DESE network for 30 minutes of simulation. Fig. 4(a) shows the moving average revenue with iterations. We observe that the convergence of the `SparseV` method is not guaranteed for longer time horizons. This can be explained by the graph in Fig. 4(b) which shows the number of new states visited in each iteration. The graph starts flattening out towards the later half of the simulation at a value around 100, that is, the `SparseV` method is still exploring an average of 100 new states in the later iterations across both agents. Convergence can be expected when new states are not explored and instead, the values of the older states are updated. Nevertheless, in the process of iterating the `SparseV` method, the toll profiles which led to the highest revenue are shown in Fig. 4(c) and (d) for agents 1 and 2, respectively.

Table II compares the best revenue obtained from the simulated policies. As observed, the revenues generated by the `Density` and `Ratio` heuristics are $70-75\%$ percent lower than than of the `SparseV` algorithm. The `SparseV` generates revenue which is 9.42% lower than the best revenue obtained by the `Random` algorithm; however it only takes 3 minutes of simulation time on a 2.8Ghz 64-bit Windows machine, in contrast to the 8 minutes of computation for the `Random` policy. We also observe that the `Density` and `Ratio` based heuristics lead to an average of 37% violations (defined as the proportion of the simulation

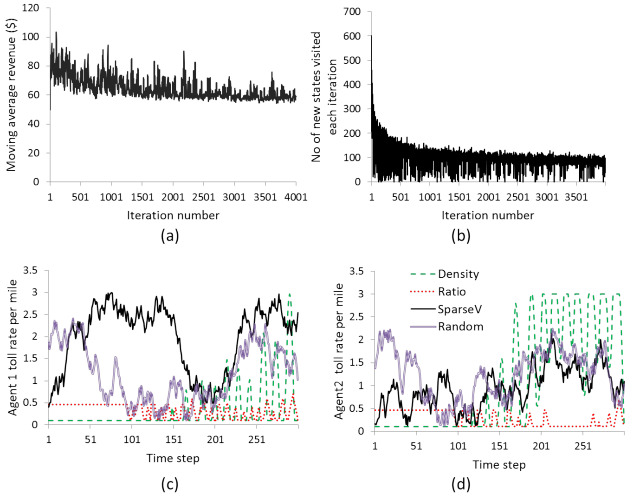

Fig. 4. Tests on DESE network. (a) Converge rate of the `SparseV` method with iterations; (b) Number of new states explored each iterations; Agent 1 (c) and Agent 2 (c) toll rate with time

time period the managed lane is congested) on the managed lane where additional 605 and 661 vehicles are let onto the managed lane causing the speed in the lanes to fall below the free flow speed. The best toll policy from `SparseV` only causes 5% violations.

TABLE II
COMPARISON OF REVENUES ACROSS DIFFERENT ALGORITHMS FOR DESE AND LBJ NETWORK

DESE network			
Algo.	Max. revenue	% Violations	Extra vehicles on ML
`Density`	38.32	37.7%	605.00
`Ratio`	33.09	36.6%	661.00
`Random`	146.00	0	0
`SparseV`	132.25	5%	38
LBJ network			
Algo.	Max. revenue	% Violations	Extra vehicles on ML
`Density`	125.21	32.33%	1313.00
`Ratio`	109.63	32.0%	3358.00
`Random`	602.38	0%	0
`SparseV`	795.32	0%	0

Fig. 5 and Table II show the performance of the four algorithms on the LBJ test network. As observed, the `SparseV` algorithm's best toll policy generates 24.3% more revenue than the best policy generated by the `Random` method, and $75-86\%$ more revenue than the `Density` and `Ratio` heuristics. The `Density` and `Ratio` heuristics continue to perform worse due to their inability to coordinate the tolls between agents.

Better performance of the toll policies generated by `Random` and `SparseV` algorithms can be explained by the jam-and-harvest nature of the optimal policies [3], [13]. For the DESE network, as shown in Fig. 4(c) and (d), the `SparseV` and `Random` policies charge higher toll in the earlier time steps to let the GP lanes become congested ("jam") and then continue charging higher toll rate in the later time steps to obtain more revenue when there is higher

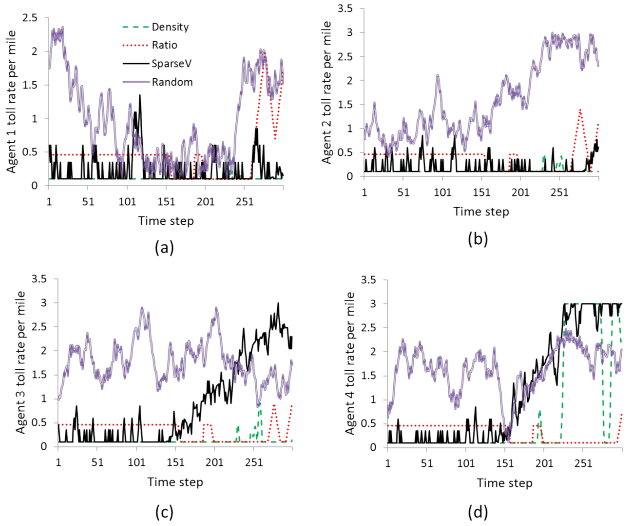

Fig. 5. Toll profiles for the 4 agents compared for each of the four algorithms

demand trying to enter the facility ("harvest"). This behavior of the optimal revenue policies is a characteristic of our model and can be avoided in practice using regulations on toll changes, studying which will be a part of our future work.

Similarly for the LBJ network, the `SparseV` algorithm strategically charges lower toll for agent 2 at earlier time steps that diverts more vehicles towards the managed lane, and once these vehicles arrive the diverge point for agent 3, they are faced with a higher travel time savings on the managed lane because the GP lane is congested due to the spillback from the downstream bottleneck. Thus, agent 3 can charge higher toll in later time steps to generate higher revenue. The same is true for agent 4 charging higher toll towards the later half of the simulation.

Overall, the results show that the `SparseV` method is successful in predicting better policies than the other heuristics used in practice, though the policies may not converge to optimal and may exhibit undesirable characteristics like "jam-and-harvest".

V. Conclusion

In this article, we proposed a multiagent reinforcement learning algorithm for the dynamic pricing of managed lanes with multiple entrances and exits where each agent regulates its own toll and coordinates with other agents to optimize the system performance. For the revenue maximization objective, we propose the `SparseV` algorithm which solves the joint optimal action by exploring the continuous action space.

Our experiments on two test networks show promising results. The `SparseV` algorithm performed better than the `Density` and `Ratio` heuristics by generating revenues $70\% - 86\%$ higher than those heuristics. `SparseV` also did comparably well to the `Random` heuristic, producing revenues within $9 - 20\%$ of the heuristic. `SparseV` has an advantage over the `Random` heuristic that it takes less computation time and does not enumerate toll policies, but rather builds on a MDP structure.

Though the `SparseV` method shows promising results, it has several limitations which need to be addressed. The first is the issue of convergence where the algorithm continues to oscillate heavily. This issue is inbuilt in all Q-learning based algorithms because they depend heavily on the Q or value functions initialization. Second, the algorithm is shown to converge to values which are suboptimal. This lack of convergence to the optimal depends on the aggregation level used for the state space, which we will improve in the future work. Third, the "jam-and-harvest" nature of the optimal policies is not desired in practice and thus constraints on toll policies to prohibit this nature will be modeled.

Acknowledgment

Partial support for this research was provided by the Data-Supported Transportation Operations and Planning University Transportation Center and the National Science Foundation Grant No. 1254921. The authors are grateful for this support.

References

[1] F. Zhu and S. V. Ukkusuri, "A reinforcement learning approach for distance-based dynamic tolling in the stochastic network environment," *Journal of Advanced Transportation*, vol. 49, no. 2, pp. 247–266, 2015.

[2] L. Yang, R. Saigal, and H. Zhou, "Distance-based dynamic pricing strategy for managed toll lanes," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2283, pp. 90–99, 2012.

[3] V. Pandey and S. D. Boyles, "Dynamic pricing for managed lanes with multiple entrances and exits," in *97th Annual Meeting of Transportation Research Board*, 2018.

[4] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man, and Cybernetics, Part C*, vol. 38, no. 2, pp. 156–172, 2008.

[5] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Springer, 2016, pp. 47–66.

[6] K. Rezaee, "Decentralized coordinated optimal ramp metering using multi-agent reinforcement learning," Ph.D. dissertation, University of Toronto (Canada), 2014.

[7] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[8] F. Zhu and S. V. Ukkusuri, "Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach," *Transportation research part C: emerging technologies*, vol. 41, pp. 30–47, 2014.

[9] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[10] J. R. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, vol. 7, no. Sep, pp. 1789–1828, 2006.

[11] C. F. Daganzo, "The cell transmission model, part II: network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995.

[12] C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," in *ICML*, vol. 2, 2002, pp. 227–234.

[13] C. Göçmen, R. Phillips, and G. van Ryzin, "Revenue maximizing dynamic tolls for managed lanes: A simulation study," Tech. Rep., 2015.