2011 14th International IEEE Conference on
Intelligent Transportation Systems
Washington, DC, USA. October 5-7, 2011

Reinforcement Learning with Average Cost for Adaptive Control of Traffic Lights at Intersections

Prashanth L A[†] and Shalabh Bhatnagar[⋆], *Senior Member, IEEE*

***Abstract*— We propose for the first time two reinforcement learning algorithms with function approximation for average cost adaptive control of traffic lights. One of these algorithms is a version of Q-learning with function approximation while the other is a policy gradient actor-critic algorithm that incorporates multi-timescale stochastic approximation. We show performance comparisons on various network settings of these algorithms with a range of fixed timing algorithms, as well as a Q-learning algorithm with full state representation that we also implement. We observe that whereas (as expected) on a two-junction corridor, the full state representation algorithm shows the best results, this algorithm is not implementable on larger road networks. The algorithm PG-AC-TLC that we propose is seen to show the best overall performance.**

***Index Terms*— Traffic signal control, reinforcement learning, Q-learning, policy gradient actor-critic.**

I. INTRODUCTION

Traffic signal control forms a crucial component of any intelligent transportation system. Designing an adaptive traffic signal control algorithm that maximizes the long-term traffic flow is an extremely challenging problem, considering the fact that a model of the system is not available in most of the real traffic environments. A second handicap to the traffic light control (TLC) algorithm is that the critical inputs - queue lengths and/or elapsed times (since signal turned red) on the lanes of the road network - are hard to obtain precisely in realistic settings and the TLC algorithm has to work with coarse estimates of these inputs. It is thus necessary to develop a TLC algorithm that minimizes a long-term cost objective using the coarse inputs of queue lengths and/or elapsed times and without assuming a system model. The TLC algorithm should be online, computationally efficient and possess the necessary convergence properties.

The problem that we consider in this paper is one of designing TLC algorithms that minimize a long term average cost objective, while possessing the properties outlined above. Reinforcement learning (RL) is an efficient technique for developing model-free algorithms that minimize a long-term cost objective based on sample observations from simulations. RL based TLC algorithms that minimize a long run discounted cost have been proposed in literature, for instance, in [1] and [2]. However, to the best of our knowledge, we are the first to design RL-based TLC algorithms that minimize a long-run "average cost" criterion. The motivation behind using an infinite horizon average cost framework is to understand the steady state behaviour of the traffic control system. We develop two new TLC algorithms, one based on Q-learning and the other a policy gradient actor critic algorithm for solving the traffic signal control problem in an average cost setting. While the Q-learning based TLC algorithms ([1], [2]) proposed for discounted cost problem are stochastic approximation analogues of the value iteration algorithm, they do not extend easily to the average cost setting. The Q-learning based TLC for average cost that we develop here is based on relative Q-value iteration scheme and has been adapted from [3]. The second TLC algorithm that we develop is a two-timescale actor critic algorithm that incorporates policy gradient for the actor recursion and temporal difference learning for the critic.

We now review some TLC algorithms previously proposed in the literature. Off-line techniques for traffic signal control have been proposed, for instance in [4]. The signal timings in these are generated off-line, using for example a static optimizer (see [4]), and the traffic light controllers at the intersections are programmed accordingly. In [5], the authors develop an optimization model of the traffic signal control problem. Genetic algorithm based approaches have been proposed, for instance in [6], [7]. These are heuristic techniques to solve the traffic signal optimization problem. Some approaches based on neural networks have been proposed, for instance in [8], [9]. They use SPSA gradient estimates in a neural network (NN) feedback controller to optimize traffic signal timings. A distributed multi-agent model for traffic signal control is presented in [10]. Markov decision process (MDP) based approach for traffic light control has been proposed in [11]. This approach however requires a precise model of the system, which in general is hard to obtain in real systems. An adaptive traffic light control algorithm that uses approximate dynamic programming is proposed in [12]. Reinforcement learning with full state representation has been proposed in [1]. However, only the case of an isolated traffic junction is considered there. In [2], an RL algorithm with function approximation together with certain graded-feedback policies is proposed. This algorithm is seen to perform well over many road traffic network scenarios involving several junctions. However, this algorithm is again for the discounted cost setting. Unlike RL algorithms based on full-state representations, the computational effort required by the algorithm in [2] remains reasonable even for large scale networks because of the use of function approximation. We develop in this paper, the first RL based TLC algorithms with function approximation that minimize a long term average cost objective. Whereas a discounted cost objective is more

[†]Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India. E-Mail:prashanth@csa.iisc.ernet.in

[⋆]Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India. E-Mail: shalabh@csa.iisc.ernet.in.

978-1-4577-2197-7/11/$26.00 ©2011 IEEE

suitable for optimizing short term performance, the long run-average cost objective assigns equal weightage to all stages and is concerned with the steady-state system behaviour.

A. Our Contributions

- We develop, for the first time, two reinforcement learning algorithms with function approximation for the problem of average cost traffic signal control.
- Our first algorithm is a Q-learning based TLC algorithm and is the function approximation analogue of the Q-learning with average cost algorithm proposed in [3].
- The second algorithm is a policy gradient actor critic based TLC algorithm. This algorithm is also shown to converge to the optimal sign configuration policy that minimizes the long run average cost.
- Both our algorithms require only coarse information on the level of congestion (for instance, low, medium or high) and do not require precise queue length information. This in unlike the algorithm of [1] that requires precise queue length estimates.
- We study the performance of our TLC algorithms in the context of a two-junction corridor, a five-junction corridor and a 2x2-grid network. From the performance comparisons of our algorithms between themselves and with a range of fixed timing TLC algorithms, we observe that the policy gradient actor critic based TLC algorithm performs the best, while also converging rapidly to the optimal sign configuration policy.

The rest of the paper is organized as follows: In Section II, we describe in detail the problem framework. In Section III, we present our learning algorithms with average cost for traffic light control. In Section IV, we discuss the implementation of the various TLC algorithms and present the performance simulation results. Finally, in Section V we provide the concluding remarks.

II. The Traffic Signal Control Problem

We consider the problem of finding an optimal schedule for the sign configuration of a traffic junction, with the aim of maximizing the traffic flow. The sign configuration here refers to the signals associated with a phase i.e., those that can be switched to green simultaneously. The traffic junction controller is assumed to periodically receive information about congestion (on individual lanes) through an array of sensors that are assumed embedded in the roads leading to the junction. In essence, the problem is to tune the sign configuration policy to the optimum that minimizes a certain long term average cost objective.

The traffic light control (TLC) algorithm uses as input - coarse estimates of the queue lengths along the individual lanes leading to the intersection and the time elapsed since the last signal light switch over. The queue length input is considered to minimize the average junction waiting times of the road users, while the elapsed time input ensures fairness i.e., no lane is allowed to stay green for a long time at the cost of other lanes.

We formulate this problem in the MDP setting and develop two TLC algorithms - one based on Q-learning and the other based on policy gradient actor-critic. An MDP framework [13] requires the identification of states, actions and costs, that we present below for our problem.

The state is the vector of queue lengths and the elapsed times since the signal turned red on those lanes that have a traffic signal at the various junctions in the road network. We assume centralized control for this purpose where the controller receives this information from the various lanes and makes decision on which traffic lights to switch green during a cycle. We do not assume that perfect information (on the queue lengths and signal timings) is available to the controller. Instead, the TLC algorithm that we subsequently present, work with coarse estimates of congestion levels on the various lanes of the road network. The controller's decision on which lights to switch green during a cycle is relayed back to the individual TLCs. We assume no propagation and feedback delays for simplicity. The elapsed time counter for a lane with green signal stays at zero till the time the signal turns red. For a road network with m junctions and a total of K signalled lanes across junctions, the state at time n is

$$s_n = (q_1(n), \ldots, q_K(n), t_1(n), \ldots, t_K(n)), \qquad (1)$$

where $q_i(n)$ is the queue length on lane i at time n and $t_i(n)$ is the elapsed time for the red signal on lane i at time n.

The actions comprise the sign configurations across junctions and have the form: $a_n = (a_1(n), \ldots, a_m(n))$, where $a_i(n)$ is the sign configuration at junction i in the time slot n and m is the number of junctions in the road network.

As with [2], the cost function here is designed to ease traffic congestion by minimizing the waiting queue lengths and at the same time, ensuring fairness so that no lane suffers being red for a long duration. This is achieved by letting the cost function be the sum of queue lengths and elapsed times on the lanes of the road network. Further, prioritization of traffic i.e., giving more importance to traffic on main roads than on the side roads is also incorporated into the cost function. Let I_p denote the set of indices for lanes whose traffic should be given higher priority. Then the cost $c(s_n, a_n)$ has the form

$$\begin{aligned} & c(s_n, a_n) \overset{\triangle}{=} c_{n+1} = \\ & r_1 * (\textstyle\sum_{i \in I_p} r_2 * q_i(n) + \sum_{i \notin I_p} s_2 * q_i(n)) \\ & + s_1 * (\textstyle\sum_{i \in I_p} r_2 * t_i(n) + \sum_{i \notin I_p} s_2 * t_i(n)), \end{aligned} \qquad (2)$$

where $r_i, s_i \geq 0$ and $r_i + s_i = 1, i = 1, 2$. Further, $r_2 > s_2$. Thus, lanes in I_p are assigned a higher cost and hence a cost optimizing strategy must assign a higher priority to these lanes in order to minimize the overall cost.

III. Our TLC Algorithms

The MDP for the problem of traffic signal control corresponds to the sequence $\{s_n\}$ with which is associated the control sequence $\{a_n\}$. Let $p(i, j, a)$ denote the transition probability of the MDP for transiting from state i

to j under action a. These probabilities satisfy $p(i,j,a) \in [0,1] \quad \forall i,j \in \mathcal{S}, a \in \mathcal{A}(i)$ and that $\sum_{j\in S} p(i,j,a) = 1$, for any given $i \in \mathcal{S}$ and $a \in \mathcal{A}(i)$. We consider an infinite horizon average cost framework. Our aim is to find a sequence $\{a_n\}$ of actions so as to minimize the "average cost"

$$\hat{\lambda} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} c(s_n, a_n) \triangleq \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} c_{n+1}, \qquad (3)$$

starting from any given state i (i.e., $s_0 = i$).

Let $h(i)$ be the differential cost function corresponding to state i. Then $h(.)$ satisfies

$$\lambda + h(i) = \min_a \sum_j p(i,a,j)(c(i,a) + h(j)), \forall i \in \mathcal{S}, \quad (4)$$

where λ is the optimal cost.

Define the Q-factors $Q(i,a), i \in \mathcal{S}, a \in \mathcal{A}(i)$ as

$$Q(i,a) = \sum_j p(i,a,j)(c(i,a) + h(j)). \qquad (5)$$

The Q-factors then satisfy the Bellman equation of optimality

$$\lambda + Q(i,a) = \sum_j p(i,a,j)(c(i,a) + \min_{b\in\mathcal{A}(j)} Q(j,b)), \quad (6)$$

for all $i \in \mathcal{S}, a \in \mathcal{A}(i)$.

Note that in order to solve this equation, one requires the knowledge of the transition probabilities $p(i,a,j)$ that constitute the system model. Moreover, the state and action spaces should be manageable in size.

The Q-learning algorithm with full state representation (that we present next) addresses the case when the system model is not known, however, state and action spaces are manageable. Further, our next two algorithms viz., Q-learning with function approximation and policy gradient actor-critic in addition to considering the case of 'lack of system model', also effectively handle large state and action spaces.

A. Q-learning with full state representations (QTLC-FS-AC)

Our QTLC-FS-AC algorithm estimates the 'Q-factors' $Q(i,a)$ of all feasible state-action tuples (i,a) i.e., those with $i \in \mathcal{S}$ and $a \in \mathcal{A}(i)$ using the relative value iteration of Q-factors and has been adapted from [3]. Let $s_{n+1}(i,a)$ denote the state of the system at instant $(n+1)$ when the state at instant n is i and action chosen is a. Let $Q_n(i,a)$ denote the Q-value estimate at instant n associated with the tuple (i,a). The relative Q-value iteration (RQVI) scheme is

$$\begin{aligned} Q_{n+1}(i,a) = \;& \textstyle\sum_j p(i,a,j)(c_{n+1} + \min_{b\in\mathcal{A}(j)} Q_n(j,b)) \\ & - \min_{r\in\mathcal{A}(s)} Q_n(s,r). \end{aligned} \qquad (7)$$

The QTLC-FS-AC algorithm is a stochastic approximation analogue of the RQVI and updates according to

$$\begin{aligned} Q_{n+1}(i,a) = \;& Q_n(i,a) + \alpha(n)(c_{n+1} + \min_{r\in\mathcal{A}(j)} Q_n(j,r) \\ & - \min_{b\in\mathcal{A}(s)} Q_n(s,b)), \end{aligned} \qquad (8)$$

for all feasible (i,a) tuples. Here $\alpha(n), n \geq 0$ are the step-sizes that satisfy $\sum_n \alpha(n) = \infty$ and $\alpha^2(n) < \infty$. Upon convergence, one obtains the optimal Q-values $Q^*(i,a)$ that are seen to satisfy the Bellman equation (6) and $\min_{a\in\mathcal{A}(i)} Q_n(i,a)$ gives the optimal differential cost $h^*(i)$. The optimal action in state i then corresponds to $arg\min_{b\in\mathcal{A}(i)} Q^*(i,b)$. A convergence proof of this algorithm can be found in [3].

B. Q-learning with function approximation (QTLC-FA-AC)

While the QTLC-FS-AC algorithm is useful in small state and action spaces, it becomes computationally expensive for larger road networks involving multiple junctions. This is because of the exponential increase in the sizes of the state and action spaces with the number of junctions. To alleviate this problem of curse of dimensionality, we incorporate feature based methods. These methods handle this problem by making computational complexity manageable. While the QTLC-FS-AC algorithm as such requires complete state information and so is less efficient, its function approximation based variant parametrizes the value function.

We now present QTLC-FA-AC, a Q-Learning based TLC that uses function approximation. Here we associate with each state-action tuple (i,a), a state-action feature denoted by $\phi_{i,a}$. The Q-function is then approximated as

$$Q^*(i,a) \approx \theta^T \phi_{i,a}. \qquad (9)$$

Let s_n, s_{n+1} denote the state at instants $n, n+1$, respectively, measured online. Let θ_n be the estimate of the parameter θ at instant n. Let s be any fixed state in $\mathcal{S}$.

The algorithm QTLC-FA-AC uses the following update rule:

$$\begin{aligned} \theta_{n+1} = \;& \theta_n + \alpha(n)\phi_{s_n,a_n}(c_{n+1} + \min_{v\in\mathcal{A}(s_{n+1})} \theta_n^T \phi_{s_{n+1},v} \\ & - \min_{r\in\mathcal{A}(s)} \theta_n^T \phi_{s,r}), \end{aligned} \qquad (10)$$

where θ_0 is set arbitrarily. In (10), the action a_n is chosen in state s_n according to $a_n = arg\min_{v\in\mathcal{A}(s_n)} \theta_n^T \phi_{s_n,v}$. For our experiments, we chose the following features (as in [2]):

$$\begin{aligned} \phi_{s_n,a_n} = \;& (\phi_{q_1(n)}, \ldots, \phi_{q_N(n)}, \phi_{t_1(n)}, \ldots, \phi_{t_N(n)}, \\ & \phi_{a_1(n)}, \ldots, \phi_{a_m(n)})^T \end{aligned} \qquad (11)$$

where for some given thresholds $L1, L2$ (on queue lengths) and $T1$ (on elapsed times),

$$\phi_{q_i(n)} = \begin{cases} 0 & \text{if } q_i(n) < L1 \\ 0.5 & \text{if } L1 \leq q_i(n) \leq L2 \\ 1 & \text{if } q_i(n) > L2 \end{cases}$$

$$\phi_{t_i(n)} = \begin{cases} 0 & \text{if } t_i(n) \leq T1 \\ 1 & \text{if } t_i(n) > T1 \end{cases}$$

Note that the parameter θ_n has dimension the same as that of ϕ_{s_n,a_n}. Again the advantage here is that instead of updating the Q-values for each feasible (s,a)-tuple as before, one estimates these according to the parametrization (9).

C. Policy Gradient Actor-Critic TLC (PG-AC-TLC)

Actor critic algorithms are reinforcement learning algorithms that are based on the policy iteration (PI) method for MDP. The classical PI algorithm proceeds via two loops

- the inner loop performs policy evaluation for a given policy while the outer loop performs policy improvement. Actor-critic algorithms use two-timescale stochastic approximation in order to perform both updates (evaluation and improvement) simultaneously at each instant. The difference in step-size schedules results in an appropriate algorithmic behaviour.

Our PG-AC-TLC algorithm incorporates policy gradients for the actor and temporal difference learning in the critic and has been adapted from [14]. The idea here is that the policy is considered to be parametrized, in addition to the value function. We consider a class of parametrized randomized policies and linear function approximation for the value function. Specifically, we assume that policies $\pi_\theta(i,a)$, parametrized by $\theta \in \Re^d$ have the form

$$\pi_\theta(i,a) = \frac{e^{\theta^\top \phi_{i,a}}}{\sum_{a' \in \mathcal{A}(i)} e^{\theta^\top \phi_{i,a'}}}, \ \forall s \in \mathcal{S} \ , \ \forall a \in \mathcal{A}, \tag{12}$$

where each $\phi_{i,a}$ is a d-dimensional feature vector as before.

Further, we let $V^\pi(i) \approx v^\top f_i, i \in \mathcal{S}$, to be the approximation to the differential cost function, where $f_i, i \in \mathcal{S}$ are $\hat{d}$-dimensional state features and v the corresponding parameter vector. Let $\psi_{ia} = \nabla \log \pi_\theta(i,a)$ denote the compatible state-action features. When $\pi_\theta(i,a)$ are selected as in (12), it can be seen that $\psi_{ia} = \phi_{i,a} - \sum_{a' \in \mathcal{A}(i)} \pi(i,a')\phi_{i,a'}$.

The PG-AC-TLC algorithm is as follows:

$$\hat{J}_{n+1} = (1-\alpha_n)\hat{J}_n + \alpha_n c_{n+1}, \tag{13}$$

$$\delta_n = c_{n+1} - \hat{J}_{n+1} + v_n^\top f_{s_{n+1}} - v_n^\top f_{s_n} \tag{14}$$

$$v_{n+1} = v_n + \alpha_n \delta_n f_{s_n}, \tag{15}$$

$$\theta_{n+1} = \Gamma(\theta_n + \beta_n \delta_n \psi_{s_n a_n}), \tag{16}$$

Here, $\{s_n\}$ is the sequence of states visited by the MDP i.e., we consider a single trajectory of states for the MDP. Further $\{a_n\}$ is the sequence of actions obtained upon following the randomized policy π. Here α_n and $\beta_n, n \geq 0$ are two step-size sequences that satisfy

$$\sum_n \alpha(n) = \sum_n \beta(n) = \infty; \sum_n (\alpha_n^2 + \beta_n^2) < \infty, \lim_{n\to\infty} \frac{\beta_n}{\alpha_n} = 0.$$

It has been shown in [14] that if one replaces δ_n by $\delta_n^\pi \stackrel{\triangle}{=} c_{n+1} - \hat{J}_{n+1} + v_n^{\pi^\top} f_{s_{n+1}} - v_n^{\pi^\top} f_{s_n}$, where $\lim_{n\to\infty} v_n = v^\pi$ (assuming fixed π), then

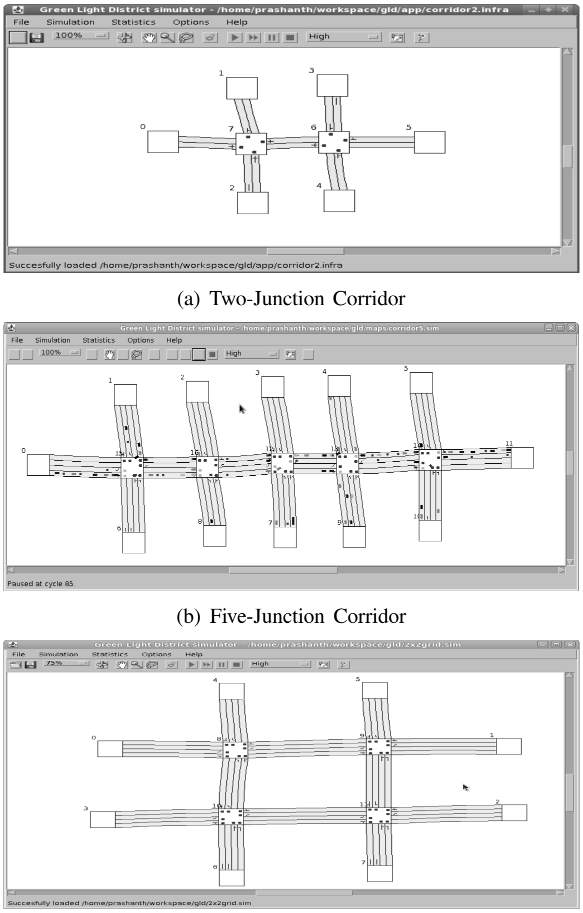
$$\mathbf{E}[\delta_n^\pi \psi_{s_n a_n} | \theta] = \nabla J(\theta) + \varepsilon(\theta),$$

where $\varepsilon(\theta)$ is an error term that arises from the use of linear function approximation. Further, it has been shown in [14] that if $||\varepsilon(\theta)||$ is "small", then $\theta_n, n \geq 0$ given by the recursions (13) - (16) converge asymptotically to a "small" neighbourhood of the local minima of $J(.)$.

IV. Simulation Experiments

A. Implementation

We use the Green Light District (GLD) simulator ([15]) for implementation and evaluation of our TLC algorithms. We implement our TLC algorithms - QTLC-FS-AC and QTLC-FA-AC and PG-AC-TLC, respectively. Recall that whereas QTLC-FS-AC incorporates full state representations, the other two algorithms viz., QTLC-FA-AC and PG-AC-TLC incorporate function approximation. For the sake of comparison, we also implement a range of fixed timing TLC algorithms that periodically cycle through the list of feasible sign configurations irrespective of the traffic conditions. The cycling period here is a tunable parameter and we show the results of the performance of these algorithms for various cycling periods.



Green Light District simulator - /home/prashanth/workspace/gld/app/corridor2.infra

Succesfully loaded /home/prashanth/workspace/gld/app/corridor2.infra

(a) Two-Junction Corridor

Green Light District simulator - /home/prashanth/workspace/gld/maps/corridor5.sim

Paused at cycle 85.

(b) Five-Junction Corridor

Green Light District simulator - /home/prashanth/workspace/gld/2x2grid.sim

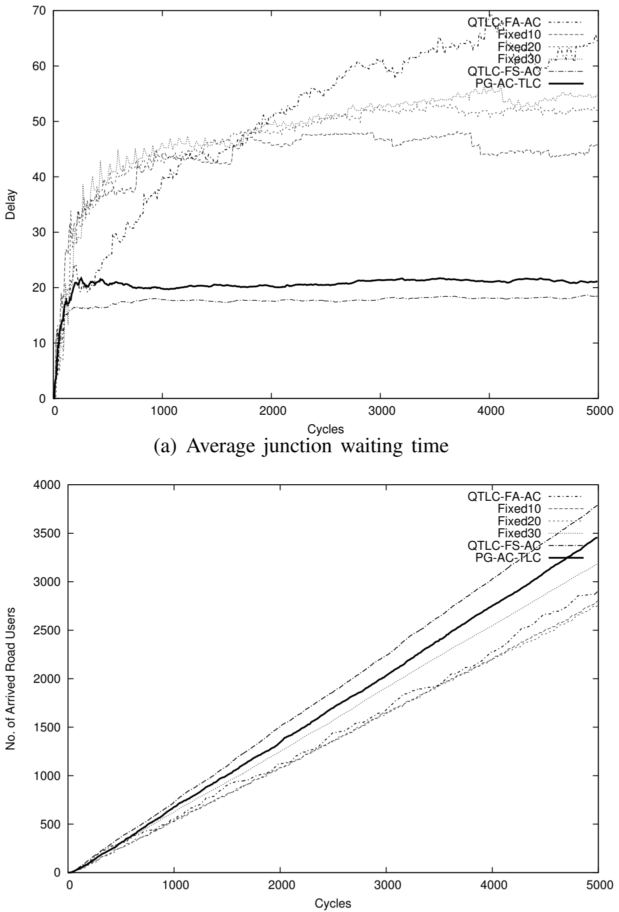
Succesfully loaded /home/prashanth/workspace/gld/2x2grid.sim

(c) 2x2-Grid Network

Fig. 1. Road Networks used for our Experiments

We consider three different network scenarios: a two-junction corridor, a five-junction corridor and a 2x2-grid network. We show snapshots of these networks obtained from the GLD software in Fig. 1. The simulations are conducted for 5000 cycles in all algorithms. Each road user's destination is fixed randomly, using a discrete uniform distribution, to choose one of the edge nodes. The spawn frequency i.e., the rate at which traffic is generated randomly in GLD, was set in a way that ensures that the proportion of cars flowing on the main road to those on the side roads is in the ratio 100:5.

The performance of QTLC-FS-AC is tested only on the two-junction corridor while that of QTLC-FA-AC and PG-AC-TLC is tested on all the settings in Fig. 1. QTLC-FS-AC could not be implemented on larger road networks because of the exponential blow up in computational complexity (described previously) with the numbers of lanes



(a) Average junction waiting time

(b) Total Arrived Road Users

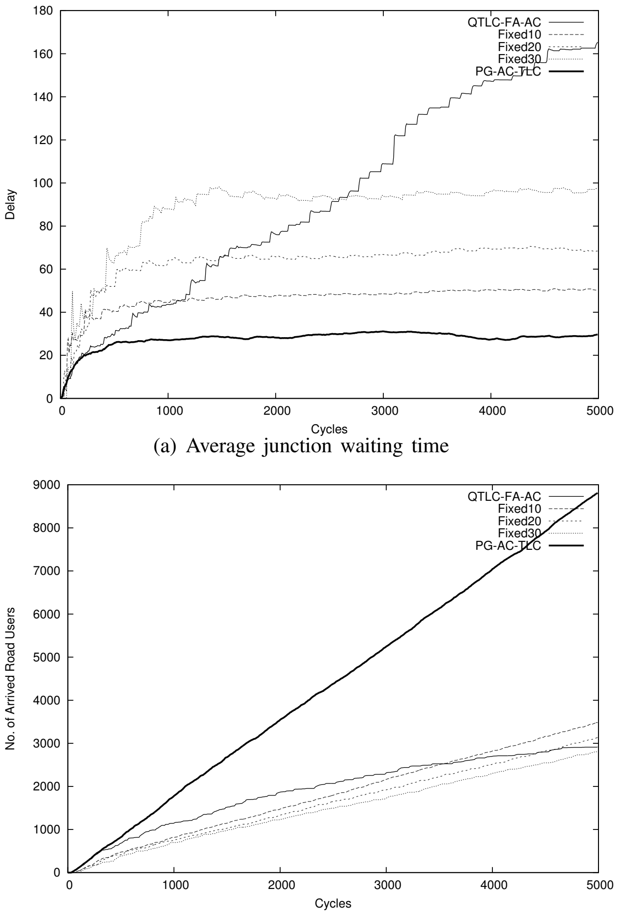
Fig. 2. Performance Comparison of TLC Algorithms - Two-Junction Corridor Case

and junctions (when full state representations are used). On the other hand, QTLC-FA-AC and PG-AC-TLC are easily implementable even on larger road networks, and require much less computation.

For all the three TLC algorithms, we set the weights in the single stage cost function $c(s, a)$ in (2) as $r_1 = s_1 = 0.5$ and $r_2 = 0.6, s_2 = 0.4$. This assignment gives a higher priority to the lanes on the main road than those on the side roads, while according equal weightage to both queue length and elapsed time components of the cost function.

B. Results

Figs. 2, 3 and 4 show the plots of average junction waiting time and total arrived road users using the various algorithms on the three road network settings. From the above plots, we observe that the policy gradient actor critic based TLC algorithm (PG-AC-TLC) shows the best overall performance as compared to the other TLC algorithms considered. In the case of the two-junction corridor, QTLC-FS-AC shows the best results. This is however expected because QTLC-FS-AC uses the knowledge of the full state whereas the PG-AC-TLC and QTLC-FA-AC use only coarse information on whether the level of congestion is in the low, medium or high range, and also whether the elapsed time is below or above a threshold.



(a) Average junction waiting time

(b) Total Arrived Road Users

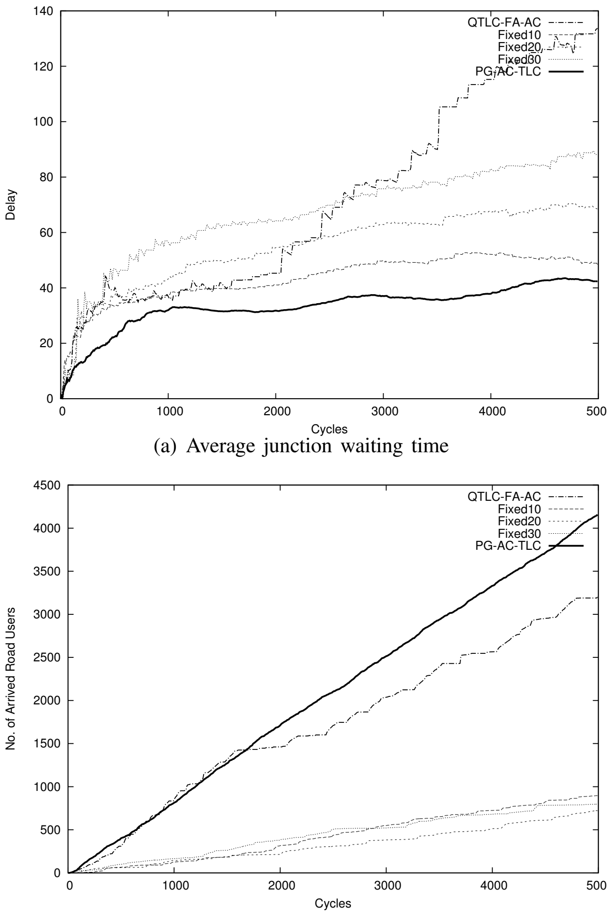
Fig. 3. Performance Comparison of TLC Algorithms - 2x2-Grid Network Case

We also observe that the parameter θ converges in the case of PG-AC-TLC algorithm, while the same is not true of QTLC-FA-AC algorithm. This is evident in Fig. 5 where the convergence of θ for PG-AC-TLC and oscillation of θ for QTLC-FA-AC are illustrated using one of the co-ordinates of θ for the case of a 2x2-grid network. This is because QTLC-FA-AC suffers from the off-policy problem [13] unlike PG-AC-TLC that does not suffer from this problem because of the use of multi-timescale stochastic approximation.

V. Conclusions and Future work

Our goal in this paper was to design an algorithm for average cost traffic signal control in order to optimize steady state system performance. We developed two reinforcement learning algorithms with average cost - one an analogue of Q-learning in a function approximation setting and another a policy gradient actor critic algorithm. From the performance comparisons, we observe that the policy gradient actor critic algorithm showed the best overall performance. On a two-junction corridor, the full state representation algorithm was better, however, it suffers from the limitation of not being implementable on larger road networks unlike algorithms that use function approximation.

In conclusion, we mention two important future work directions below:

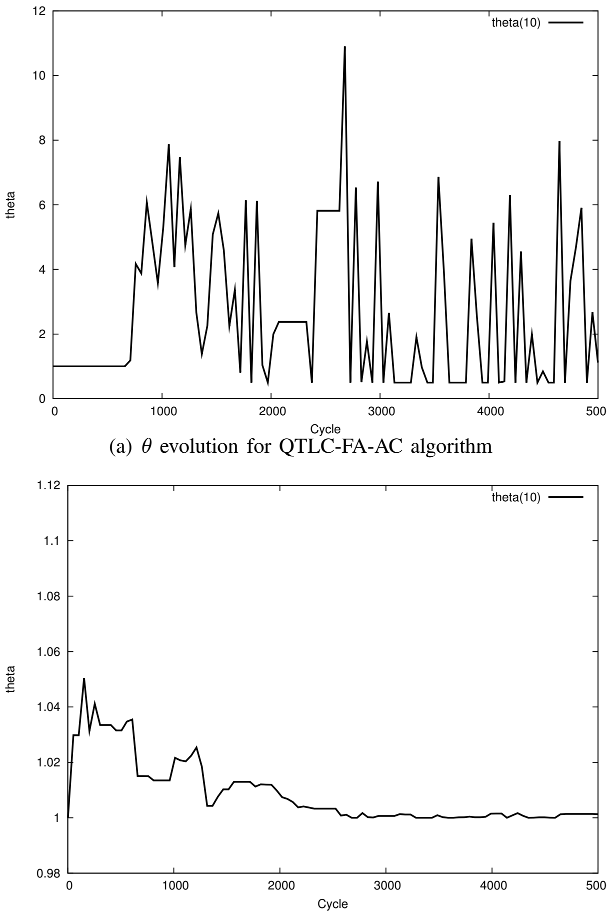


(a) Average junction waiting time

(b) Total Arrived Road Users

Fig. 4. Performance Comparison of TLC Algorithms - Five-Junction Corridor Case

- Apart from the policy gradient based actor-critic algorithms, there are other algorithms based on natural gradients proposed in [14]. It would be interesting to develop TLC algorithms that combine natural gradients and function approximation for the problem of average cost traffic control.
- We used arbitrarily set, fixed values for the thresholds $L1, L2, T1$ in our TLC algorithms. An interesting future research direction will be to develop a threshold tuning algorithm that combines with policy optimization in order to find an optimal policy via an optimal choice of thresholds.



(a) θ evolution for QTLC-FA-AC algorithm

(b) θ evolution for PG-AC-TLC algorithm

Fig. 5. Convergence of θ - illustration using one of the co-ordinates in the case of a 2x2-Grid Network

REFERENCES

[1] B. Abdulhai, R. Pringle, and G. Karakoulas, “Reinforcement learning for true adaptive traffic signal control,” *Journal of Transportation Engineering*, vol. 129, p. 278, 2003.

[2] Prashanth. L.A. and S. Bhatnagar, “Reinforcement Learning with Function Approximation for Traffic Signal Control,” *IEEE Transactions on Intelligent Transportation Systems (Accepted)*, 2010. doi: 10.1109/TITS.2010.2091408.

[3] J. Abounadi, D. Bertsekas, and V. Borkar, “Learning algorithms for Markov decision processes with average cost,” *SIAM Journal on Control and Optimization*, vol. 40, no. 3, pp. 681–698, 2002.

[4] D. Robertson, *TRANSYT: a traffic network study tool.* Road Research Laboratory Crowthorne, 1969.

[5] W. Lin and C. Wang, “An enhanced 0-1 mixed-integer LP formulation for traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 238–245, 2004.

[6] M. Girianna and R. Benekohal, “Using genetic algorithms to design signal coordination for oversaturated networks,” *Journal of Intelligent Transportation Systems*, vol. 8, no. 2, pp. 117–129, 2004.

[7] J. Sanchez-Medina, M. Galan-Moreno, and E. Rubiyo-Royo, “Traffic signal optimization in “La Almozara” district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 132–141, 2010.

[8] J. Spall and D. Chin, “Traffic-responsive signal timing for system-wide traffic control,” *Transportation Research Part C*, vol. 5, no. 3-4, pp. 153–163, 1997.

[9] D. Srinivasan, M. Choy, and R. Cheu, “Neural networks for real-time traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 261–272, 2006.

[10] B. Gokulan and D. Srinivasan, “Distributed geometric fuzzy multi-agent urban traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 714–727, 2010.

[11] X. Yu and W. Recker, “Stochastic adaptive control model for traffic signal systems,” *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 4, pp. 263–282, 2006.

[12] T. Li, D. Zhao, and J. Yi, “Adaptive dynamic programming for multi-intersections traffic signal intelligent control,” in *11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008*, 2008, pp. 286–291.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3).* Athena Scientific, May 1996.

[14] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee, “Natural actor-critic algorithms,” *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[15] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman, “Simulation and optimization of traffic in a city,” in *IEEE Intelligent Vehicles Symposium*, June 2004, pp. 453–458.