Adaptive Traffic Signal Control Using Multi-Agent Reinforcement Learning

Sonu Dixit

Indian Institute of Science

sonudixit@iisc.ac.in

Advisor - Prof. Shalabh Bhatnagar, CSA

June 17, 2019

Overview

Motivation

- Travel is an integral part of everyone's daily work. Traffic junctions play an important role in shaping city's traffic.
- Most traffic junctions use fixed time algorithms, or hand tuned algorithms.
- Adapt Signal timings according to Traffic Congestion with the following objectives:
  - Reduction in congestion
  - Reduce waiting time, travel time
  - Ease the traffic flow
  - Utilize existing infrastructure efficiently

Multi-Agent Reinforcement Learning(MARL) Framework

- RL :

  - Traffic System behaviour evolves through complex stochastic process.
  - RL approach is best suited for dynamic environment.
- Central Single Agent RL :
  - A single agent approach is computationally inefficient.
  - Not scalable due to large state and action space.
  - Delay in information sharing due to single central agent.
- Multi Agent RL with function Approximation:
  - Each junction is an RL Agent(State, Action, Rewards).
  - Each Agent observes local part of environment.

Definitions

- Markov Decision Process is a tuple $< S, A, P, R, \gamma >$.
- Policy π is a Distribution over actions given state.
- State value function $V^{\pi}(s_t) = E_{a_t, s_{t+1}, a_{t+1}, \ldots}[\sum_{l=0}^{l=\infty} \gamma^l r(s_{t+l}, a_{t+l})]$
- State Action value function $Q^{\pi}(s_t, a_t) = E_{s_{t+1}, a_{t+1}, \ldots}[\sum_{l=0}^{l=\infty} \gamma^l r(s_{t+l}, a_{t+l})]$
- Advantage : $A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$
- Objective : find π^*

$$\pi^* = \arg\max_{\pi_\theta} \sum_{t=0}^{t=\infty} E_{a_t \sim \pi_\theta(s_t), s_{t+1} \sim \rho} \gamma^t r(s_t, a_t, s_{t+1}) \tag{1}$$

where ρ is state transition probability.

Related Work

- Multi-Agent Q-learning [1] :
  - State : discrete occupancy at incoming lanes
  - Action $\in \{10sec, 20sec, 30sec\}$
  - Cost : Occupancy at neighbors incoming lanes

- Most Previous works have used Q-Learning for single intersection.
- Deep Q-Network [2]:
  - State as a boolean(Vehicle presence) Vector, Average Speed, current phase
  - Action : Change phase
  - Reward : Change in cumulative vehicle delay between actions.
- For state representation Authors [3, 4] have used combination of queue length, waiting time, speed, image representation, current phase.

Problem Formulation

Let $J = \{1, 2, ..N\}$ be set of junctions in the network. Define :

- $q^t_{i,j}$ =occupancy at incoming lane j of junction i at time t.
- $N(i)$ =set of immediate Neighbors of junction i, including i.
- $P(i)$ = set of incoming lanes at junction i.

MDP Framework

The **state** of junction i at time t is a $L+1$ dimensional vector.
$state(i)_t = \{q^t_{j,k} : \forall k \in P(j), \forall j \in N(i)\}$
Current signal phase is the last dimension of state vector.
Action space is a discrete action in $\{20,25,30,....,70\}$ seconds.
Reward received by agent i at time $t+a$

$$r^i(t+a) = \sum_{j=1}^{j=L} state(i)_{t,j} - \sum_{j=1}^{j=L} state(i)_{t+a,j} \tag{2}$$

where a is action taken by agent at time t.

Algorithm

Proximal Policy Optimization (PPO) [5] Advantage Actor Critic

- Let $r_t(\theta)$ denote the probability ratio, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta\, old}(a_t|s_t)}$.
- Objective for policy gradient methods is $max \hat{E}_t(r_t(\theta)\hat{A}_t)$
- PPO is an optimization technique for policy gradient methods, that enforces a condition which limits change in policy during gradient update.

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{3}$$

$$clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon) = \begin{cases} 1+\varepsilon, & \text{if } r_t(\theta) \geq 1+\varepsilon \\ 1-\varepsilon, & \text{if } r_t(\theta) \leq 1-\varepsilon \\ r_t(\theta), & \text{otherwise} \end{cases} \tag{4}$$

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms

PPO Advantage Actor Critic

- Entropy :

$$H(\theta) = -\sum_{a_t} \pi_\theta(a_t|s_t) log \pi_\theta(a_t|s_t) \tag{5}$$

- Final Objective for Actor :

$$max(L^{CLIP}(\theta) + \beta H(\theta)) \tag{6}$$

- For Critic :

$$y_t = \sum_{l=0}^{l=T-1} \gamma^{t+l} r_{t+l} + V_\phi(s_{t+T}) \tag{7}$$

$$min(V_\phi(s_t) - y_t)^2 \tag{8}$$

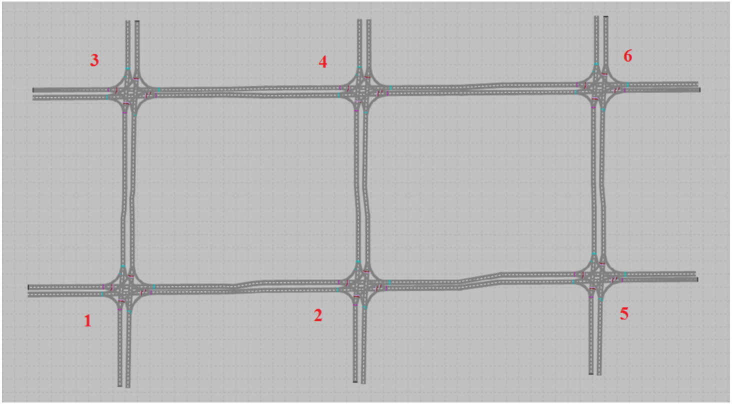
$$\hat{A}(s_t, a_t) = y_t - V_\phi(s_t) \tag{9}$$

Full Algorithm

Algorithm 1 Multi Agent Training (PPO Actor Critic)

Load network file to VISSIM simulation software
Initialize every junction as agent($\pi_{old}, \pi_{new}, V_\phi$)
for `time in range(totalTime)` **do**
    run single step of VISSIM simulation
    for `agents that need to take action` **do**
        get $s_t, a_t, r_{t+a_t}, s_{t+a_t}$, put it into buffer
        if buffer has enough samples **then**
            Estimate Advantage using (9)
            for `j in range(K)` **do**
                sample minibatch from buffer
                train π_{new} using (6) ▷ actor training
            end for
            $\pi_{old} \leftarrow \pi_{new}$
            train V_ϕ using (8) ▷ critic training
        end if
        take action $a \sim \pi_{new}(s_{t+a_t})$ ▷ Sample action
    end for
end for

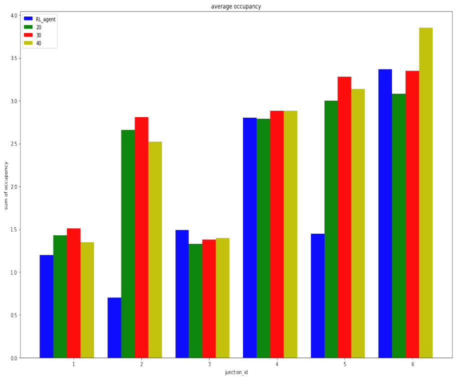
Traffic Network for Experimentation

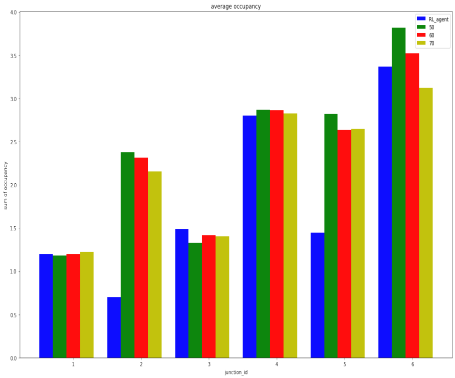


- PTV VISSIM microscopic simulation software
- 10 vehicle input points, vehicle incoming follows Poisson process, we set parameter(λ)
- 8 incoming lanes(signals, queue counters) at each junction

Experimentation and Results

- Fixed Vehicle incoming rate: 2000 vehicles/hour at each input
- Mean Congestion Comparison





- Summing up differences at all the junctions, our MARL algorithm is performing better than fixed time algorithms.

Experimentation and Results

- Fixed Vehicle incoming rate: 2000 vehicles/hour at each input
- Average Speed Comparison, calculated at the gap of 1000 seconds

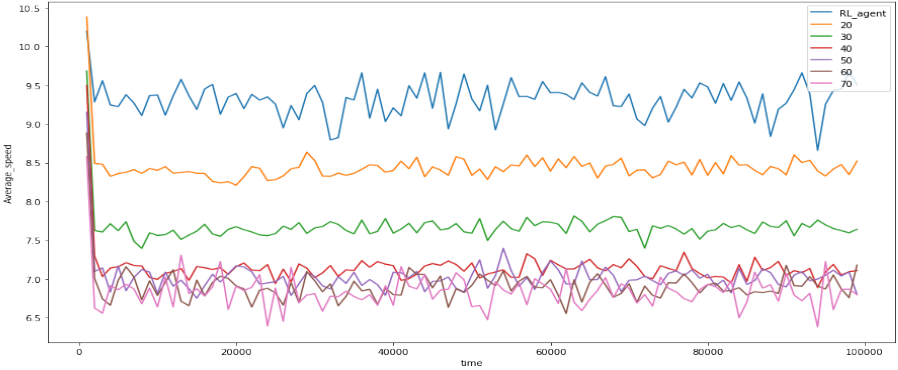


Figure: MARL algorithm has higher average speed(Blue line) than any fixed time algorithms

Experimentation and Results

- Fixed Vehicle incoming rate: 2000 vehicles/hour at each input
- Average Delay Comparison, calculated at the gap of 1000 seconds

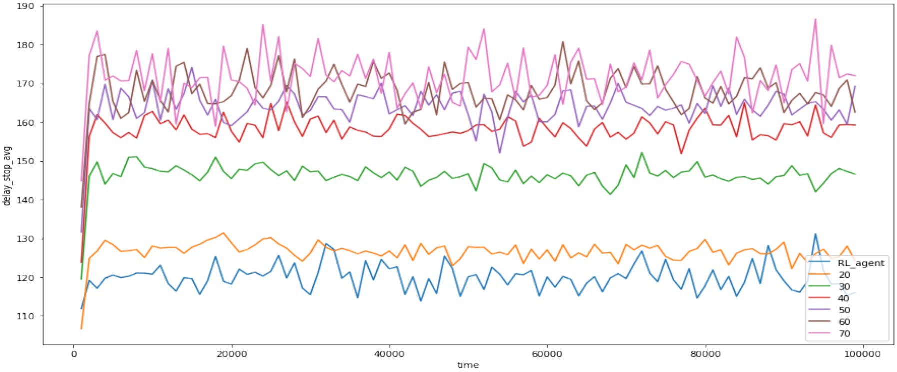
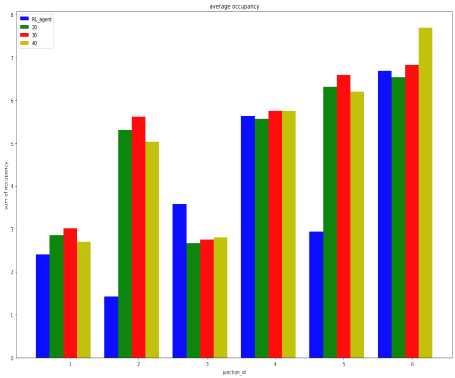
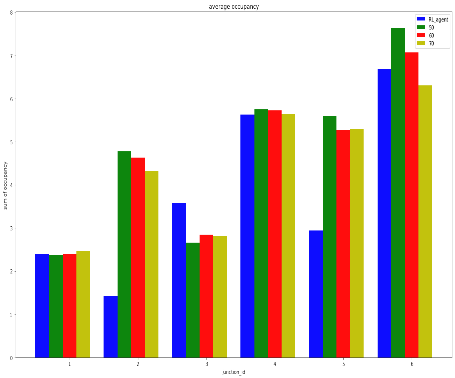


Figure: MARL algorithm has lower average delay (Blue line) time than any fixed time algorithms

Experimentation and Results

- Randomly changing incoming rate at each input in every 50 minutes
- Mean Congestion Comparison





- Summing up differences at all the junctions, our MARL algorithm is performing better than fixed time algorithms.

Experimentation and Results

- Randomly changing incoming rate at each input in every 50 minutes
- Average Speed Comparison, calculated at the gap of 1000 seconds

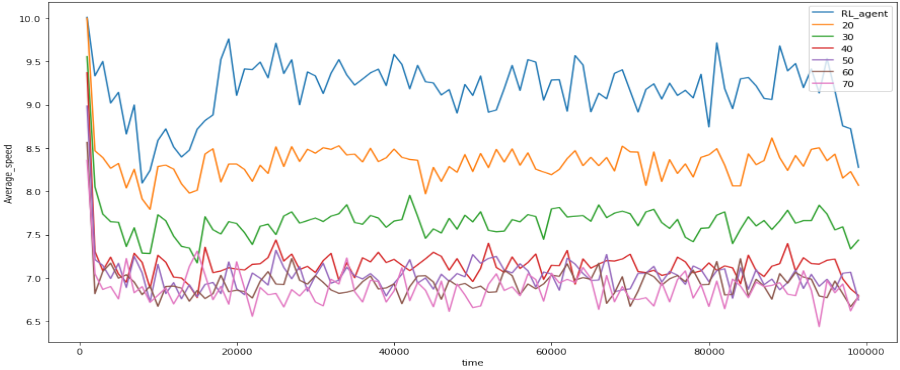


Figure: MARL algorithm has higher average speed(Blue line) than any fixed time algorithms

- Randomly changing incoming rate at each input in every 50 minutes
- Average Delay Comparison, calculated at the gap of 1000 seconds

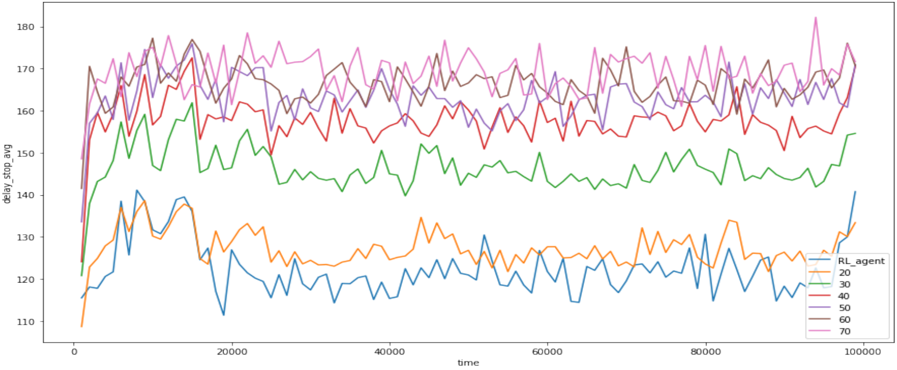


Figure: MARL algorithm has lower average delay(Blue line) than any fixed time algorithms

Conclusion

- Agents' got positive reward when they decreased congestion, so their primary objective was to decrease congestion.
- MARL algorithm has less queue occupancy than any other fixed time algorithms.
- Decrease in congestion led to :
  - Decrease in Average Delay
  - Increase in Average Speed
- Scalable, Distributed approach
- Easily Deployable in Real Scenario:
  - Occupancy data can be extracted from a image processing model
  - Detectors can be put at some distance gap from stop line
  - Wireless Communication between each neighbors

Future Work

- Comparison with other algorithms Sydney Coordinated Adaptive Traffic System (SCATS) [6], Split Cycle Offset Optimization Technique (SCOOT) [7]
- Theoretical convergence proof in case of Multi Agent PPO Actor Critic

[6] A. Sims and K. Dobinson, The sydney coordinated adaptive traffic (scat) system philosophy and benefits,

[7] P. Hunt, D. Robertson, and R. Bretherton, The scoot on-line traffic signal optimisation technique (glasgow)

References I

P. K.J., H. K. A.N, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2014, pp. 2529–2534.

W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *CoRR*, vol. abs/1611.01142, 2016. [Online]. Available: http://arxiv.org/abs/1611.01142

H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 2496–2505. [Online]. Available: http://doi.acm.org/10.1145/3219819.3220096

L. Li, L. Yisheng, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *ACTA AUTOMATICA SINICA*, vol. 3, pp. 247–254, 06 2016.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

References II

A. Sims and K. Dobinson, "The sydney coordinated adaptive traffic (scat) system philosophy and benefits," *Vehicular Technology, IEEE Transactions on*, vol. 29, pp. 130 – 137, 06 1980.

P. Hunt, D. Robertson, and R. Bretherton, "The scoot on-line traffic signal optimisation technique (glasgow)." *Traffic Engineering Control*, vol. 23, pp. 190–192, 01 1982.

Thank You