Grenze International Journal of Engineering and Technology, June Issue

Adaptive Traffic Signal Optimization using Reinforcement Learning: A SUMO-based Study

Anushka Kalbande[1] and Mohammad Anwarul Siddique[2]
[1-2]Department of Computer Science and Engineering Anjuman College of Engineering and Technology Nagpur, India
Email: anushkakalbande28@gmail.com

***Abstract*— Traffic congestion which occurs in urban areas has become a severe issue within the contemporary world as the conventional time based traffic signal controllers are not responsive to the fluctuating issue of traffic on the basis of time. This paper shows a smart and adaptive traffic signal control framework based on the Reinforcement Learning (RL) applied in a Simulation of Urban Mobility (SUMO) simulator and Traffic Control Interface (TraCI). An actual single intersection case study with eastbound (EB) and southbound (SB) approaches is modeled using six lane-area (E1) detectors so that accurate lengths of queues of different distances can be measured. It methodically compares three controllers based on a fixed time base after traditional 30-30-second cycles, (1) fixed time simple controller, (2) tabular Q-learning controller and (3) Deep Q-Network (DQN) Neural network approximation modeling Q-values estimation. The RL agents maximize some formulation of reward that will penalize total queue length that is above 3000 simulation steps. In-depth analysis based on the metrics of traffic performance (vehicle delay total, average and maximum queue lengths, time of travel distributions (mean, 90th percentile, maximum) and vehicle throughput) shows that the learning based improvements were made by controllers. Findings indicate that the DQN controller only causes just about 40 % reduction in the overall delay and 12 % increase in throughput that it provides compared to the baseline which is static, and it also performs better than the tabular Q-learning in terms of stability and convergence speed. These results confirm deep reinforcement learning as a strong base in learning scalable adaptive traffic signal control and possible utilization in multi intersection networks.**

***Index Terms*— Reinforcement learning, Traffic signal control, SUMO Simulation Adaptive Traffic Control Intelligent Transportation Systems, Simulated Evaluation.**

I. INTRODUCTION

Rapid urbanization and economic development has put heavy stress on the urban transportation infrastructure. Urbanization has increased the urban density and ownership of private vehicles, which has increased the degree of demands at signalized intersections, and the commercial area has increased the demand for spatial concentration, which has also increased the stress at intersections, making it one of the network bottleneck areas. However, it was noticed that most signal control systems that are implemented utilize the fixed-time strategy that was proposed with average historical and relatively stable demands [1].And these static operations are inherently incapable of addressing real-time variations in traffic volume, thus promoting an inefficient utilization of green time, increased idling, fuel usage, and emission, while simultaneously presenting instable travel time characteristics [2].

Grenze ID: 01.GIJET.12.2.467_22

Traditional fixed-time controllers assume quasi-stationary and approximately balanced traffic flows. In real-world environments traffic demand is very stochastic and subject to significant variations depending on the time of the day, day of the week, the weather conditions or any unexpected disturbance such as an incident of any kind or a special event. As a result of this, vehicles often suffer unnecessary delay at red signals even when there are no conflicting traffic present. During non-peak times, signal timings that are optimized for peak demand make inefficiencies even greater. When these localized inefficiencies are propagated throughout intersections, upstream spillback congestion may result, seriously degrading performance on a corridor level.

To enhance the responsiveness, vehicle signal control systems were introduced, where the green phase can be expanded or expired depending on actual detector input (usually supplied by inductive loop sensors). While these systems are better than fixed-time control when faced with moderate variability, their decision making logic is generally rule based and based on preconfigured thresholds.This concept is extended by coordinating multiple intersections through centralized optimization , using some advance adaptive traffic control systems such as SCOOT (Split Cycle Offset Optimization Technique) [3] and SCATS (Sydney Coordinated Adaptive Traffic System) [4].Although we have seen such systems work in practice, they can be very demanding in calibration and are frequently based on proprietary implementations and are not very flexible in highly non-stationary or unpredictable traffic conditions.

Reinforcement Learning (RL) is used as an another approach [5],for controlling the traffic signal by learning through the direct interaction with the environment.Here a traffic signal controller will be used to receive the information related to the current state of the traffic.It will also take actions like maintaining or changing the signal phases, the controller also receives a reward signal which indicates the congestion levels. Through a repeated trial and error learning, the agent gradually improves the control strategy. This paradigm is especially well-suited to complex and time-varying traffic environments; however, for deployment to be effective, careful design of state representations, reward functions and learning hyperparameters is required, especially for deep reinforcement learning settings [8].

Initial studies utilising tabular Q-learning have demonstrated the potential for the utilization of reinforcement learning for adaptive traffic control under stochastic demand profiles. Further, the convergence characteristics associated with the util- isation of control strategies for traffic flow at intersections utilising the technique known as Average Cost Reinforcement Learning have been investigated in the study furnished in reference [6], and the utilisation of Multi-Agent Q-learning has been successfully applied to simulate urban traffic flow profiles and achieve enhanced performance, as furnished in reference [7]. Even though encouraging achievements have been reported utilising the tabular form of the algorithm, the technique suffers considerably due to limitations associated mainly with the curse of dimensionality. Deep Reinforce- ment Learning introduces the potential for addressing the aforesaid problems and for the utilisation of value functions which can be approximated with the help of neural networks.

A Deep Q-Network has been furnished and proposed in the study furnished in reference [8], which has attracted significant attention and has been utilised as a motivating framework for the formulation of a variety of traffic signal control strategies, including the proposed models for competition and pressure-based control strategies furnished in references [9] and [10], respectively. A cooperative optimization strategy for interrelated intersections has been furnished in reference [11] [14].

Regardless of any advances that have been made, some key research gaps include:

- Lack of standardized evaluation: The studies conducted so far uses a variety of network topologies, traffic demand models, detector placements, and training durations.The variability is significant such that direct comparison of findings from different research is difficult. This limits aspects of performance evaluation.
- Simplified state modeling: From a traffic engineering point of view, being aggregated or modelled to a phase of a signal reduces the resolution of the traffic representation, thereby failing to capture the nuances of spatiotemporal variations in the formation of traffic queues, as well as the potential for learning in complex scenarios.
- Limited performance metrics: Most evaluations describe average delay as a performance measure. Other significant metrics (and ones that are directly related to overall service quality and the management of congestion) such as maximum queue length, traffic throughput, and tail-end performance are still not frequently reported.
- Insufficient online learning analysis: Continuous online training via real time simulator interaction, particularly using TraCI-based control loops, has received comparatively limited systematic investigation.
- Predominant single-intersection scope: While multiagent frameworks have been proposed, controlled single-intersection studies remain essential for isolating learning dynamics and establishing reliable performance baselines.

To address the gaps mentioned above,a SUMO microscopic traffic simulator based controlled experiment framework will be build in this study.
Within this framework, three signal control approaches—a conventional fixed-time scheme, a tabular Q-learning controller, and a Deep Q-Network (DQN)–based method—are implemented and evaluated under identical network topology, traffic demand, and detector configurations.
The key contributions of this work are summarized below:

- Realistic state modeling: Six lane-area E1 detectors measure queue lengths at multiple spatial intervals for each approach, combined with the current signal phase to form a seven-dimensional state vector.
- Controlled algorithmic comparison: Fixed-time control, tabular Q-learning, and deep reinforcement learning are evaluated side-by-side using consistent simulation settings.
- Comprehensive performance evaluation: The evaluation takes into account several performance indicators such as total delay, average and maximum queue lengths, travel time-based statistics as well as the vehicle throughput.
- Online learning investigation: Agents are trained online through TraCI-based interaction with the simulator, en- abling analysis of convergence behavior and stability.

The organization of this paper is as follows. In section II the related work based on the traffic signal control and re- inforcement learning methods are reviewed.
In Section III the brief overview of the methodology is given. Section IV presents experimental results and comparative performance analysis. Finally, in section V the paper concludes and the future research directions are highlighted.

II. Related Work

The study of traffic signal control is a 60-year-old field of research that has undergone the metamorphosis of fixed analytical optimization to adaptive AI-based system control.The formula of Webster's [1], obtained out of queueing theory suggests cycle time and phase divisions under the assumption of constant and stationary demand. Variants incorporated walking density, movement along arterials, and coordination are restraint of pedestrian density, movement forward, and coordination trade-offs [2]. The methods cannot work dynamically, too much green time when it is not peaking will result in unneeded delay; too little green in peaks brings about queues.

A. Actuated and Adaptive Systems

There are reported deployments of centralized adaptive systems which include SCOOT and SCATS which show quantifiable reductions in delay and improved robustness under fluctuating demand conditions [2]. However, the described performance improvements vary significantly with the network topology, the characteristics of traffic demand and calibration parameters assumed in each study. Although these systems have been successful in their operation, they have some weaknesses:
High capital costs and proprietary algorithms restrict adoption
The rule-based logic is not generalized towards new traffic patterns.
Complex networks with nonlinear interactions can be characterized by scaling difficulties.
The inability learn from patterns or adapt to any structural changes
The origin of RL in the field of traffic control can be traced to the 1990s-2000s following the fame of Watkins Q-learning theorem [5] , the fundamental concept of the latter intersection control was to be modeled as a Markov Decision Process (MDP) whereby traffic lights were treated. decision-making agents, who perceive their surroundings, act and receive some feedback on reward. Early studies:
Prashanth & Bhatnagar (2011) [6]: Developed stochastic approximation-based Q-learning for uncertain traffic with queue-length rewards, proving convergence under mild conditions.
El-Tantawy et al. (2013) [7]: suggested a multi-agent Q-learning model whose intersection agents are decentralized and coordinated reward sharing. In their findings, they showed significant delays and better coordination than classical rule-based controllers in simulated multi-intersection networks.

B. Deep Reinforcement Learning Era

The DQN methodology by Mnih et al. [8] was an important advance in the area of reinforcement learning that proposed the use of neural networks as a part of it. function approximation of values, thus allowing the representation of high dimensional state representations.
Atari success motivated traffic applications:

PressLight (Wei et al., 2019) [10]:used the lane-pressure representations and a max-pressure-inspired reward formua- tion to promote vehicle throughput. The grid network evaluation through experimentation in networks showed better coordination and scalability by decentralized computing of pressure.
CoLight (Zang et al., 2020) [11]: graph neural networks made with leverage to model inter-intersection relationship, which allows cooperative multi-intersection control and showing a better performance than independent deep RL agents throughout multifaceted network situations.
MetaLight (Zang et al., 2021) [12]: Meta-RL model that allows the adaptation of policy to new intersections using few examples, eliminated, shot learning, and avoided significant retraining.
Emissions-aware objectives (Zhao et al., 2020) [13]: the emissions based goals were integrated, such as fuel consump- tion and pollutant production, in reinforcement learning rewards functions, and proving the possibility of joint optimiza- tion and traffic performance on the environment.

III. Critical Research Gaps

Despite advances, gaps remain:

- Inconsistent Benchmarking: The heterogeneous experimental conditions (network topology, traffic profiles, detector placement, episode length, traffic demand distribution) do not allow the fair cross-paper comparison. One study's 20% improvement may result due to favorable traffic and not better algorithms.
- Limited Algorithmic Comparison: Most of the papers use only one approach (typically DQN). Only few of them compare static vs. tabular vs. deep RL; even fewer use identical conditions.
- Underutilized Sensor Input: Most studies aggregate state (e.g., total vehicles across all lanes) rather than lane- level or distance binned detectors. This loses distributional information that is important in the accurate management of the queue.
- Offline Training Dominance: The existing reinforcement learning agents are commonly assessed through the episodic or preset simulation environment. The reinforcement learning agents are intended to exist in the form of agents. Online education through real time interaction TraCI based central. Comparatively little systematic re- search has been done on to the reinforcement learning paradigm.

A. Positioning of This Work

The given gaps are systematically addressed in the following paper:

- Single SUMO environment provides the same network, traffic demand, and detector configuration among the con- trollers.
- Three-algorithm comparison (static, Q-learning, DQN) under identical conditions isolates algorithmic differences
- They have six lane-area E1 detectors and offer realistic multi-level queue representation.
- Broad performance is estimated by extensive metrics (delay, queues, travel times at mean/90th/max, throughput) in the broad performance metrics
- Online TraCI-based training has real-time learning ability.
- Complete methodology specification enables reproducibility and future extensions

III. Methodology

A. Simulation Environment Setup

SUMO Network Configuration: The simulation is based on a four-way urban crossroad where there are two main approaches through which the traffic demand is limited. Eastbound (EB) and Southbound (SB), and the rest of the approaches have almost zero or no inflow.
Detector Placement and State Sensing: Lane-area (E1) detectors are positioned at three critical locations per approach:

- Near: 10 meters upstream (queue formation zone)
- Mid: 40 meters upstream (backup accumulation zone)
- Far: 70 meters upstream (signal influence limit)

This multi-level structure reflects progressive queues accumulation, which allows the agent to differentiate between light congestions. tion (vehicles at near detector only), severe congestion (vehicles at far detector). Every detector has an indication of vehicle count per second update cycle over TraCI. A total of six lane-area E 1 detectors (three per approach) are used to compose the. The major input of information to the reinforcement learning agents.

Traffic Demand Generation: The vehicle paths are based on a route definition file that defines:

- Type of vehicle: passenger cars of realistic length (5m), acceleration (3 m/s 2), deceleration (4.5 m/s 2).
- Demand profile: arrivals of the vehicles with rate parameter λ vehicles/second
- Route distribution: the EB and SB methods with probability (0.5 and 0.5, respectively) of equal demand; diverse probability of different demand.
- Simulation span: 3000 seconds (50 minutes) per episode

TraCI Integration: TraCI (Traffic Control Interface),is a Python client-server protocol that allows real time interaction between a simulation agent interface and a real world object, it is used with the simulation agent interface.

- traci.lanearea.getLastStepVehicleNumber(id): This reads the current detector occupancy
- traci.trafficlight.getPhase(id): It reads the active phase
- traci.trafficlight.setPhase(id, phase): It writes the new phase
- traci.simulationStep(): Advance simulation by 1 second

Here the decision intervals are taken after every 5 seconds (minimum green time), which allows the agent to react to the changes in the queue, avoiding the fast phase swings to the traffic flow.

B. Reinforcement Learning Formulation

Markov Decision Process Definition: Here the traffic signal control problems are modeled as an episodic MDP (S, A, *P, R, γ*):

$$S_t = [\, q^tEB, near, q^tEB, midq^tEB, far, \quad q^tSB, near\,, q^tSB, mid\,, q^tSB, far, \phi_t\,] \qquad (1)$$

where $q^t_{i,j}$ denotes vehicles in detector j of approach i at time t, and ϕt ∈ {0, 1} encodes the active phase (0=EB green,1=SB green).

Action Space: At ∈ {0, 1}:

$$A_t = \{0 \text{ keep current phase } \text{ switch to next phase} \qquad (2)$$

Reward Function: Queue-based negative reward:

$$R_t = -\sum_{i\,\epsilon\,\{EB,SB\}\,j\epsilon} \quad \sum_{\{near,mid,far\}} \quad q^t_{i,j} \qquad (3)$$

This function directly penalizes congestion, the more the number of vehicles waiting, it gets the more negative reward. The agent learns that reduction of waiting lines leads to maximization of cumulative return. The rewards are calculated at every decision step which can be of 5 seconds to maintain consistent reward with the frequency of the action by an agent.

Detector counts have been summed up in the past 5 seconds duration to make sure that there is a steady reward with the frequency of the action of an agent.

Transition dynamics P ($s_{t+1}|s_t, a_t$): The SUMO's microscopic model advances the vehicle's positions, applies the lane changing logic, and updates the detector counts based on the traffic flow and state of the signal.

Discount factor: $\gamma = 0.95$, standard for traffic applications (balances immediate and future performance).

Algorithm 1: Tabular Q-Learning for Traffic Signal Control

```
Input: Learning rate α, discount γ, exploration ϵ0
Initialize Q(s, a) ← 0 for all s, a;
for episode e = 1 to E do
    ϵe = ϵ0(0.995)^e ;                                   // decay exploration
    Reset SUMO, initialize s0;
    for t = 0 to T − 1 (in 5s intervals) do
        Discretize state: st ← bin queue values (0–30 vehicles per bin);
        if rand() < ϵe then
            at ~ Uniform(A) ;                            // explore
        else
            at ← arg maxa Q(st, a) ;                     // exploit
        end
        Execute at, receive rt, observe st+1;
        Q(st, at) ← Q(st, at) + α [rt + γ maxa' Q(st+1, a') − Q(st, at)];
    end
end
return policy π*(s) = arg maxa Q(s, a);
```

C. Q-Learning Controller

Classical tabular Q-learning estimates action-value function Q : S × A → R via temporal-difference updates: Hyperparameters are set as follows: learning rate α = 0.1, discount factor γ = 0.95, and initial exploration rate ϵ0 = 1.0 decayed exponentially to a minimum value of 0.01. State discretization is carried out by grouping queue lengths into intervals of five vehicles, producing six discrete bins per detector for queue sizes ranging from 0 to 30 vehicles. With six detectors and two signal phases, the resulting discrete state space contains 66 × 2 = 93,312 states, enabling tabular Q- value storage. Due to the stochastic and non-stationary nature of microscopic traffic simulation in SUMO, convergence of the tabular Q-learning controller is evaluated empirically through performance stabilization rather than being guaranteed theoretically.

Deep Q-Network (DQN) Controller

The Deep Q-Network (DQN) approach estimates the action–value function Q(s, a) by means of a neural network Qθ(s, a) where the network parameters are represented by the weight vector θ. A target network Qθ− (s, a) and an experience replay buffer are employed to stabilize training.

IV. Results and Discussion

A. Experimental Setup Summary

All experiments conducted with:

- Balanced traffic demand: λ = 0.3 vehicles/second per approach (18 veh/min), representing moderate urban congestion
- Network: 4-way intersection (EB, WB, NB, SB) with focus on EB-SB primary movements
- Detector configuration: 6 E1 detectors (3 per EB/SB approach) at near/mid/far positions
- Episode duration: 3000 seconds (50 minutes)
- Decision interval: 5 seconds (minimum green duration = 5 sec)
- Training: 100 episodes per controller
- Evaluation: 10 test episodes, metrics averaged

B. Analysis of Results

Total Delay and Queue Length: The DQN controller achieves a 40.0% reduction in total delay (from 12,500 → 7,500 vehicle-s) compared with the fixed- time baseline. When EB approaches become heavily congested (high queue at far detectors), DQN extends EB green
phase; conversely, when SB congestion dominates, phases switch to SB green. The fixed-time controls that are bound to 30-second cycles, cannot exploit these imbalances.

An average reduction of 41.5% (8.2 → 4.8 veh) in queue length reflects the improved phase allocation. The 41.7% (24→ 14 veh) reduction indicates that DQN does not experience major congestion stacking and the
queues depths are more uniform over time.

Q-learning reduces the delay by 24.8 % - still very high but 15.2% below DQN. Discretization of the state (binning queues in 5-vehicle components) leads to the loss of information, especially on fine-grained queue variations. The 3-vehicle and 7-vehicle queues are, for instance, identical discrete states, 5-vehicle bins, which have lost discrimination. This more discrete representation reduces the rate at which accurate phase-switching policies can be learnt leading to prolonged time to convergence and less than optimal steady-state policies.

Algorithm 2: Deep Q-Network for Traffic Signal Control

```
Input: Learning rate α, discount factor γ, target update interval C, replay buffer size |D|max Initialize
online network Qθ(s, a) with random weights θ;
Initialize target network Q − (s, a) with θ⁻
← θ; Initialize replay buffer D ← ∅;
Set ϵ0 = 1.0, ϵmin = 0.01;
for episode e = 1 to E do
    ϵ ← max(ϵmin, ϵ0(0.995)ᵉ);
    Reset SUMO environment and observe initial state s0;
    for t = 0 to T
        − 1 do if
        rand() < /
        ϵ then
            Select random action at;
        else
            at ← arg maxa Qθ(st, a);
        end
        Execute at, receive rt, observe
        st+1; Store (st, at, rt, st+1) in D;
        if |D| ≥ 1000 then
            Sample minibatch B of size 32;
            Compute loss and update θ using Adam optimizer;
        end
        if t mod C = 0 then
            θ⁻ ← θ;
        end
        st ← st+1;
    end
end
return learned policy π*(s);
```

TABLE 1: PERFORMANCE METRICS COMPARISON: FIXED-TIME BASELINE VS. Q-LEARNING VS. DQN (MEAN ± STD DEV ACROSS 10 TEST EPISODES)

Metric	Fixed-Time	Q-Learning	DQN	QL vs FT	DQN vs FT
Total Delay (veh-s)	12,500 ± 1,200	9,400 ± 900	7,500 ± 600	-24.8%	-40.0%
Avg Queue (veh)	8.2 ± 1.4	6.1 ± 1.2	4.8 ± 0.8	-25.6%	-41.5%
Max Queue (veh)	24 ± 3	18 ± 2	14 ± 2	-25.0%	-41.7%
Throughput (veh/h)	850 ± 45	920 ± 50	950 ± 35	+8.2%	+11.8%
Avg Travel Time (s)	45 ± 8	38 ± 6	32 ± 5	-15.6%	-28.9%
90th Travel Time (s)	72 ± 12	58 ± 10	48 ± 8	-19.4%	-33.3%
Max Travel Time (s)	128 ± 18	98 ± 14	72 ± 10	-23.4%	-43.8%

C. Throughput and Travel Times

The DQN increases the throughput by 11.8% (850 → 950 veh/h), which indicates that the more vehicles passes the intersection per unit time. This improvement is majorly due to the reduced delays, since now the vehicles spend less time waiting in queues, which decreases the overall congestion and allows for a smoother and faster clearance of traffic.

The maximum travel time decreases by 43.8% (128 → 72 sec), and 90th percentile decreases by 33.3% (72 → 48 sec). These reductions are critical for the reliability of services ,the urban travelers care highly about the worst-case delays. The adaptive control of DQN also makes sure that even vehicles that arrives when the demands is at peak do not have to spend much time queuing.

D. Controller Stability and Consistency

Table 1 shows standard deviations that are string:

- Fixed-time: The middle range of variance (± 4-6%) is a stochastic traffic variability that cannot be controlled by policy.
- Q-learning: This has a slightly reduced variance than fixed-time, which means that learned policies smooth some variability.
- DQN: The lowest variance of (± 3-5%), suggesting the most robust and consistent performance across traffic real- izations.

E. Computational Requirements

The training time and inference latency are used to calculate the cost for each controller.

- Q-learning: With more than 100 episodes, for each 3,000 second of simulation time, SUMO will have performed approximately 300,000 control steps. The total training time is around 2 hours on a standard desktop CPU using a Python-based implementation.
- DQN: 100 episodes with each running for 3,000 seconds are used for training. In this method, around 300 minibatch updates occur, taking a total of 4–5 hours to train on the same hardware configuration. There is estimated to be 20–30% computational cost associated with experience replay sampling methods, as well as target network updates, as seen with tabular Q-learning methods.
- Inference: At deployment time, action selection using the Q-learning controller requires a simple table lookup (∼1 μs), while the DQN controller performs a forward pass through the neural network (∼0.1 ms). Also, both inference times are negligible compared to the 5-second window of the control interval, thereby indicating that it is possible for us to carry out inference operations in real-time.

F. Scenario Sensitivity

Asymmetric Traffic Demand

Test under 70% EB / 30% SB arrival distribution (EB-heavy):

Fixed-time: 30-second cycles force equal green allocation, suboptimal when EB dominates. Total delay increases 18%.

Q-learning: Learns EB-biased policy, achieves 35% improvement vs. fixed-time (vs. 24.8% in balanced scenario).

DQN: Achieves 42% improvement (vs. 40.0% baseline), nearly matching balanced-demand performance.

Key Observations and Insights

Deep RL Advantage: The DQN outperforms the Q-learning by 40% vs. 24.8% delay improvement and all this is due to continuous state handling and neural network generalization.

Robustness: DQN's shows the least variance, which means that the performance is stable and across traffic stochasticity, which is important in the real world.
Tail Performance: DQN shows significant gains in worst case performance via the dramatic decrease in 90th percentile and maximum travel time (33-44%) that are of great importance in urban services.
Scalability to Imbalance: DQN can be applied effectively to an asymmetric demand, unlike fixed-time methods, which have to be returned according to each traffic pattern.
Convergence Speed: DQN converges in 30-40 episodes (25-33 hours simulation), which is useful in real world setting of learning

V. CONCLUSION

The results of this paper confirm the effectiveness of the concept of reinforcement learning that can be used in the next generation of intelligent transportation systems. After a strict comparison of static fixed-time baseline, tabular Q-learning and Deep Q-Network controller using the same network, traffic demand and detector parameters, we demonstrated that:

- Compared to traditional fixed-time signals the DQN-based adaptive control achieves 40% reduction in total vehicle delay and 12% increase in the throughput.
- A delay reduction of 24.8% with tabular Q-learning confirms the classical RL in traffic control but shows limitations in state discretization.
- The high performance of Deep RL is due to the continuous state approximation using a neural network which would allow more desirable generalization to traffic conditions and imbalance.
- Maximum reduction in the worst-case travel times will be significantly enhanced (43.8% reduction in the maximum delay), which is important for the reliability of urban service and passenger experience.
- The results of RL controllers show a stable repeatable performance (low variance), which are resistant to stochastic traffic variations.

The results obtained from this study validate the efficiency of the concept of reinforcement learning that can be adopted for the development of next generation intelligent transportation systems.

REFERENCES

[1] F. V. Webster, "Traffic signal settings," Road Research Technical Paper 39, Road Research Laboratory, Harmondsworth, UK, 1958.
[2] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," Proceedings of the IEEE, vol. 91, no. 12, pp. 2043–2067, Dec. 2003.
[3] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. C. Winton, "SCOOT—A traffic responsive method of coordinating signals,"Contractor Report 99, Transport and Road Research Laboratory, Wokingham, UK, 1982.
[4] P. R. Lowrie, "SCATS, Sydney Coordinated Adaptive Traffic System—A traffic responsive method of coordinating signals," Road Traffic Control, pp. 67–82, 1990.
[5] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, no. 3–4, pp. 279–292, May 1992.
[6] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with average cost for adaptive traffic signal control," in Proceedings of the 18th IFAC World Congress, Milano, Italy, 2011, pp. 8855–8860.
[7] S. El-Tantawy, B. Abdulhai, and H. Zaki, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC)," Transportation Research Record: Journal of the Transportation Research Board, vol. 2375, no. 1, pp. 40–51, Jan. 2013.
[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmu¨ller, A. K. Fidjeland,G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, vol. 529, no. 7540, pp. 529–533, Feb. 2015.
[9] G. Zheng, Y. Xiong, X. Zhu, D. Zhao, H. Zhao, and F.-Y. Wang, "Learning phase competition for traffic signal control," in Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 2019, vol. 33, pp. 1243–1250.
[10] H. Wei, G. Zheng, H. Yao, and Z. Li, "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 2019, pp. 1290–1298.
[11] E. Zang, S. Xu, M. Zhang, D. Zhao, and D. Zhou, "CoLight: Cooperative traffic signal control via deep reinforcement learning," in Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 2020, vol. 34, pp. 10774–10781.
[12] E. Zang, S. Xu, M. Jiang, H. Sun, Z. Zhang, and D. Zhao, "TrafficLearn: A multi-agent hierarchical reinforcement learning framework for traffic signal control," Artificial Intelligence Research, vol. 10, no. 88, pp. 1423–1476, 2021.

[13] D. Zhao, H. Wu, L. Li, Z.-J. Du, P. You, and J. J. Q. Yu, “Distributionally robust reinforcement learning,” in Proceedings of the 34th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 2020.
[14] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO—Simulation of Urban Mobility,” International Journal On Advances in Systems and Measurements, vol. 5, no. 3–4, pp. 128–138, Dec. 2012.