

14th Conference on Transport Engineering: 6th–8th July 2021

Reinforcement Learning for Traffic Signal Control: Comparison with Commercial Systems

Alvaro Cabrejas Egea^{a,*}, Raymond Zhang^a, Neil Walton^a

^a*The Alan Turing Institute, 96 Euston Rd, London NW1 2DB, United Kingdom*

Abstract

Recently, Intelligent Transportation Systems are leveraging the power of increased sensory coverage and computing power to deliver data-intensive solutions achieving higher levels of performance than traditional systems. Within Traffic Signal Control (TSC), this has allowed the emergence of Machine Learning (ML) based systems. Among this group, Reinforcement Learning (RL) approaches have performed particularly well. Given the lack of industry standards in ML for TSC, literature exploring RL often lacks comparison against commercially available systems and straightforward formulations of how the agents operate. Here we attempt to bridge that gap. We propose three different architectures for TSC RL agents and compare them against the currently used commercial systems MOVA, SurTrac and Cyclic controllers and provide pseudo-code for them. The agents use variations of Deep Q-Learning and Actor Critic, using states and rewards based on queue lengths. Their performance is compared in across different map scenarios with variable demand, assessing them in terms of the global delay and average queue length. We find that the RL-based systems can significantly and consistently achieve lower delays when compared with existing commercial systems.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review statement: Peer-review under responsibility of the scientific committee of the 14th Conference on Transport Engineering.

Keywords: Modelling and Simulation; Urban Traffic Control; Reinforcement Learning

1. Introduction

Traffic Signal Control (TSC) can be used to ensure the safe and efficient utilisation of the road network at junctions, where traffic can change directions and merge, having to manage conflicting individual priorities with the global needs of the network. Traffic congestion has a major financial impact. A study by INRIX (2019) shows that traffic congestion in 2019 cost £6.9 billion in the UK alone, with similar patterns being observed in other developed countries. Cities around the globe are starting to explore the deployment of smart Urban Traffic Controllers (UTCs) that use real time data to adjust their stage schedule and green time duration. Traditionally, fixed time plans have been used, with systems that optimise the green time splits in a deterministic manner, requiring costly site-specific knowledge and typical demand profiles to provide effective control. These methods are not easily scalable and deteriorate over

* Corresponding author. Tel.: +44-7511-398-261.

E-mail address: acabrejasegea@turing.ac.uk

time as the traffic demand changes (Bell and Bretherton (1986)). With the development of induction loops, real time actuated UTCs were created in two variants: those that optimise single isolated intersections with systems such as MOVA (Vincent and Peirce (1988)), and those that cover multiple intersections such as SCOOT (Hunt et al. (1982)). To remedy the scalability problem, other systems are based on local rules, generating a self-organising area traffic controllers, such as SurTrac (Smith et al. (2013)). With the breakthrough of Deep Reinforcement Learning (DRL) on complex problems such as Atari games or Go (Mnih et al. (2013); Silver et al. (2017)), attention has turned towards adapting these approaches to generate industry-grade controllers for traditionally noisy and systems such as TSC.

This paper aims to reproduce some of the results of the main and most successful RL approaches on intersections of increasing complexity, while comparing different architectures of DRL TSC agents, since, given the complexity of their implementation, most available literature only deals with a single class.

2. State of the Art

We discuss the state of the art in reinforcement learning in signal control. We discuss the state of the art systems for commercial simulation and traffic signal control which we analyse in this study.

2.1. Previous Work

RL is an area of ML that tries to imitate how biological entities learn, where an agent evolves in an unknown environment and learns how to perform a task for which no prior information is given based on its interactions said environment. The agent aims to maximise a reward signal it receives as a feedback for its actions. RL methods have been applied to TSC in experimental setups. While there is a variety of approaches in the literature that craft successful RL-based TSC systems, most do not present comparisons against commercial systems that are the concern of this paper.

Recent works Liang et al. (2018); Genders (2018); Liu et al. (2018) use neural networks as function approximators to avoid the dimensionality and computing limitations of table based methods in large state-action spaces, showing DRL TSC can be more efficient than some earlier methods. Gao et al. (2017) used a CNN and discrete cell encoding with a Target Network for a value-based agent. The results were compared against a fixed time and a heuristic system (longest queue first), finding RL to perform better. In Mousavi et al. (2017), raw pixels were used as input for a CNN that parametrises two agents: a policy-gradient agent and a value-based agent. The variation in the delay between actions was used as reward, and while both agents were found to have near-identical performance, they were not compared against any reference system. Later, in Wan and Hwang (2018) a DQN using discrete cell encoding as state was implemented. It used a CNN architecture and a delay-based reward. It was compared against a fixed time system, obtaining better performance. Liang et al. (2018), used the same approach and included speed information in the state, using a reward based on variations on aggregated wait time for all vehicles. It compared against two different fixed-time systems, ranking better than both and providing some early evidence of the benefits of using Double DQN (Hasselt (2010)), Duelling architecture (Wang et al. (2016)) and Prioritised Experience Replay (PER) (Schaul et al. (2016)). Genders and Razavi (2018) evaluated different state representations, finding little difference in the performance of the agents as a result of the change in the magnitudes observed. Regarding comparisons with established systems, in Stevanovic and Martin (2008) the authors compare SCOOT with a Genetic Algorithm-based control method. It is shown that SCOOT's performance can be surpassed by more adaptive Genetic Algorithms that, in turn, tend to be less effective at learning than RL methods.

Despite these previous works, most results are hard or impossible to reproduce given the lack of industry standards in terms of simulators, performance metrics, the lack of availability of commercial algorithms for comparison and the fierce protection of their internal workings, and the lack of open-source code of proposed RL models.

2.2. Commercial Traffic Signal Control Simulators and Optimisers

PTV Vissim. All our simulations are conducted on the PTV simulator, which is a state-of-the-art commercial traffic control simulator that is able to produce a wide variety of traffic demands over an array of different signal controls and road traffic scenarios. We interface our Machine learning algorithms with the PTV Vissim's COM interface. This allows to interface Vissim's control interface with Tensorflow to construct deep learning agents.

MOVA (Microprocessor Optimised Vehicle Actuation, Vincent and Peirce (1988)) is a traffic controller designed by TRL Software. It aims to reduce delay on isolated junctions. The basic functioning of MOVA involves two induction loop detectors estimating the flow of vehicles in each lane. The system makes a virtual cell representation of the lanes within MOVA, and then it computes a performance index based on the delays calculated. If the index results lower than a certain threshold, the signal is changed to the next stage, otherwise the stage it is extended.

Surtrac (Scalable URban TRaffic Control, Smith et al. (2013)) is an adaptive TSC system published in 2013. A real-world deployment on 20 intersections in Pennsylvania showed a performance improvement of 20-40%. Its operation is decentralised, each intersection allocates green time independently and asynchronously based on incoming flows. Each intersection is controlled by a local scheduler and communicates projected outflows to the downstream neighbouring junctions, modelling vehicles as a sequence of clusters. This communication allows for locally balancing competing flows while creating larger "green corridors" by finding an optimal sequence such that the input jobs (ordered clusters) are cleared while minimising the joint waiting time of all vehicles.

3. Methods

Traffic Control as a Markov Decision Process. The control problem can be formulated as a Markov Decision Process (MDP) defined in terms of a 5-tuple: A set of possible environment states $s \in \mathcal{S}$, a set of actions of the agent $a \in \mathcal{A}$, a stochastic transition function $\forall a \in \mathcal{A}, \mathcal{T}_{s,s'}^a \triangleq \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$, a scalar real valued reward function $R(s_t, s_{t+1}, a_t)$ providing a performance measure to the transition generated by progressing into the state s_{t+1} after taking action a_t while in state s_t , and a discount factor γ that will provide the balance between immediate exploitation and approaches that aim to maximise returns over time. In the case of TSC, the MDP is modelled as partially observable, following an unknown stochastic transition function.

Reinforcement Learning. The goal of the agents will be to maximise their future discounted return, $G_t = \sum_{t=0}^{+\infty} \gamma^t r(s_t, a_t)$ with $\gamma \in [0, 1]$. This is done by learning a policy π , parametrised by the weights θ of the neural network performing the approximation of the reward function and mapping states to actions: $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The reward function maps an action given a state to a reward scalar value: $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. The action-value function or Q-value is $Q_\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a]$. It represents the total episodic return by following policy π after being in state s and taking action a .

Value-based Reinforcement Learning Methods - DQN. Tabular value-based methods, such as Q-Learning, attempt to learn an optimal policy $Q_\pi^* = \max_\pi \mathbb{E}[r_t | s_t = s, a_t = a]$ by iteratively performing Bellman updates on the Q-values of the individual state-action pairs:

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha (y_t - Q_\pi(s_{t+1}, a_{t+1})), \quad \text{with } y_t = r_t + \gamma \max_{a'} Q_\pi(s_{t+1}, a', \theta'), \quad (1)$$

where α is the learning rate and y_t is the Temporal Difference (TD) target for the value function.

Deep Q-Network (DQN) agents are an evolution of Q-Learning. The purpose of the agent is to find an approximation of Q_π^* by tuning the weights θ of a neural network. The agent keeps a second neural network, the target network, parametrised by the weights vector θ' which is used to generate the TD targets.

The experience replay memory increases training stability, obtaining samples that cover a wider range of situations and that can be used several times for gradient descent. Three additional modules have been applied to the agent to improve performance, Double Q Learning, PER, and Dueling Architecture. We refer the reader to the online supplementary material for pseudo-code of these standard deep learning algorithms and the hyper-parameters used.

Policy Gradient Reinforcement Learning Methods - A2C. Policy Gradient in RL is based on the idea that obtaining a direct policy $\pi(s)$ mapping states to actions can be easier than estimating the value function or the state-action values. It has an added benefit in that it can learn stochastic policies, generating a probability distribution over the potential actions. The goal is to find the policy that maximises the reward. To do so one has to perform gradient ascent on the performance measure $J = \sum_{a \in \mathcal{A}} \mathbb{E}[Q(s_0, a)\pi(a|s)]$.

The Advantage Actor Critic (A2C) method tries to reduce the variance in the policy method by combining the direct mapping from actions with the value-based approximation method. The goal is to learn an actor

$$\pi_\theta = \mathbb{P}_\theta[a_t = a | s_t = s], \quad \text{and a critic } V_\pi^\theta(s) = \mathbb{E}_\theta[G_t | s_t = s], \quad (2)$$

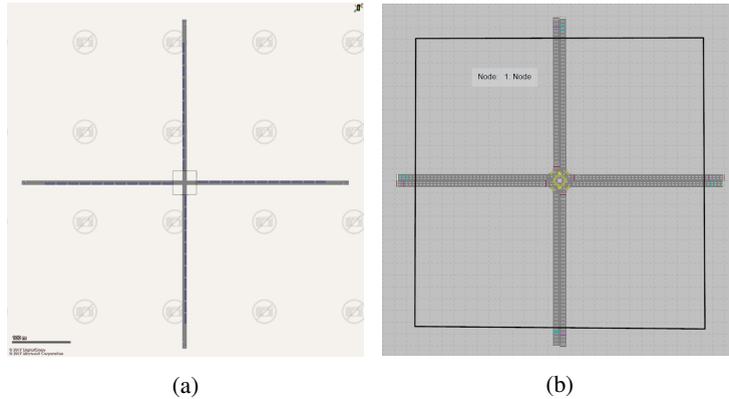


Fig. 1: (a): Cross Straight Map. (b): Cross Triple Map.

both of which are parametrised by the neural network weights vector θ .

All the experiments here presented use the same descriptions for simulator state and reward calculation, although they differ in the number of actions available to them. The state of an intersection of l lanes will be presented to the agents as a state vector $s \in \mathbb{R}^{l+1}$, in which each component represents the length of the queue of vehicles measured upstream from the traffic light in metres. The last component will be current stage being implemented.

While marginal improvements in performance can be obtained by using different variables for reward (Cabrejas Egea et al.; Cabrejas-Egea and Connaughton (2020)), as per the discussion of Heydecker (2004), queues can be a reasonable choice for states and rewards, being able to transmit useful information to the agent relative to the mean rate of delay of the system. Based on this, the reward after an action will be calculated as the negative sum of the length of the queues of all lanes immediately upstream from the intersection: $r_t = -\sum_l q_{l,t}$.

The agent has a set of actions \mathcal{A} that varies depending on the intersection being controlled. Once the agent chooses an action a , the stage corresponding with the ID of a is implemented. Green stages are set to a minimum of 6 seconds. Once this time has passed, the agent is requested a new action. If the agent chooses the same action again, the current stage is extended for a further 3 seconds. There are no inbuilt limitations as to how many times an agent can extend a stage, leaving it for the agents to learn. If the agent chooses a different action than the currently active one, a 3 seconds amber stage is implemented in the lights that were green, after which, the new stage is implemented.

Agent Benchmarking. In order to compare the agents performance, a testing framework was defined. For each model, a demand profile will be created, following the shape found in a *typical day* using the methodology introduced in Cabrejas Egea et al. (2018) and expanded in Cabrejas-Egea (2020). The profile will be split on 10 segments of length 6 minutes. Each of these will correspond with a level of demand. The demand levels are obtained by setting the maximum demand the intersection will suffer, setting that magnitude to coincide with the peaks of the distribution that could be found on said *typical day*, and will be specified in each experiment's section. Random seeds are updated after every simulation episode, training or testing. The quantitative metrics on which the system will be evaluated are the Global Cumulative Delay (deviations from free-flow time) and the Average Queue Length generated during the evaluation.

4. Experimental Results

We conducted experiments across three realistic traffic intersections. See 1. MOVA is highly optimized for the first of these junctions. The second junction is a typical of a US arterial traffic junction.

Experiment 1: Cross Straight. The first test is conducted on the simplest junction, shown in Fig 1a. The junction referred to as *Single Cross* is composed on 4 lanes distributed in 4 arms coinciding with the cardinal directions of the model. The controller for the junction has two stages, a north-south stage and an east-west stage, and turning is not allowed. The testing aim was to perform an initial performance comparison of DRL algorithms against MOVA, SUTRAC, and a cyclic controller. Here the goal for the agent was to exert fine adaptive timing control while extrapolating, rather than using complicated transitions between stages that would rarely, if ever, appear in sequence in cyclic

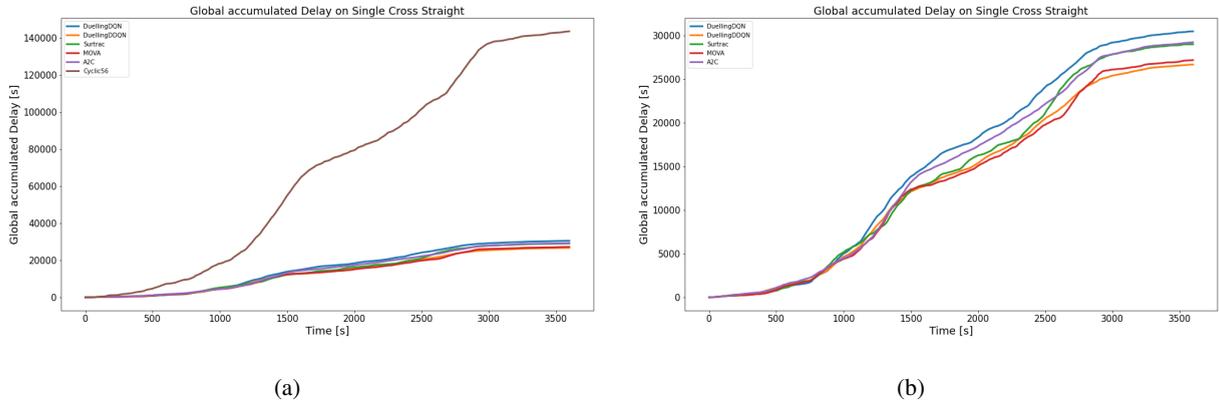


Fig. 2: Global Cumulative Delay in Single Cross for DQN variants, A2C, Surtrac, MOVA and the reference Cyclic controller.

control. MOVA was configured using loop detectors set in accordance to its manual, the implementation of Surtrac follows the work of [Xie et al. \(2012\)](#).

The agents were trained using a fixed vehicle demand of 400 vehicles per hour on each of the incoming lanes. Both DQN variants were trained for 400 episodes, using an ϵ geometrically annealed from 1 to 0.001. The A2C agents were trained for 100 episodes until they converged. Best performing agents in each class were selected for benchmarking and evaluated in scenarios lasting one hour. During evaluation, an average of 2120 vehicles are inserted in the model, with 2 peaks of demand of 3000 vehicles/hour for 6 minutes each.

Figure 4 and Table 1 show the Global Cumulative Delay and average queue length for the network. As expected, the cyclic solution is outperformed by all adaptive controllers. The different controllers are on a par with a slight advantage for the DuelingDDQN which saves the community an average of 3000 seconds compared to MOVA on this hour of simulation, which represents on average 1-2 seconds per vehicle. RL agents also seem slightly more robust against changes in demand, producing lower slopes in the delay graphs in sections of extreme demand.

| Controller | Cumulative Delay [s] | Average Sum of Queues [m] |
|------------|----------------------|---------------------------|
| Cyclic | 143660.50 | 132.37 |
| MOVA | 27187.53 | 60.59 |
| SURTRAC | 29008.36 | 72.41 |
| A2C | 26382.14 | 56.07 |
| DDQN | 28303.94 | 50.11 |
| DDDQN | 21286.86 | 49.42 |

Table 1: Cumulative Delay and Cumulative Stop Delay in seconds across Controllers on Single Cross Straight.

The cyclic controller resulted in saturated lanes during both peaks and queues in excess of 150 metres during a great part of the simulation. MOVA suffered two moments in which at least a sensor was saturated coinciding with the peaks in demand, however the queues were close to lengths of around 50 metres during the most part of the simulator. Surtrac followed a similar pattern, having a single lane saturated coinciding with the second peak in demand. RL agents as suffered no saturation in any of their lanes during the length of the evaluation. They all managed a more balanced distribution of queues in their respective lanes, displaying a higher ability to balance loads during peak times.

Because of the simplicity of this 2 actions intersection, there is not a lot of delay difference between adaptive UTCs. As it will be appreciated shortly, these results will change when we consider more complex junctions.

Given the difference in performance between the adaptive and cyclic controllers, which is expected to become greater on more complex intersections, and the increasing difficulty in setting them in large intersections, the cyclic

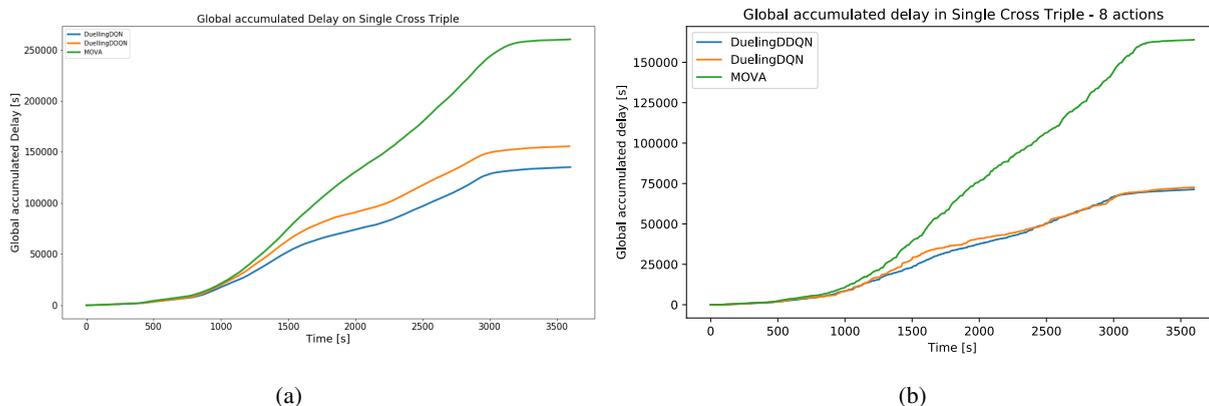


Fig. 3: (a) Global Cumulative Delay in Single Cross Triple - 4 actions. (b) Global Cumulative Delay in Single Cross Triple - 8 actions.

controller will be omitted for the next examples. Given that the A2C agent has been clearly outperformed in this experiment by those based on the DQN architecture, the following experiments will focus on the performance of this last architecture compared with commercial systems.

Experiment 2: Cross Triple - 4 actions. This junction, as shown in Fig. 1b, displays a much higher complexity than the intersection presented in the previous section. It is composed of 4 incoming links of 3 lanes each. In each incoming link, the left lane serves a dedicated nearside turning lane, the central allows for forward travel and the right lane allows for both offside turning and going straight. To mitigate this, the first experiment was run with agents that would take 4 queue inputs, plus the state of the traffic signal as state input. The action set was consequently limited to 4 different actions, being allowed only those that set to green the 3 traffic lights serving the lanes of the same incoming link. This allows for turning vehicles, but prevents more sophisticated stages from happening.

The agents were trained on a fixed demand of 400 vehicles per hours on each incoming link. The DQNs were trained for 400 episodes of one hour of simulated time, with ϵ annealed geometrically between 1 and 0.001. A2C agents were trained for 100 episodes until they converged. During the hour of evaluation, the demand profile from the last experiment was used with a scaling factor of 1.5, an average of 3180 vehicles were introduced to the model, with 2 peaks of demand of 4500 vehicles/hour for 6 minutes each.

As it can be seen in Table 2 the UTC using MOVA performs poorly compared to the DQN-based agents. During this hour of simulation RL agents halve the cumulative delay, saving over 27 hours of travel time for all vehicles involved, an average of over 32 seconds of per vehicle. The length of the queues in those intersections controlled by RL agents during the test scenario were lower than the ones controlled by MOVA. Further in Figure 4b we see that DDQN significantly outperforms MOVA over a wide range of loads and traffic scenarios. It can be seen that the agent using Dueling Q-Learning has a better performance than that Dueling Double Q-Learning.

| Controller | Cumulative Delay [s] | Average Max Queue Length [m] |
|------------|----------------------|------------------------------|
| MOVA | 260257.65 | 179.27 |
| DDQN | 135220.91 | 153.58 |
| DDDQN | 155563.22 | 128.20 |

Table 2: Cumulative Delay in seconds and Average Sum of Max Queue length in meters on Single Cross Triple - 4 actions.

Experiment 3: Cross Triple - 8 actions. In order to allow the use of a higher variety of stages in the controllers the map was reworked. All original lanes were partitioned into their own independent links, allowing extra space for lane changes. While these modifications allowed using information from all lanes in an akin manner to what modern sensors would achieve, due the limitations to lane changing, direct comparisons with Experiment 2 must be handed

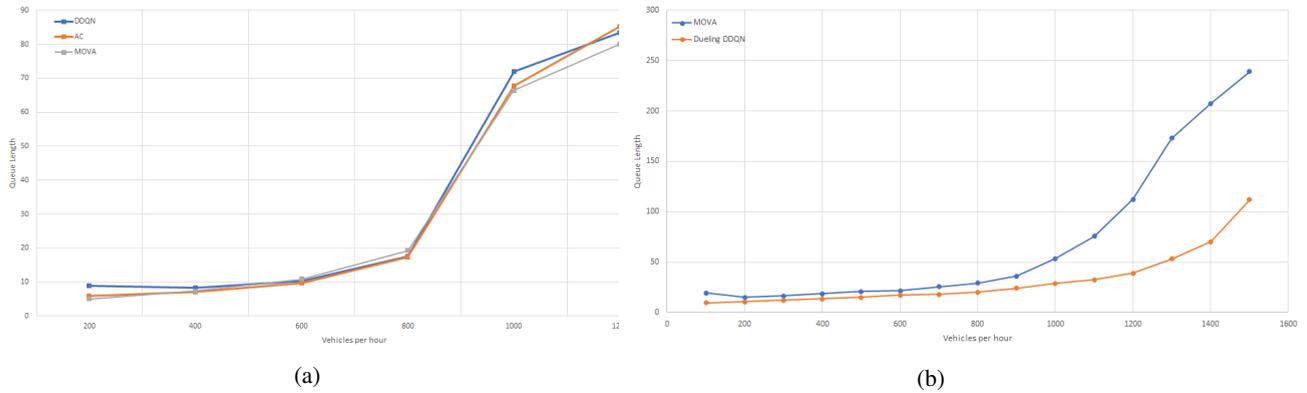


Fig. 4: (a) Queue length by demand level in Cross Straight. (b) Queue length by demand level in Cross Triple.

with care. Both models share name and rough geometry, but the layout of lanes is changed and so are the routing possibilities open to the vehicles.

The results presented below, use DQN agents taking 12 queue length inputs plus the state of the signal. There are 8 different stages. Here there are 4 additional stages available for non-conflicting cross traffic. No specific stage order is enforced, and the agents are free to change between any combination of stages.

| Controller | Cumulative Delay [s] | Average Sum of Queues [m] |
|------------|----------------------|---------------------------|
| MOVA | 165456.44 | 339.41 |
| DDQN | 72642.59 | 123.52 |
| DDDQN | 71245.61 | 119.86 |

Table 3: Cumulative Delay and Cumulative Stop Delay in seconds across Controllers on Single Cross Triple - 8 actions.

The RL agents display a similar gap in performance with MOVA as in the previous experiment, with both classes benefiting from the increased actions pool. RL agents manage to generate about a third of the delay produced by MOVA. While this appears to be a great success, these results have to be put into context. MOVA has a lot of internal parameters meant to be fine tuned by a traffic engineer with site-specific knowledge. Our settings did produce a successful control loop, operating in line with what was expected of the configuration process. None of the RL agents has been fine tuned to the level that would be expected during commercial operation. The layers and neuron distribution wasn't optimised, nor were the activation functions, meaning that the RL agents can still be improved upon.

5. Discussion and Future Work

Several neural network architectures for RL controllers were tested. The agents did not require extensive or complex configurations to adequately control traffic junctions, outperforming the commercial controllers. RL agents showed great stability and robustness to control situations within their training envelope as well as outside of it. Additionally, agents trained on relatively low uniform demand showed they can perform better than commercial systems during evaluation tests that included variable demand 5 times higher than anything experienced during training.

In Experiment 1 MOVA and the RL agent following a DuelingDDQN architecture obtained very similar results, with a slight advantage for the RL agent. Experiment 2 implies less granularity in the data and makes the control task more challenging. The results followed the same pattern with a DuelingDDQN agent obtaining the best performance over an array of loadings, despite the lower resolution in the input data. Experiment 3 saw the introduction of a much more complex intersection. Once again RL agents obtained significantly better results than MOVA, with the DuelingDDQN agent obtaining the lowest global and stop delay. The gap between the performance of MOVA and RL agents is increased here with respect to the last experiment. Most likely reasons are higher granularity in the data and extra actions being available to the agent, allowing it to display more complex sequences of actions.

We find that Reinforcement Learning applied to UTC can significantly outperform current adaptive traffic controllers in realistic commercial simulation softwares. These experiments provide credible evidence that Reinforcement Learning based UTC will be part of the next generation of traffic signal controllers.

Acknowledgements. This work was part funded by EPSRC Grant EP/L015374, by The Alan Turing Institute and the Toyota Mobility Foundation. The authors thank Dr. W. Chernicoff for the discussions that made this project possible.

References

- Bell, M.C., Bretherton, R.D., 1986. Ageing of fixed-time traffic signal plans, in: International conference on road traffic control.
- Cabrejas-Egea, A., 2020. Wavelet Augmented Regression Profiling (WARP): improved long-term estimation of travel time series with recurrent congestion, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. doi:[10.1109/itsc45102.2020.9294318](https://doi.org/10.1109/itsc45102.2020.9294318).
- Cabrejas-Egea, A., Connaughton, C., 2020. Assessment of Reward Functions in Reinforcement Learning for Multi-Modal Urban Traffic Control under Real-World limitations. arXiv preprint arXiv:2010.08819 [arXiv:arXiv:2010.08819](https://arxiv.org/abs/2010.08819).
- Cabrejas Egea, A., De Ford, P., Connaughton, C., 2018. Estimating Baseline Travel Times for the UK Strategic Road Network, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, IEEE. pp. 531–536. doi:[10.1109/ITSC.2018.8569924](https://doi.org/10.1109/ITSC.2018.8569924).
- Cabrejas Egea, A., Howell, S., Knutins, M., Connaughton, C., . Assessment of Reward Functions for Reinforcement Learning Traffic Signal Control under Real-World Limitations, in: IEEE International Conference on Systems, Man and Cybernetics, Proceedings, SMC. doi:[10.1109/smc42975.2020.9283498](https://doi.org/10.1109/smc42975.2020.9283498).
- Gao, J., Shen, Y., Liu, J., Ito, M., Shiratori, N., 2017. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. arXiv preprint arXiv:1705.02755 .
- Genders, W., 2018. Deep reinforcement learning adaptive traffic signal control. Ph.D. thesis.
- Genders, W., Razavi, S., 2018. Evaluating reinforcement learning state representations for adaptive traffic signal control. *Procedia computer science* 130, 26–33.
- Hasselt, H.V., 2010. Double Q-learning, in: *Advances in neural information processing systems*, pp. 2613–2621.
- Heydecker, B.G., 2004. Objectives, stimulus and feedback in signal control of road traffic. *Journal of Intelligent Transportation Systems* 8, 63–76.
- Hunt, P.B., Robertson, D.I., Bretherton, R.D., Royle, M.C., 1982. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control* 23.
- INRIX, 2019. Scorecard. URL: <http://inrix.com/scorecard/>.
- Liang, X., Du, X., Wang, G., Han, Z., 2018. Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks XX, 1–11. URL: <http://arxiv.org/abs/1803.11115>, [arXiv:1803.11115](https://arxiv.org/abs/1803.11115).
- Liu, Y., Liu, L., Chen, W.P., 2018. Intelligent traffic light control using distributed multi-agent Q learning. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC 2018-March*, 1–8. URL: <http://arxiv.org/abs/1711.10941>, doi:[10.1109/ITSC.2017.8317730](https://doi.org/10.1109/ITSC.2017.8317730), [arXiv:1711.10941](https://arxiv.org/abs/1711.10941).
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 .
- Mousavi, S.S., Schukat, M., Howley, E., 2017. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems* 11, 417–423.
- Schaul, T., Quan, J., Antonoglou, I., Silver, D., 2016. Prioritized experience replay. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings* [arXiv:1511.05952](https://arxiv.org/abs/1511.05952).
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Others, 2017. Mastering the game of go without human knowledge. *nature* 550, 354–359.
- Smith, S., Barlow, G., Xie, X.F., Rubinstein, Z.B., 2013. SURTRAC: Scalable Urban Traffic Control. *Transportation Research Board Annual Meeting* , 15URL: https://www.ri.cmu.edu/publication/_view.html?pub_id=7408.
- Stevanovic, A., Martin, P.T., 2008. Split-cycle offset optimization technique and coordinated actuated traffic control evaluated through microsimulation. *Transportation Research Record* 2080, 48–56.
- Vincent, R.A., Peirce, J.R., 1988. 'MOVA': Traffic Responsive, Self-optimising Signal Control for Isolated Intersections. *Traffic Management Division, Traffic Group, Transport and Road Research*.
- Wan, C.H., Hwang, M.C., 2018. Value-based deep reinforcement learning for adaptive isolated intersection signal control. *IET Intelligent Transport Systems* 12, 1005–1010.
- Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N., 2016. Dueling Network Architectures for Deep Reinforcement Learning. *33rd International Conference on Machine Learning, ICML 2016* 4, 2939–2947. [arXiv:1511.06581](https://arxiv.org/abs/1511.06581).
- Xie, X.F., Smith, S.F., Lu, L., Barlow, G.J., 2012. Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies* 24, 168–189. doi:[10.1016/j.trc.2012.03.004](https://doi.org/10.1016/j.trc.2012.03.004).