A Deep Reinforcement Learning Approach for Dynamic Traffic Light Control with Transit Signal Priority

Tobias Nousch[1], Runhao Zhou[2], Django Adam[3], Angelika Hirrle[4], and Meng Wang[5]

[1]Postdoctoral Researcher, Technische Universität Dresden, Germany
[2]PhD Student, Technische Universität Dresden, Germany
[3]Researcher, Technische Universität Dresden, Germany
[4]Postdoctoral Researcher, Technische Universität Dresden, Germany
[5]Full Professor, Technische Universität Dresden, Germany

SHORT SUMMARY

Traffic light control (TLC) with transit signal priority (TSP) is a cost-effective way to deal with urban congestion, person delay and traffic emissions. The growing amount of available connected vehicle data offers opportunities for signal control with transit priority, but the conventional control algorithms fall short in fully exploiting them. In this paper, we propose a novel approach for dynamic TLC with TSP at an urban intersection. We propose a novel Double Deep Q-Learning (DDQL) model inspired by van Hasselt's work to deal with the complex real-world intersections. The optimization focus on TSP while balancing the waiting time of all vehicles. A two-layer high-dimensional state space is defined to capture the real-time traffic information, i.e. vehicle position, type and incoming lane. The discrete action space includes the optimal traffic light phase and dynamic phase duration based on local traffic situation. A complex intersection in the inner city of Jena is conducted in an open-source microscopic traffic simulator SUMO. A time-varying traffic demand of motorized individual traffic (MIT), the current TLC controller used by the city, as well as the original timetables of the public transport (PT) are implemented in simulation to formulate a realistic traffic environment. The results of the simulation with our DDQL model indicate a significant enhancement in the performance of traffic light controller by reducing the waiting time of all vehicles, and especially minimizing the loss time of PT.

Keywords: Double Deep Q-Learning, Traffic light control, Transit signal priority, Two-layer state space, Reward.

1 INTRODUCTION

Travel delay and traffic congestion are common problems that disturb the economic and sustainable development of our society. 3.5 billion Euros or 371 Euros per German motorist, were lost in 2021 due to traffic congestion in Germany Pishue (2021). There is also an urgency to reduce CO_2 emissions and fuel consumption. At the supply side, traffic light control (TLC) is one cost-effective measure to respond to the needs, which can minimize unnecessary stops, optimize the traffic flows, and reduce the motorized traffic and person delay. At the demand side, promoting public transport (PT) is a strategic way to address urban traffic problems. From this viewpoint, it is indispensable to make the PT as attractive as possible. One approach to achieve this goal is transit signal priority (TSP) combined with a particular ameliorating traffic light control strategy. The great challenges are to minimize loss time for all road users, coordinated traffic flows and traffic lights throughout the entire road network, and to find an optimal prioritization strategy for PT.

Today, many cities are still deploying traditional TLC or TSP strategies. Traditional TLC strategies can be categorized into three types: pre-timed, actuated and adaptive control Koonce et al. (2008). The disadvantages of pre-timed and actuated control, such as inability to adjust for spontaneous changes in traffic flows and to directly change phase duration, are usually addressed via adaptive control, which dynamically adjusts signal phase duration based on real-time acquired traffic data from sensors (e.g. camera, loop sensor), i.e. the widely-used adaptive control systems such as SCATS Lowrie (1990) and SCOOT Hunt, Robertson, Bretherton, and Royle (1982).

However, there are still shortcomings: 1) The traffic volume data is acquired by the section-based sensors such as loop sensors and cameras. Many urban intersections in USA are not equipped with them or the they are rarely maintained. The data is collected when vehicles pass through sensors or cameras, hence, the information about the vehicle is partially provided by the sensors or cameras. 2) These adaptive control methods are developed based on models and with simplified assumptions about traffic dynamics, which can causes an inaccurate evaluation of the situation and leads to non-optimal actions.

Traditional TSP has two types: Passive priority and active priority Lin, Yang, Zou, and Franz (2015). Both types are currently facing two difficulties. One is the processing the conflict of multiple requests. Christofa et al. Christofa, Papamichail, and Skabardonis (2013), He et al. He, Head, and Ding (2014) and Hu et al. Hu, Park, and Lee (2016) designed different person-delay-based scheduling techniques to optimize and manage the multiple conflicting priority requests at an isolated intersection. These optimization procedures are mixed-integer nonlinear and linear programs, respectively, whose final objectives minimize the person delay. The second drawback is how to reduce delay to motorized individual traffic (MIT). Ma et al. Ma, Liu, and Yang (2013) designed a dynamic programming (DP) approach to generate the conflicting requests serving sequence to maximize TSP and not to delay MIT. They can maintain an appropriate level of saturation degree for each vehicular movement direction. Guo and Wang G. Guo and Wang (2021) proposed the proximal policy optimization with model-based acceleration (PPOMA). They can extract critical features from the raw state information by a deep neural network and utilize model prediction control as model-based method to accelerate the training of DRL. Although these DP and PPOMA methods show good performance on traffic efficiency, they just considered limited bus or tram routes, fixed occupancy rate and fixed schedule deviation, which is too simple for actual traffic conditions. All aforementioned works rely on traffic models, the model assumptions restrict the controller performance and applicability to the designed traffic situations.

To overcome above disadvantages of traditional TLS or TSP strategies, and adaptively control traffic light or transit signal based on real-time traffic information, researchers have been utilizing deep reinforcement learning (DRL) techniques. In the past decade, DRL based TLC or TSP has gained huge attention from both academia and industry. The neural network technology for function approximation has improved RL, enabling it to complete more challenging and complicated tasks. For the DRL based TLC, Wei et al. Wei, Zheng, Yao, and Li (2018) proposed a DRL model based on the Convolutional Neural Network (CNN) architecture and the value-based approach. Van der Pol et al. van der Pol and Oliehoek (2016) integrated transfer planning and max-plus coordination into the conventional Deep Q Network (DQN). The CNN architecture and the value-based approach are the foundations of the DRL model. Both approaches allow one to analyze visual imagery while mapping each state-action pair to a state value and optimize the Q-values to resolve inappropriate traffic phase sequence. Liang et al. Liang, Du, Wang, and Han (2018) divided the whole intersection into small grids to quantify the complex traffic pattern as states. A CNN was designed to match the states and anticipated rewards. Guo et al. M. Guo, Wang, Chan, and Askary (2019) considered the spatial-temporal characteristics of urban traffic in DRL model. For the DRL based TSP, Long et al. Long, Zou, Zhou, and Chung (2021) proposed a DRL framework to solve the priority request conflict in connected vehicle environment. The action is discrete and traffic signal phases can be skipped. However, to the best of our knowledge, the existing researches do not represent the complex dynamics of urban traffic, mainly because the existing approaches are usually either trained in a fixed traffic demand or lack phase-skipping capabilities. Hence, the control strategies are trained with a non-realistic traffic environment setting, which cannot be applied to complex and realistic traffic demand, and traditional human driver environment.

In summary, some works simplify the traffic environment that are far from reality. Secondly, the capability of neural network to extract features from high-dimensional data is not well-addressed in previous works. The state matrix of most previous works is two-dimensional and cannot capture all influencing factors of traffic. Finally, a significant research gap is that the limitation of existing DRL-based control strategies is that they are applied on TLC or TSP only. Hence, the research activities of combined signal control strategy for multi-modal transport have to be conducted.

In this paper, we redefine a Double Q-Learning model inspired by van Hasselt, Guez, and Silver (2015) and propose a novel traffic light phase controller utilizing Double Deep Q-Learning (DDQL) model. A two-layer high-dimensional state space is proposed to capture the influencing factors, especially the incoming lanes. And a reward function to minimize the loss times of all vehicles are conducted. Meanwhile, TSP is integrated in our DDQL model. To train and validate our model, we set up a realistic traffic environment with varying traffic demand based on the road network of the city of Jena in Germany, which includes the original PT timetables. The phase controlling system presented can be readily superimposed to an existing local traffic light control, and minimize implementation costs.

2 DDQL-BASED MODEL

Because Double Deep Q-Learning is not plagued by the overestimation bias and has good performance on handling high-dimensional data, we redesign the neural network architecture of DDQL according to the complex traffic environment. In DDQL, during training, there are two Q-networks and three important elements S, A and R, where S is the state space, A is the action space, and R is the reward function. The system architecture is shown in Fig. 1.

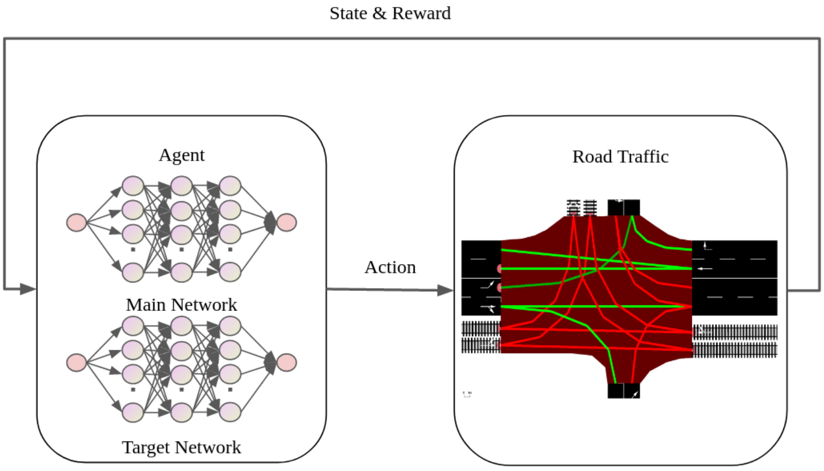


Figure 1: Double Deep Q-Learning Cycle. The agent receive the state space and reward, and perform actions in the road traffic environment

Agent design

A. The neural network architecture of Double Deep Q-Learning

Our model is equipped with experience replay buffer and implemented with the python framework TensorFlow, and consists of the following layers:

- CNN-Layer with 10 filters, kernel size $(1, 10)$ and "ReLu" as activation
- Pooling-Layer with kernel size $(1, 5)$
- CNN-Layer with 1 filter, kernel size $(1, 5)$ and "ReLu" as activation
- Dropout-Layer with 10% dropout
- Dense-Layer with 1 neuron for the Q-value estimation
- Dense-Layer with 7 neurons according to the size of the actions space

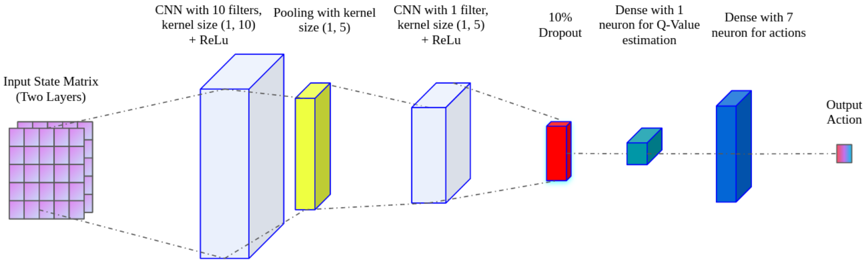


Figure 2: Main Q-Network architecture

The architecture of main network is shown in Fig. 2. The target network has the same settings as the evaluation network and obtains update every 200 training steps. The replay buffer saves 10000 simulation steps and overwrites the memory from the beginning if exceeding the buffer size.

B. State space

The state representation is one crucial aspect in creating a DRL model. In previous works van der Pol and Oliehoek (2016); Wei et al. (2018); M. Guo et al. (2019), the authors represent the entire state in a matrix, such that the entries of the matrix form an overlaying grid of the given state. It has the advantage of being easily adapted to any given environment without much efforts to reconfigure the state space. However, this approach maps huge amount "dead" space, which indicates all the space next to the actual road segments, to state matrix. It requires a lot of computational memories and slows down the computational process and training of the agent considerably.

Therefore, we propose a two-layer state space which is derived as follows: Vehicles are considered to be entering an intersection $500m$ in front of the traffic light on one of the incoming lanes k of the intersection. We partition each lane into segments of length $l = 1m$. Hence, the whole intersection is divided into small grids of equal size. We are now able to form a matrix $P_t \in \mathbb{R}^{500\times k}$, which is called position layer. Each element of the matrix is in $\{0, 1\}$, where it is non zero if there is a vehicle in the corresponding grid. In the same way, we construct a matrix T_t, called type layer. It is of same size as P_t but with entries, representing the type of vehicle, which is zero for MIT and one for PT. We now derive our state matrix as in equation (1). An example of the first layer of the state matrix is shown in Fig. 3.

$$\mathcal{S}_t = \begin{pmatrix} \mathcal{P}_t \\ \mathcal{T}_t \end{pmatrix} \tag{1}$$

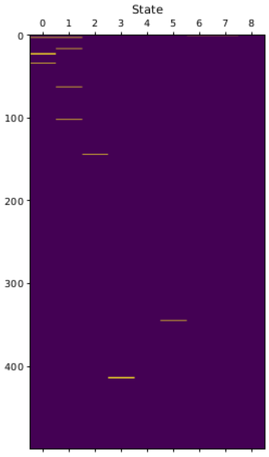


Figure 3: First layer of the state space, where each line represents the position of a vehicle.

If a transition phase or the minimum green time has to be executed, the agent due to the legal constraints can only make a decision after it has been completed. It ensures that all design constraints are fulfilled in every time step according to the received local legal guideline.

C. Action space

The action space contains all possible actions and corresponding duration for a given state. In our case study, the agent decides in an interval of one second whether to keep the current signal phase or to switch to one of the other phases which controls the current traffic efficiently.

D. Reward function

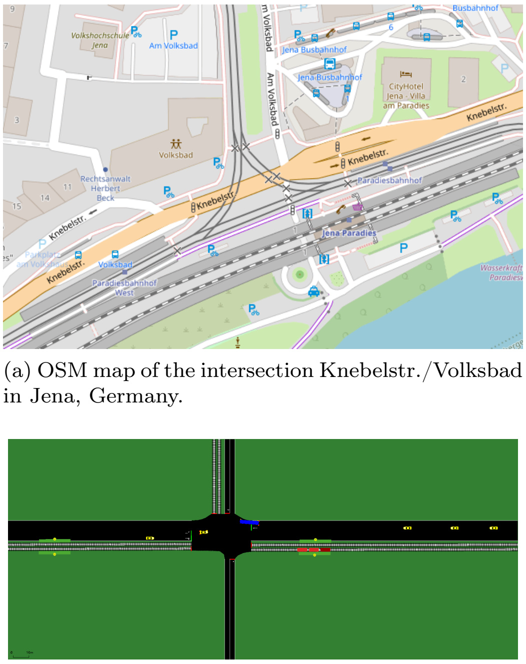
One of the great challenges in DRL is to setup a reward function that represents all the desired properties that the agent can learn and act upon. Our reward function r_t for the selected action in the time step t is formulated as follow.

$$r_t = -\sum_{T} \sum_{\upsilon_T} \left[\eta_T \left(\frac{\tau_{\upsilon_T}}{C_T} \right)^{\rho_T} \right] - \vartheta \sum_{l} q_l \quad . \tag{2}$$

In the first term of (2), T indicates the type of a vehicle, υ_T denotes the number of every vehicle type, the waiting time is τ_{υ_T}, C_T represents the conventional waiting time according to the empirical experience, η_T and ρ_T are defined as specific computational parameters. This term indicates the waiting time of every vehicle in front of the intersection. In the second term of (2), l indicates the index of every incoming lane, ϑ is a normalization parameter, and q_l denotes the queue length of every lane. The second term denotes the total queue length of all lanes.

Simulation setup

To implement a complex training environment and sufficient realistic network layout, we set up a simulation based on a real intersection of the city of Jena in Germany in the microscopic software Simulation of Urban MObility (SUMO) (see Fig. 4). The peculiarity of this intersection is the multi-modal traffic, especially trams in the extra track require prioritization. In addition, the nearby tram stops "Paradiesbahnof" and "Paradiesbahnof West" generate a special challenge for the DDQL model to coordinate TLC and PT priority.



(a) OSM map of the intersection Knebelstr./Volksbad in Jena, Germany.

(b) SUMO simulation of a intersection Knebelstr./Volksbad in Jena, Germany.

Figure 4: Representation of the implemented intersection in real-world map and SUMO

The varying traffic demand at a urban intersection is approximated by a sine function, with the traffic demand on morning and evening being the high peaks and at midday and night being the low peaks. We therefore simplify dynamic traffic demand function to (3).

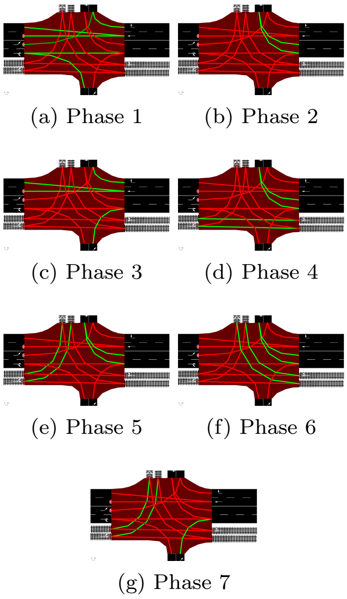
$$Demand = BaseFlow \times (1 + sin^2(\frac{t}{AddFrequency_T})) \tag{3}$$

In (3) index T indicates the type of vehicle, *BaseFlow* indicates the basic traffic flow at an intersection and $AddFrequency_T$ is the frequency of adding new vehicles in road network. The settings of *BaseFlow* and $AddFrequency_T$ are show in Tab. 1.

Table 1: Demand parameters

T	**MIT**	**PT**
BaseFlow	20 to 50 vehicles/hour	6 vehicles/hour
$AddFrequency_T$	1800 or 3600 or 7200 vehicles/day	3600 vehicles/day

To simulate the traffic light control, we use the original signal phases and their transitions which are relevant for the observed traffic and provided by the city administration of Jena. Thus, we implemented seven different signal phases (see Fig. 5) and accordingly 6 × 7 transitions. The main idea is that our phase controlling system could be superimposed to an existing local traffic light control, without any reconfiguration and evaluating the whole signalized intersection. Hence, We hard coded legal requirements, such as minimum green time, clearing times and maximum blocking time, firmly in the source code to form the preset phase duration and technical evaluations keep valid, and the presented model is dynamically selecting the phase duration and next phase.



(a) Phase 1 (b) Phase 2

(c) Phase 3 (d) Phase 4

(e) Phase 5 (f) Phase 6

(g) Phase 7

Figure 5: Traffic Light Phases

3 RESULTS AND DISCUSSION

Compared to previous researches based on simplified traffic scenarios, we present the performance of our DDQL model in a more realistic and complex traffic environment and preliminary results. The agent of our DDQL model is able to learn to control a complex signalized intersection and reduce the waiting time of MIT as well as PT. The represented results are achieved with the reward parameter shown in Tab. 2.

In Fig. 6 we compare, as preliminary results, the developed DDQL model to the original traffic control on two parallel running simulations of the same intersection with the same characteristics

Table 2: Reward parameters

	Car	**Bus**	**Tram**
η	0.001	0.01	0.03
ρ	2	2	2
C	60	10	5
ϑ	0.01	0.01	0.01

and traffic flows. Different vehicles are generated randomly. In the first subfigure, the total accumulated reward is depicted over an simulation of one hour. The second subfigure shows the reward calculated by (2) for each simulation step. The third and fourth subfigures show the waiting time for the PT and MIT. And the fifth subfigure illustrates the total amount of vehicles on all incoming lanes of the controlled intersection. Preliminary results show that for MIT and PT the waiting time is reduced, and the agent controlled intersection gains in total a higher system scores. The results we have so far are promising and we hope to extend the model for validation in a more complex environment.

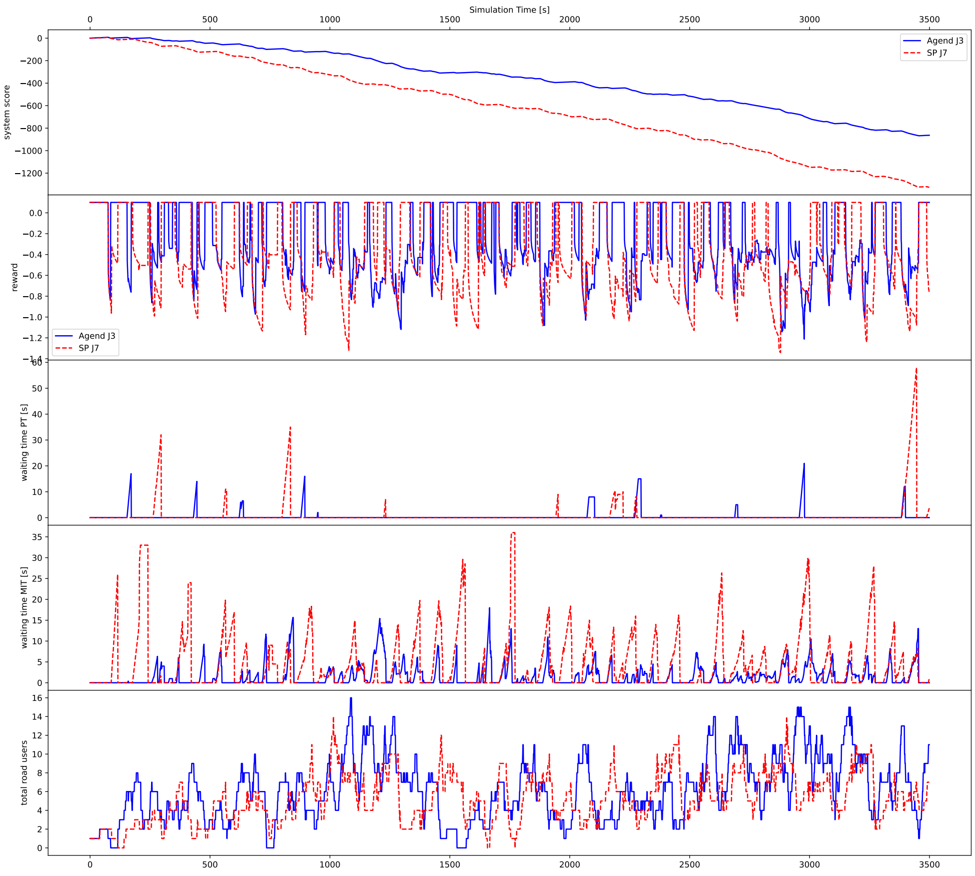


Figure 6: Comparison of the original traffic control of the city Jena (red dashed line) and traffic control via DDQL (blue line)

The proposed model adapts flexibly to the local traffic flows and could generate non-cyclic switching patterns respectively phase-skipping. This behavior is required particularly in situations with low traffic volumes and demonstrates the advantage of our model over other controls.

4 CONCLUSIONS

In this paper, we propose to solve the traffic light control problem using a deep reinforcement learning model. The traffic information is gathered from the road network, and the original traffic light information is provided by the city administration of Jena. The state space includes two layers of vehicle position and type, and each layer is two-dimension values that consists the index of every incoming lane and the length of lane with partition in 500 grids that each grid denotes $1m$. The actions are modeled as a Markov decision process, and the reward function is the negative cumulative waiting time and total queue length of all lanes with a normalization parameter. To handle the complex traffic scenario in our problem, we propose a Double Deep Q-Learning (DDQL) model with novel neural network architecture and experience replay buffer. The proposed model can learn a good policy under varying traffic demand, outperform the existing traffic light control system of the city of Jena in waiting time, which is shown in extensive simulation in SUMO and TensorFlow. In a next step of our work we will optimize the phase control even further including other types of road users such as pedestrians and cyclists.

5 CRediT AUTHORSHIP CONTRIBUTION STATEMENT

Tobias Nousch: Conceptualization, Formal analysis, Methodology, Simulation, Validation, Visualization, Writing - original draft & review & editing. **Runhao Zhou:** Conceptualization, Simulation, Validation, Visualization, Writing - original draft & review & editing. **Django Adam:** Project management, Traffic management consultation, Simulation, Wrting - review & editing. **Angelika Hirrle:** Project management, Writing - review & editing. **Meng Wang:** Supervision, Writing - review & editing.

6 DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7 ACKNOWLEDGEMENT

This research is supported by the project "**5G-basierte V2X-Vernetzung zur Steigerung der Verkehrssicherheit sowie zur Optimierung des multimodalen Verkehrs und der Energieversorgung in Jena**" funded by the German Federal Ministry for Digital and Transport.

References

Christofa, E., Papamichail, I., & Skabardonis, A. (2013). Person-based traffic responsive signal control optimization. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1278-1289. doi: 10.1109/TITS.2013.2259623

Guo, G., & Wang, Y. (2021). An integrated mpc and deep reinforcement learning approach to trams-priority active signal control. *Control Engineering Practice*, *110*, 104758. Retrieved from https://www.sciencedirect.com/science/article/pii/S0967066121000356 doi: https://doi.org/10.1016/j.conengprac.2021.104758

Guo, M., Wang, P., Chan, C., & Askary, S. (2019). A reinforcement learning approach for intelligent traffic signal control at urban intersections. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 4242-4247.

He, Q., Head, K. L., & Ding, J. (2014). Multi-modal traffic signal control with priority, signal actuation and coordination. *Transportation Research Part C-emerging Technologies*, *46*, 65-82.

Hu, J., Park, B. B., & Lee, Y.-J. (2016). Transit signal priority accommodating conflicting requests under connected vehicles technology. *Transportation Research Part C: Emerging Technologies*, *69*, 173-192. Retrieved from https://www.sciencedirect.com/science/article/pii/S0968090X16300614 doi: https://doi.org/10.1016/j.trc.2016.06.001

Hunt, P. B., Robertson, D. I., Bretherton, R. D., & Royle, M. (1982). The scoot on-line traffic signal optimisation technique. *Traffic engineering and control*, *23*.

Koonce, P., Rodegerdts, L. A., Lee, K., Quayle, S., Beaird, S., Braud, C., ... Urbanik, T. (2008). Traffic signal timing manual..

Liang, X., Du, X., Wang, G., & Han, Z. (2018). Deep reinforcement learning for traffic light control in vehicular networks. *ArXiv*, *abs/1803.11115*.

Lin, Y., Yang, X., Zou, N., & Franz, M. (2015). Transit signal priority control at signalized intersections: a comprehensive review. *Transportation Letters*, *7*(3), 168-180. Retrieved from https://doi.org/10.1179/1942787514Y.0000000044 doi: 10.1179/1942787514Y.0000000044

Long, M., Zou, X., Zhou, Y., & Chung, E. (2021). Deep reinforcement learning for transit signal priority in a connected environment. Retrieved from https://ssrn.com/abstract=3992999 doi: http://dx.doi.org/10.2139/ssrn.3992999

Lowrie, P. (1990). Scats: Sydney co-ordinated adaptive traffic system: a traffic responsive method of controlling urban traffic..

Ma, W., Liu, Y., & Yang, X. (2013). A dynamic programming approach for optimal signal priority control upon multiple high-frequency bus requests. *Journal of Intelligent Transportation Systems*, *17*(4), 282-293. Retrieved from https://doi.org/10.1080/15472450.2012.729380 doi: 10.1080/15472450.2012.729380

Pishue, B. (2021). *2021 inrix global traffci scorecard* (Tech. Rep.). Washington.

van der Pol, E., & Oliehoek, F. A. (2016). Coordinated deep reinforcement learners for traffic light control..

van Hasselt, H., Guez, A., & Silver, D. (2015). Deep reinforcement learning with double q-learning.

Wei, H., Zheng, G., Yao, H., & Li, Z. J. (2018). Intellilight: A reinforcement learning approach for intelligent traffic light control. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.