Distributed traffic light control at uncoupled intersections with real-world topology by deep reinforcement learning

Mark Schutera
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Research and Development
ZF Friedrichshafen AG
mark.schutera@kit.edu

Niklas Goby
Chair for Information Systems Research
University of Freiburg
IT Innovation Chapter Data Science
ZF Friedrichshafen AG
niklas.goby@is.uni-freiburg.de

Stefan Smolarek
IT Innovation Chapter Data Science
ZF Friedrichshafen AG
stefan.smolarek@zf.com

Markus Reischl
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
markus.reischl@kit.edu

Abstract

This work examines the implications of uncoupled intersections with local real-world topology and sensor setup on traffic light control approaches. Control approaches are evaluated with respect to: Traffic flow, fuel consumption and noise emission at intersections. The real-world road network of Friedrichshafen is depicted, preprocessed and the present traffic light controlled intersections are modeled with respect to state space and action space. Different strategies, containing fixed-time, gap-based and time-based control approaches as well as our deep reinforcement learning based control approach, are implemented and assessed. Our novel DRL approach allows for modeling the TLC action space, with respect to phase selection as well as selection of transition timings. It was found that real-world topologies, and thus irregularly arranged intersections have an influence on the performance of traffic light control approaches. This is even to be observed within the same intersection types (n-arm, m-phases). Moreover we could show, that these influences can be efficiently dealt with by our deep reinforcement learning based control approach.

1 Introduction

Traffic, abstracted onto the macro level, underlies complex, time-varying [20] and stochastic [38] dynamics. At the same time traffic management at intersections today is mostly conducted through rigid traffic regulations, such as right of way or pretimed traffic light signals. Ever increasing numbers of traffic participants result in a decreasing flow efficiency and other endemic traffic problems: Environmental pollution from the transport sector account to 15% of overall greenhouse gas emissions [16, 21]. Long-term average levels of noise, in particular from traffic, cause harmful effects on human health [12, 29]. Finally there is the problem of decreased traffic flow [13]. As of today, these issues are addressed by developing autonomous vehicles [17, 25] or deploying intelligent infrastructures [11, 13], especially in the domain of traffic light control (TLC) [1].

Preprint. Work in progress.

2 Related Work and Problem Statement

As of today, local TLC, directed at uncoupled intersections, has been addressed by various approaches, which contribute to the progress in the area of dictating conflicting traffic movements efficiently. In the following strategies are distinguished between: Pretimed control, which is based on fixed control schemes that are predetermined offline, usually modeled by Webster's formula [36]. Actuated control, in which based on traffic detectors a predefined control scheme is selected. Adaptive control, in which based on traffic detector data the priority is given to specific lanes (phase). Common schemes are MOVA [35] and CRONOS [6]. However these approaches evince characteristic drawbacks [9]. The pretimed controller lacks of adaptation with respect to changes in traffic volume, due to the fixed cycle-time. The actuated control overcomes this hindrance being able to switch between predefined control schemes, which are rarely ideal for the traffic flow demand at hand. In comparison, adaptive control strategies follow an optimization algorithm that determines the control action based on the entire intersection state. Hereby, adaptive strategies are prone to be incapable of real-time deployment. In current approaches [39], distributed controllers are deployed at individual intersections, to increase real-time adaptation. Further, the state-action pairs are cast into Q-networks by data driven modeling through Deep Reinforcement Learning (DRL). TLC through DRL has proven to be operative and competitive on real-world data [19, 24, 34, 37].

State-of-the-art approaches neglect the topology of real-world intersections, addressing plain four-arm intersections with regular geometry, symmetric paths and uniform path lengths within the intersection and the road network. Further, the state space is often enhanced far above the actual detection capabilities of commonly deployed real-world sensor-setups, such as induction loop detectors (ILD). Increased traffic flow demand can be met by adapting the road network infrastructure itself (network reconstruction, road extension, roundabouts, etc.). However this is more than often not applicable at intersections with irregular topology, due to opposing boundary conditions (limited construction area). What is more, irregular topologies add to the complexity of the optimization problem of TLC schemes. Consequently, before anything else TLC needs to be approached with respect to irregular topologies under real-world conditions.

The aim of this study is thus to examine the implications of real-world topologies and sensor-setups on the performance of DRL TLC approaches. For this purpose the following stages have been undertaken:

- Real-world intersection topologies are integrated into the micro-traffic simulation.
- A DRL model is deployed in order to cope with local real-world intersection topologies.
- Ours and baseline TLC-approaches are evaluated with respect to traffic flow metrics.

In Section 3, the used framework and environment are outlined. Our method is presented in Section 4 and the experimental results are depicted in Section 5. A conclusion is given in Section 6.

3 Framework

3.1 Micro-Traffic Simulation of the Roadnetwork Friedrichshafen

The road network is modeled as a bidirected graph with nodes (intersections) and edges (road segments). For learning the TLC policy and training our DRL algorithm, we utilize the Simulation of Urban Mobility (SUMO) [22]. SUMO is a free and open traffic simulation suite, which allows simulation of intermodal traffic (vehicles, pedestrians, etc.) and infrastructure (e.g. traffic lights). SUMO provides flexible APIs for road network design, traffic volume simulation and traffic light control.

The area of interest for the investigations is based on the real-world roadnetwork of the German city Friedrichshafen (see Fig. 1). The sensor setup is based on ILDs, which are as of today the most commonly used means of measuring traffic in real-world applications, as is the case in Friedrichshafen. ILDs are installed in each lane of the intersection. An ILD samples, with frequency f, the number (flow) and duration (occupancy) of vehicles passing over [5, 22].

The traffic flow has been generated following a probability distribution: In each simulation step a vehicle is instantiated with probability 0.2 on a random lane. The vehicle are deployed with

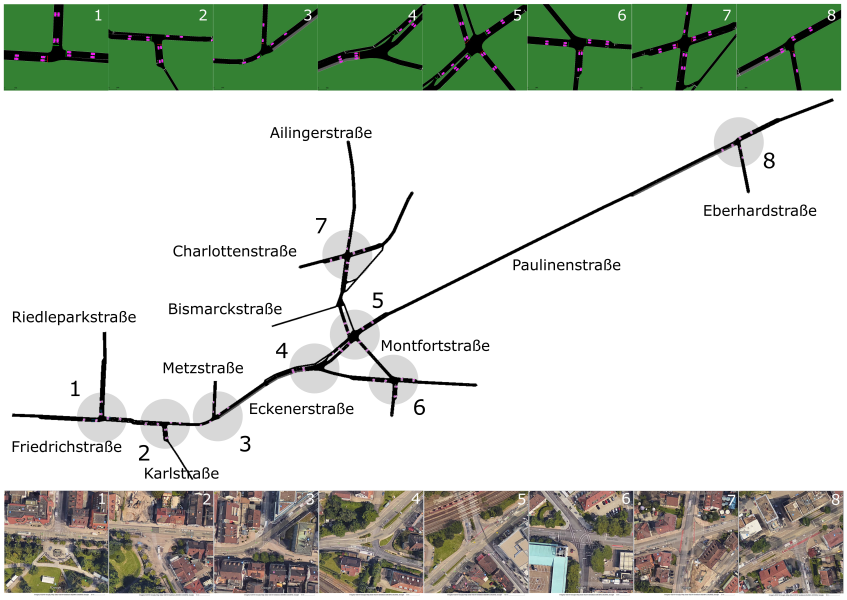

Figure 1: Overview of the Friedrichshafen roadnetwork with streetnames and the locations of the considered junctions in the center. In the top row the junctions are displayed as being present in SUMO. On the bottom row the google maps visualizations of the intersections themselves are shown.

homogeneous characteristics: $maxVelocity = 50\,km/h$, $acceleration = 3\,m/s^2$, $deceleration = 3\,m/s^2$, $minGap = 2.5\,m$, $vehicleLength = 5\,m$)

3.2 Traffic Light Control Approaches

The basic unit of TLC strategies, are non-conflicting vehicle movements linked together to phase groups [10]. Vehicle movements are enabled or suppressed by traffic light signals ($Green$: Move with priority, $green$: Move without priority, $orange$: must yield, red: must stop) [8]. A phase group definition for n traffic lights, thus consists of n signals. As outlined in Sec. 2, there are a variation of TLC strategies. In the following three are considered as baseline for benchmarking, the implementation can be found within the TraCI python package[1].

Fixed-time control, generates a fixed cycle with a default cycle time (here: $31s$), for all traffic lights [22]. All green phases are followed by an orange clearing phase of 6s. Left-turns are allowed (move without priority) at the same time as oncoming straight traffic.

Gap-based control, is common in Germany and works through extending phase durations over $31s$ when a continuous stream of traffic is detected ($< 3s$) or abbreviating phase durations when a sufficient time gap ($> 3s$) in the stream is detected (within duration limits: $[6, 45]s$) [22]. This affects cycle duration in response to dynamic traffic conditions. This state space is determined by ILDs.

Time-based control, extends phase durations according to the presence of delayed vehicles [22]. For delay detection a detector that covers the full range of a lane is necessary (e.g. camera, extended ILDs). The delay of a vehicle is defined as $1 - \frac{v}{v_{max}}$, where v is its current velocity and v_{max} the allowed maximal velocity on the lane. Once the $timeLoss$ oversteps the tolerated loss ($1s$), the corresponding $Green$ phase is extended by $timeToPass$, as such that the vehicle is able to pass the junction.

4 Deep Reinforcement Learning for Traffic Light Control

Deep reinforcement learning (DRL) combines the two frameworks deep learning (DL), for representation learning and reinforcement learning (RL), for learning to take actions in an initially

[1] TraCI python package: https://github.com/eclipse/sumo/tree/master/tools/traci

unknown environment [31]. This new generation of algorithms has recently achieved human like results in mastering complex tasks in model-free settings with a very large state space and with no prior knowledge [27, 28, 32]. For more background on RL, see [33, 18, 2], for background on DL, see [14], and for an overview of DRL, see [23].

Following the general reinforcement learning setup, at each time-step t, the agent observes state s, extracted from the SUMO environment and takes action a (i.e. the next phase configuration) according to a learned ϵ-greedy strategy and receives reward r. The goal of the learning agent is to find a policy that maximizes the cumulative reward over: $R_t = \sum_{j=0}^{\infty} \gamma^j r_{t+j+1}$.

State Space Due to the irregular intersection geometry, the state space of the DRL problem needs to be variable in order to tackle the irregularity. For each intersection, we determine a set of all admissible phase configurations P, and a set L of all induction loop detectors of all lanes. Each phase $p_k \in P$ can be uniquely assigned via an index k. Given these two sets, the state space S for each intersection geometry is defined as tuple $s \in S$ and $s = (k, l_1, \ldots, l_{|L|})$ where k represents the index of the intersections current traffic light phase p_k , and l_i the mean occupancy of the last time step in percent of the ith induction loop detector. The resulting state provided for the agent is a stack consisting of the last $\tau = 10$ observed states $s_{t-\tau+1}, ..., s$.

Action Space The action space A consists of integer values from the interval $[0, \ldots, |P| - 1]$. An action $a \in A$ corresponds to the phase transition from the current phase to phase p_a. To overcome crashes and emergency breaks of vehicles, we introduce an orange-phase for each lane to transition from green to red signals. This is done automatically and cannot be influenced by the agent.

Reward Function The reward function plays an important role in learning the agent's behavior. Different reward functions lead to different, mostly unwanted, policies. Focusing solely on the delay or the waiting time as proxy for the average travel time of vehicles results in either a mixture of jammed and high speed lanes or flickering traffic lights, as shown by [34]. Therefore, they came up with a combination of these measures and included penalties for teleportation, emergency stops, and indicators for configuration changes [34]. A similar reward function is also used in [37]. Here, the authors added penalties for the queue length, the total number of vehicles and the travel time of vehicles that passed the intersection. Motivated by promising results of these two works, our one-step reward r is calculated by the sum of the average delay d per lane, defined as $1 - \frac{v}{v_{max}}$, the average waiting time per lane w, a teleportation penalty e and a flickering penalty f. The final one-step reward signal $r_t \in [-1; 0]$ at time step t is given by iterating over all lanes of the map and computing penalties, where i is the lane index:

$$r_t = -0.1f - 0.1e - 0.4 \sum_{i=1}^{L} d_i - 0.4 \sum_{i=1}^{L} w_i, \tag{1}$$

where L is the number lanes in the network. The coefficients of the reward (see Eq. 1) are based on coefficients, which have been successfully deployed in similar studies [34, 37].

Learning strategy The agent uses the Rainbow algorithm introduced in [15]. Rainbow is based on the DQN algorithm [27, 28] which has been supplemented with additional components such as n-step Bellman updates [26], prioritized experience replay [30] and distribution reinforcement learning [4].

5 Experiments

5.1 Implementation and Evaluation

The experiments are developed in python using Google Dopamine [3] for implementing the agent and running the experiments and OpenAI gym [7] as framework for creating the simulation environment that interacts with the agent. Each model has been trained on a NVIDIA K80 GPU or a NVIDIA M60 GPU, for approximately ten hours.

Hyperparameters for the training process and the architecture have been based on previous studies, such as: The rainbow DRL approach [15], the C51 hyperparameter configuration [4] and default values introduced by the dopamine framework [3]. The Q-network architecture is motivated by the DQN-architecture presented in [28] and adapted to the present environment.

Table 1: Hyperparameters of the architecture and the training process of our DRL TLC approach. Hyperparameters not explicitly mentioned are set according to dopamine [3] or the C51 configuration presented in [4].

Architecture Q-Network		
Name	Value	Description
Input Layer	$len(s) \cdot \tau$	Flattened
1. Hidden Layer	$2 \cdot len(s) \cdot \tau$	Fully connected, ReLU
2. Hidden Layer	$2 \cdot len(s) \cdot \tau$	Fully connected, ReLU
Output Layer	$\|A\|$	Fully connected
Training		
Name	Value	
Epsilon (Greedy policy) ϵ	0.05	
Initial learning rate	0.0000625	
Epsilon (Adam optimizer)	0.00015	
Number of iterations	100	
Training steps	36000	
Max. steps per episode	3600	

The baseline approaches, as well as our approach, are evaluated and compared with respect to minimizing $timeLoss$. It is assumed, that every vehicle desires to reach its destination with the least time expenditure possible. The $timeLoss$ is defined as the time expended due to driving below the $maxSpeed$. Thus, slowdowns due to stops at intersections will incur $timeLoss$ [22].

5.2 Results and Discussion

The approaches have been evaluated within SUMO, by running them on a test-sequence of 3600 simulation steps respectively. By comparing our results with the presented baseline methods with respect to the $timeLoss$ (see Sec. 4), it is observed that (see Fig. 2):

The investigated real-world topologies show a critical influence on the DRL TLC performances (visible in both ours and the baseline). This is for one expressed by the differences of the mean $timeLoss$, ranging from $4 - 11$ seconds on intersections with the same amount of phases and arms (e.g. J1, J2, J3, J4), yet differing topologies. It is to be remarked that as of today the study of real-world topologies, with respect to DRL TLC, has been neglected and simplified within state-of-the-art research.

The interquartile range of our approach is smaller than six seconds, depicting the ability of our approach to have a smoothing effect on the majority of vehicle traffic flows. With the exception of the junction J4, our method clearly outperforms the baseline approaches by a difference in mean $timeLoss$ of five seconds at least. It is to be remarked, that our approach suffers from outliers, implying that the traffic flow is optimized at the expense of single vehicles. Nevertheless, those outliers are mostly within the range of the results achieved by the baseline approaches.

6 Conclusion

The influence of real-world topologies at uncoupled urban intersections in the presence of distributed TLC, has been investigated with respect to $timeLoss$. It was observed that topologies have a crucial impact on DRL TLC performance, yet being neglected in state-of-the-art research. It was demonstrated, that distributed TLC strategies are applicable at uncoupled intersections with real-world topology. A beneficial effect of such strategies with respect to the $timeLoss$, has been shown compared to baseline algorithms. Also to be highlighted is the novel modeling of the DRL TLC action space, which allows for a phase selection rather than the mere selection of transition timings, latter being prevalent in the state-of-the-art. SUMO is a vast simplification of real-world traffic. Introducing pedestrians and cyclists to the environment, as well as emulating traffic flow with respect to real-world data, is thus considered the next step to get closer to real-world conditions. Future work

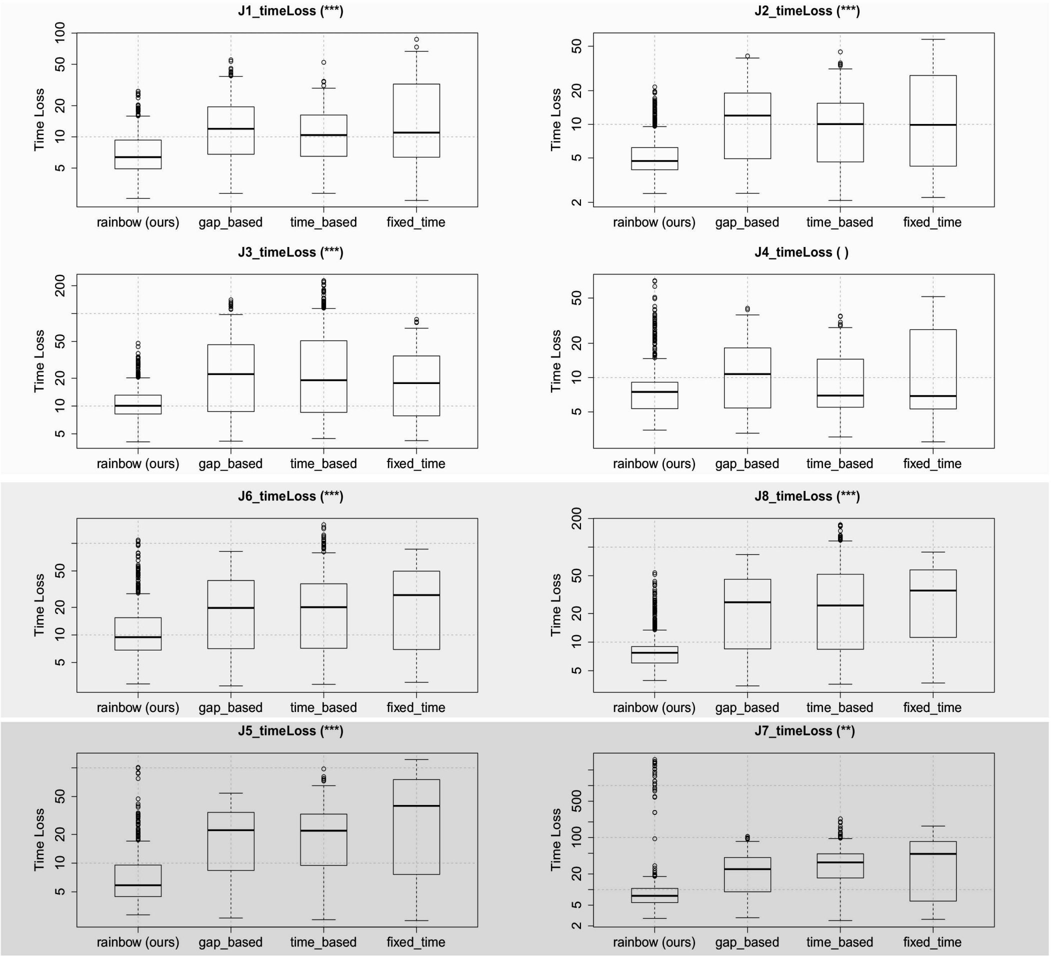

Figure 2: Evaluation with respect to the $timeLoss$ the vehicles experience due to the traffic light control at the junction (in seconds on a log-scale). The different shades mirror the characteristics of the junction J whereas the number stands for the junctions ID (see 1), from top to bottom: three-arm junction with four phases, three-arm junction with six states and four-arm junctions with eight states. The results of our Welch two sample t-test indicate that there is a statistically significant difference between the mean $timeLoss$ for our approach and the baseline approaches at nearly all junctions, displayed for each intersection next to the figure title.

should also investigate the influence of differing sensor setups and thus extensions of the observed state space of the DRL TLC. This might include adding information about daytime, date, current weather or road friction, up to an omniscient state perception. It is to be remarked that there is no guarantee, that the sum of locally optimized strategies also leads to a global optimum when being coupled. Our ongoing work thus seeks to investigate the implications of coupled intersections and the interaction between distributed systems. A deeper understanding of underlying action patterns of the DRL TLC should contribute to further functional validation, that's why we suggest a future action space pattern analysis. Judging from our results, it can be noted that the reward function matters most. Further analysis aiming at reducing outliers is highly recommended.

Acknowledgments

We thank Dr. Jochen Abhau from Research and Development, as well as the whole Data Science Team at ZF Friedrichshafen AG, for supporting this research. Thank you for all the assistance and comments that greatly improved this work. We would also like to express our gratitude to Prof. Dr. Ralf Mikut from the Institute for Automation and Applied Informatics, Karlsruhe Institute of

Technology, and Prof. Dr. Dirk Neumann for providing insight and expertise that greatly enhanced our research.

References

[1] Sahar Araghi, Abbas Khosravi, and Douglas Creighton. A review on computational intelligence methods for controlling traffic signal timing. *Expert System Application*, 42(3):1538–1550, February 2015.

[2] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[3] Marc G. Bellemare, Pablo Samuel Castro, Carles Gelada, Saurabh Kumar, and Subhodeep Moitra. Dopamine. *github.com*, 2018.

[4] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[5] Peter J. Bickel, Chao Chen, Jaimyoung Kwon, John Rice, Erik van Zwet, and Pravin Varaiya. Measuring traffic. *Statistical Science*, 22(4):581–597, 11 2007.

[6] F. Boillot, S. Midenet, and Jc Pierrelee. *The real-time urban traffic control system CRONOS: Algorithm and experiments*, volume Vol 14, Issue 1. Elsevier, 2006.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.

[8] J.C. Chedjou and K. Kyamakya. Cellular neural networks based local traffic signals control at a junction/intersection. *IFAC Proceedings Volumes*, 45(4):80 – 85, 2012. 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control.

[9] Jean Chamberlain Chedjou and Kyandoghere Kyamakya. A review of traffic light control systems and introduction of a control concept based on coupled nonlinear oscillators. In *Recent Advances in Nonlinear Dynamics and Synchronization*, pages 113–149. Springer, 2018.

[10] Department for Transport. General principles of traffic control by light signals. Traffic advisory leaflets, UK Government, 2009.

[11] G. Dimitrakopoulos and P. Demestichas. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1):77–84, March 2010.

[12] European Parliament and of the Council of 25 June 2002. Directive 2002/49/ec relating to the assessment and management of environmental noise. Directive, European Parliament, 2002.

[13] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira, and J. L. Martins de Carvalho. Towards the development of intelligent transportation systems. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pages 1206–1211, Aug 2001.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, Cambridge, Massachusetts and London, England, 2016.

[15] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017.

[16] International Transport Forum. Reducing transport greenhouse gas emissions: Trends and data 2010. Research report, OECD, 2010.

[17] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *CoRR*, abs/1704.05519, 2017.

[18] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[19] Nishant Kheterpal, Kanaad Parvate, Cathy Wu, Aboudy Kreidieh, Eugene Vinitsky, and Alexandre Bayen. Flow: Deep reinforcement learning for control in sumo. In Evamarie Wießner, Leonhard Lücken, Robert Hilbrich, Yun-Pang Flötteröd, Jakob Erdmann, Laura Bieker-Walz, and Michael Behrisch, editors, *SUMO 2018- Simulating Autonomous and Intermodal Transport Systems*, volume 2 of *EPiC Series in Engineering*, pages 134–151. EasyChair, 2018.

[20] R.M. Kimber and P.N. Daly. Time-dependent queueing at road junctions: Observation and prediction. *Transportation Research Part B: Methodological*, 20(3):187 – 203, 1986.

[21] Daniel Krajzewicz, Michael Behrisch, Peter Wagner, Raphael Luz, and Mario Krumnow. Second generation of pollutant emission models for sumo. In Michael Behrisch and Melanie Weber, editors, *Modeling Mobility with Open Data*, pages 203–221, Cham, 2015. Springer International Publishing.

[22] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - simulation of urban mobility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[23] Yuxi Li. Deep reinforcement learning: An overview. *CoRR*, abs/1701.07274, 2017.

[24] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. Deep reinforcement learning for traffic light control in vehicular networks. *CoRR*, abs/1803.11115, 2018.

[25] Schutera Mark, Goby Niklas, Neumann Dirk, and Reischl Markus. Transfer learning versus multi-agent learning regarding distributed decision-making in highway traffic. In *Proc. of the 10th International Workshop on Agents in Traffic and Transportation (ATT 2018), co-located with ECAI/IJCAI, AAMAS and ICML 2018 conferences*, volume 2129, pages 57–62, 2018.

[26] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.

[27] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[28] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[29] Renez Nota, Robert Barelds, and Dirk van Maercke. Harmonoise WP 3 Engineering method for road traffic and railway noise after validation and fine-tuning. Technical Report Deliverable 18, HARMONOISE, 2005.

[30] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2015.

[31] David Silver. ICML 2016 Tutorial: Deep reinforcement learning, 2016.

[32] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[33] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. Adaptive computation and machine learning. MIT Press, 1998.

[34] Elise Van der Pol and Frans A Oliehoek. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, 2016.

[35] R.A. Vincent, J.R. Peirce, Transport, Road Research Laboratory, Transport, and Road Research Laboratory. Traffic Management Division. *MOVA: Traffic responsive, self-optimising signal control for isolated intersections*. Research report: Transport and Road Research Laboratory. Traffic Management Division, Traffic Group, Transport and Road Research Laboratory, 1988.

[36] F.V. Webster. *Traffic Signal Settings*. Road research technical paper. H.M. Stationery Office, 1958.

[37] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery 38; Data Mining*, KDD '18, pages 2496–2505, New York, NY, USA, 2018. ACM.

[38] S.C. Wong. Group-based optimisation of signal timings using the transyt traffic model. *Transportation Research Part B: Methodological*, 30(3):217 – 244, 1996.

[39] Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys*, 50(3):34:1–34:38, June 2017.