Smart Traffic Light Control using Actor-Critic algorithms

Asmina Nassar, Het Darshan Mehta and Shalini Pal
Otto-von-Guericke-University, Magdeburg, Germany
Digital Engineering
asmina.nassar@st.ovgu.de, het.mehta@ovgu.de, shalini.pal@st.ovgu.de

Keywords: Adaptive Traffic Signal Control, Deep Reinforcement Learning, Actor-Critic algorithms, Advantage Actor-Critic (A2C)

1 INTRODUCTION

The continuous increase in urban population and the subsequent rise in social and economic activities have significantly escalated the demand for transportation in metropolitan areas. However, existing traffic systems are insufficient to accommodate the growing volume of vehicles, leading to persistent traffic congestion that disrupts daily mobility and undermines urban efficiency [4]. Traffic congestion arises from multiple factors, and its negative impacts extend beyond road blockages and prolonged travel times. It also contributes to vehicular exhaustion and leads to unnecessary fuel consumption, further straining environmental and economic resources [2]. These issues not only affect infrastructure but also have a profound impact on the daily lives of individuals, making the mitigation of traffic congestion a critical area of concern for policymakers and researchers alike.

Potential solutions to mitigate the road congestion issues should be compatible enough to be integrated with the existing infrastructure without requiring huge and new investments [7][15][2]. One such cost-effective solution is Adaptive Traffic light controller. It is a strategy in which the traffic signal light and it's frequency adapts to the real-time traffic condition such as – number of approaching vehicles, number of waiting vehicles to optimize parameters such as total waiting time for a vehicle at an intersection [9][14][17].

Various AI algorithms, such as reinforcement learning, deep learning, fuzzy control, heuristic methods, genetic algorithms, and immune network algorithms, have been employed by researchers to address the traffic signal control problem, leveraging their ability to optimize in real-time [27][8]. Heuristic methods were deemed unsuitable because they rely on precise assumptions and mathematical models, which are impractical for large-scale, dynamic road networks [8][27]. Additionally, the stochastic nature of traffic patterns and the complex interactions between adaptive traffic signals make preprogrammed behaviours ineffective for managing such systems [2]. Similarly, fuzzy control was dismissed due to its reliance on extensive expert knowledge to create effective fuzzy rule tables, which is neither practical nor sustainable [27].

The need for a learning mechanism that finds the global optimal solution through interaction with the stochastic traffic environment makes RL the most appropriate option [24][20]. Unlike traditional systems that rely on predefined rules or training data (input-output pairs), RL-based traffic signal controllers learn through interaction with their environment (traffic intersections) without any human intervention [1][3][22][21][18][2]. The traffic signals controllers (agent) receive feedback in the form of a scalar reward, which indicates how effective the controller's actions were in achieving its goal (e.g., minimizing vehicle waiting times). The goal of the RL-based traffic signal controller is to maximize the total rewards over time. By experimenting with different signal timing plans, the controller learns to select those that yield the highest cumulative rewards in the long run. Over time, they build an understanding of which actions are most likely to yield better outcomes. This adaptability makes the system responsive to dynamic traffic conditions. The value function serves as a critical tool for estimating the long-term effectiveness of actions, helping the system make informed decisions to optimize traffic flow [24].

Traditional RL typically uses simple models like

piecewise constant tables and linear regression (LR) [9], which limit scalability and optimality in real-world applications. Now, deep neural networks (DNNs) can be integrated with RL to improve the learning capacity of more complex tasks[8]. Deep neural networks excel at extracting and transforming raw data into useful and high-dimensional features that can enable RL algorithms to perform effectively [19].

To leverage the capabilities of deep reinforcement learning (RL), suitable RL methods must be adopted. The three main approaches are: policy-based (Actor-Only), value-based (Critic-Only) and Actor-Critic methods.
Policy-based methods optimize a parameterized policy directly, which allows the generation of continuous actions, a significant strength in certain environments. However, the high variance in gradient estimation makes learning slow [2][16]. This method updates the policy using sampled episode returns and operates in an on-policy manner, where transitions are nonstationary during episodes due to continuous policy updates. That's why policy-based methods are flexible but less efficient in terms of data usage [8].

Value-based methods focus exclusively on approximating the value function, without explicitly defining a policy. While they offer the advantage of low variance in return estimates, determining the optimal action for each state is computationally intensive, particularly in environments with continuous action spaces [2]. Since these methods are off-policy, updates are performed using bootstrapped experience replay and rely on one-step temporal difference learning [25]. However, successful convergence depends on the assumption of stationary Markov Decision Process (MDP) transitions, which may not be valid in dynamic settings of the adaptive traffic signal control [8].

Actor-critic methods combine policy-based and value-based approaches, reducing bias and gradient variance by integrating a critic to parameterize the value function [16]. This method has the advantages of both policy-based and value-based methods. They produce continuous actions and adapt smoothly to changing states [12][2]. This makes them particularly effective for tasks requiring adaptive and continuous control.

In this study, actor-critic methods have been applied to traffic signal control for the Cologne and Ingolstadt topologies. A comprehensive explanation of the actor-critic framework is presented in Section 2, while Section 4 covers the description of the models, libraries, and the simulation software used. Finally, Section 5 analyzes the results for both topologies, with varying numbers of intersections.

2 BACKGROUND

2.1 Deep Reinforcement learning

Reinforcement learning (RL) is a branch of machine learning that diverges from the conventional supervised and unsupervised learning paradigms. Unlike these methods, RL does not depend on labeled or unlabeled datasets. Instead, an agent (learner) learns by interacting with its environment, transitioning between states, and receiving scalar feedback in the form of rewards or penalties. This feedback mechanism enables the agent to iteratively refine its strategy (optimizing the policy) that maximizes cumulative rewards.
When reinforcement learning is combined with neural networks, the latter acts as the agent that interacts with the environment. Neural networks take an input representation of the current state and parameterize actions in the output as a function of the continuous state space. These neural networks serve to approximate one of the following key functions:

- **Policy** ($\pi(a \mid s)$): A mapping from states to actions, which defines the behavior of the agent.
- **Q values** ($Q(s,a)$): A value function estimating the expected cumulative reward for taking a specific action in a given state.

Different models have been designed to approximate these functions, with the primary objective being to select the best possible action. To achieve this, the neural network processes the input state and produces one of two possible outputs:

- **Action Probabilities**: A probability distribution over all possible actions.
- **Action Values (Q-values)**: Numerical estimates of the expected rewards for each action in the given state.

The agent then selects its next action based on these outputs. It can choose the action with the highest probability of generating the maximum cumulative reward, as indicated by the action probabilities. Alternatively, it may select the action with the highest Q-value, which represents the expected cumulative reward starting from the current state, taking the specific action, and following the policy thereafter.

2.2 Actor-Critic Model

The Actor-Critic model combines both policy-based and value-based methods to leverage the strengths of

each approach. In the context of Deep Reinforcement Learning, the model employs two neural networks: one for the **actor** (policy-based) and another for the **critic** (value-based).

- **Actor**: The actor generates a probability distribution over the action space and selects the action with the highest probability, effectively representing the policy.
- **Critic**: The critic evaluates the value of the current state-action pair, estimating the inherent value of the state based on the action taken by the actor.

The critic's value estimation is used to compute the **Advantage function**, which quantifies how good/worse the chosen action is compared to the average actions. The value estimate serves as a baseline to aid in the calculation of the advantage function.

$$\begin{aligned} A(s,a) &= Q(s,a) - V(s) \\ \implies A(s,a) &= r + \gamma V(s') - V(s) \end{aligned} \tag{1}$$

Here, a denotes action, s denotes state, $A(s,a)$ is the advantage function, r is the reward, γ is the discount factor, $V(s)$ is the actual value estimate and $V(s')$ is the predicted value estimate.

The model employs loss function to train both the networks. The actor uses the policy loss to guide the agent toward taking actions that maximize the advantage, while the critic uses the value loss to ensure accurate value estimation, thereby stabilizing the advantage calculation. Gradient descent is performed to minimize the Temporal Difference (TD) error in value estimation.
Through this process, the actor network iteratively optimizes the policy. Over time, the actor learns from the outcomes of its actions, gradually improving and converging toward an optimal policy.

3 RELATED WORK

Adaptive Traffic Signal Control (ATSC) is a widely researched topic, with significant advancements over time. Early studies primarily utilized traditional RL models for policy optimization, with tabular Q-learning being the first RL algorithm applied to an isolated intersection [26][8]. More recently, Deep RL methods have been explored, mostly with Deep Q-learning. However, many of these studies rely on impractical assumptions or oversimplified traffic environments, limiting their real-world applicability [8].
The use of Multi-Agent Reinforcement Learning (MARL), which is particularly relevant for traffic light control problems, has been relatively limited. Most MARL studies in this domain focus on Deep Q-Learning. Among the limited literature on actor-critic methods applied to traffic signal control, the following works are notable.
[8] proposed the Multi-Agent A2C (MA2C) model to stabilise the convergence of the Independent A2C (IA2C). The study was conducted on synthetic Monaco traffic data showed. MA2C outperformed all other methods and demonstrated robustness, optimality and scalability. [11] examined various state representations forming a spectrum of current traffic data resolutions. The implementation of the A3C algorithm revealed minimal performance differences across the various state representations. The study concludes that high-resolution data is likely to perform well only when paired with advanced neural network architectures. [6] performed a comparative analysis of Deep Q-Learning (DQN), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Advantage Actor-Critic (A2C) for vehicle traffic control in Webots World and SUMO simulations. The study found that A2C outperformed the other models, demonstrating superior performance compared to fixed-time controllers. [5] investigated the application of three on-policy reinforcement learning algorithms—A2C, Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO). Comparative performance evaluations revealed that A2C demonstrated reliable convergence and stability under changing traffic dynamics, whereas TRPO and PPO failed to converge and stabilize.

To summarize, even though ATSC has been extensively researched studies on Actor Critic methods are limited. A2C has demonstrated good performance in various simulation environments and shown reliable convergence under dynamic traffic constraints. However, there is a lack of research addressing the practical, real-world implementation of Deep RL methods, for traffic signal control.

4 METHODOLOGY

This section details the methodology adopted for the implementation of adaptable traffic light control utilizing an Actor-Critic model. The "Actor" component proposes actions based on the current policy, dynamically adjusting traffic light operations to enhance flow and reduce congestion at signalized intersections. Meanwhile, the "Critic" assesses the quality of the action taken by the Actor by estimating the value function of the state after the action is taken. This integrated method improves adaptability and learning efficiency by simultaneously updating policy and value assessments, offering more precise

adjustments and faster convergence than Q-learning or pure policy-based methods alone.[16][13]
Our evaluation explores the model's capacity to dynamically adjust traffic light phasing in response to real-time traffic variations, aiming to improve coordination across multiple intersections and thus minimizing vehicle queues. [13]

4.1 Reinforcement learning variables

We formulated the problem in a reinforcement learning framework with 3 main components including: state, action, reward.

4.1.1 State Definition

The state of each environment at any given intersection is defined by several key metrics: the number of waiting vehicles per lane, lane densities indicating vehicle concentration, queue lengths to assess congestion levels, vehicle speeds to gauge traffic flow and average travel time as the global parameter.

4.1.2 Action Space

In the traffic management system , the action set for each intersection comprises the following four traffic signal phases: vehicles traveling straight along the North-South axis, those traveling straight along the East-West axis, left-turning vehicles from the North-South direction, and left-turning vehicles from the East-West direction. At any given time step, the RL agent can choose between maintaining the current traffic signal phase or switching to the next phase.
We used SUMO (Simulation of Urban Mobility) as simulation software for this study. In SUMO, the phases are defined in a particular convention. For instance for 16 lanes the phase will be represented as *'rrrrrGGGggrrrrrGGGgg'*.This shows that the intersection geometry has an influence on the phases.

For our implementation below 4 phases were considered for 1 specific intersection:

'rrrrryyyggrrrrryyygg',
'rrrrrrrryyrrrrrrrryy',
'yyyggrrrrryyyggrrrrr',
'rrryyrrrrrrrryyrrrrr'

However, this combination of phases might differ for every intersection according to the lanes connected to the available action. Safety measures such as the inclusion of a yellow light phase ensure that transitions between actions do not compromise vehicle safety [23].

4.1.3 Reward Function

The reward function is calculated in terms of speed reward and vehicle throughput reward.

4.2 A2C Model Architecture

The proposed model for the Advantage Actor-Critic (A2C) smart traffic control system leverages a deep neural network architecture optimized for efficient decision-making in large state spaces.[13] The model comprises a shared feature extraction layer followed by two distinct actor and critic networks.

4.2.1 Shared Network

The shared feature extraction layers are the backbone of the model [10], processing the input state, which encapsulates various traffic metrics such as the number of waiting vehicles and vehicle speeds at each intersection. We have defined a shared network with 5 hidden layers with 256 as input and output dimensions. These layers use LeakyReLU activation functions to enhance non-linear feature extraction, allowing the model to capture complex patterns and dependencies in the input data.

4.2.2 Actor and Critic Networks

The actor network is tasked with generating a probability distribution over possible actions, which correspond to different traffic signal configurations. It uses a softmax layer to ensure that the output probabilities are normalized. Conversely, the critic network evaluates the actions taken by the actor by estimating the value of the state from which the actions were taken. This network outputs a scalar value representing the expected cumulative future rewards from the current state, guiding the actor towards more beneficial actions [23].
Both networks are initialized with Xavier uniform weight initialization to promote convergence and stability during training.

4.2.3 Loss fuction

The model's training objective combines policy loss, value loss, and an entropy regularisation term to encourage exploration and prevent premature convergence. The policy loss is computed using the log probabilities of actions weighted by their corresponding advantages, ensuring that the network prioritises high-reward actions. The value loss, computed as the mean squared error between predicted state values

and observed rewards, ensures accurate value function predictions. An entropy coefficient further promotes diversity in action selection, dynamically balancing exploration and exploitation. Finally an aggregate of these losses was used for the gradient update.

4.2.4 Optimizer

The entire system is optimized using SharedAdam optimizer, which facilitates efficient parameter updates and suits environments with shared parameters across multiple threads or processes. In our implementation, the optimizer accumulates gradients from multiple agents and uses them to adjust the global parameters (avg_travel_time). This ensures smooth and stable updates even in high-dimensional and noisy environments.

5 RESULTS

Here, we are showcasing the results for Cologne and Ingolstadt data with various junctions. The graphs depict the waiting time and global reward calculation for each topology.

5.1 Ingolstadt1

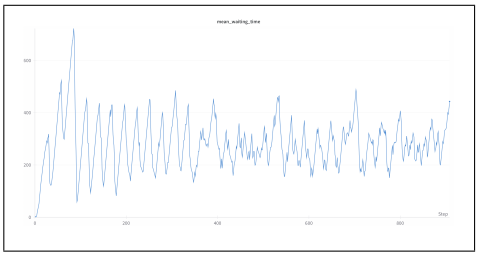

Figure 1: Waiting Time - Ingolstadt1

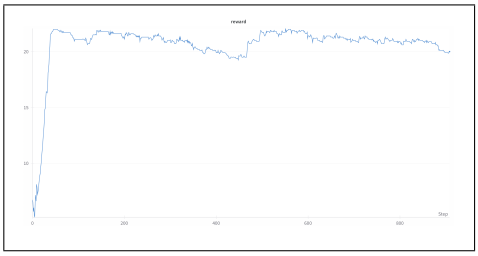

Figure 2: Reward - Ingolstadt1

Figure 1, which plots the change in waiting time for 1 intersection, shows high fluctuations in the beginning, but eventually, the waiting time stabilizes with no downward trend. Figure 2 shows the reward trend with a huge spike, highlighting the model's fast learning. However, then it fluctuates a little without any improvements in the results(i.e. mean waiting time). We can say that the results are not fully satisfactory and also require further tuning for better visualization and insights.

5.2 Cologne3

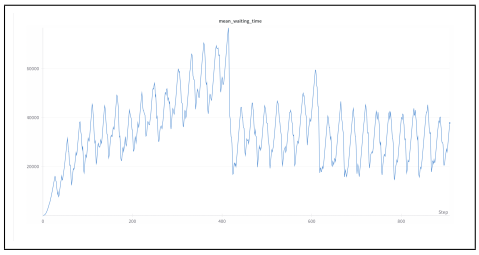

Figure 3: Waiting Time - Cologne3

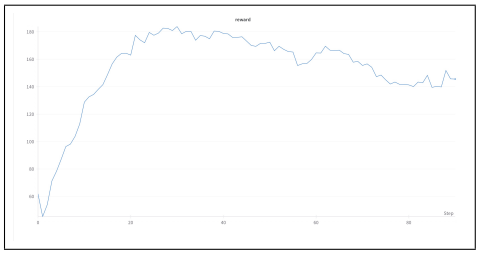

Figure 4: Reward - Cologne3

Figure 3 shows a very high value for wait time and exhibits a downward trend over the training steps. There are fluctuations, but the waiting time stabilises and decreases significantly as the time steps increase. In Figure 4, we get a promising reward curve, although only initially. Following that, we can observe a slight dip in the curve. From this we can conclude that if the model is trained for more iterations the waiting time can be further reduced, the reward will eventually converge and attain stability.

5.3 Cologne8

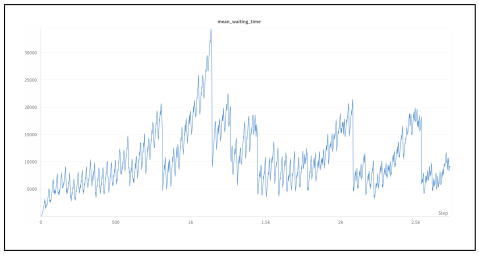

Figure 5: Waiting Time - Cologne8

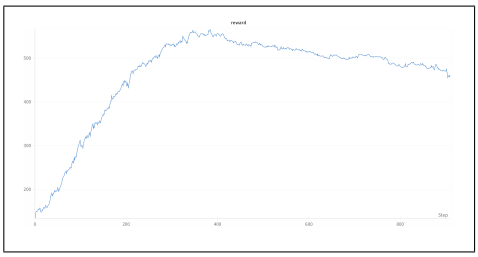

Figure 6: Reward - Cologne8

The results in Figures 5 and 6 show partial success in reducing waiting times and improving rewards. However, it is evident that the policy is not yet robust enough to handle 8 intersections effectively. The slightly declining reward and small fluctuations in waiting times suggest the model is not performing well in optimizing results for more intersections that could be optimized using advanced methods (e.g. MARL).

5.4 Ingolstadt21

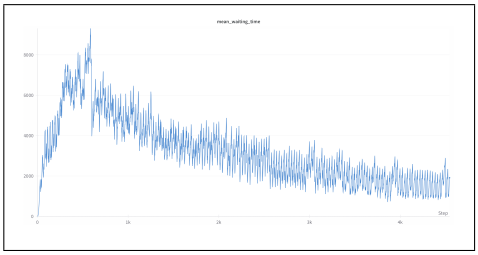

Figure 7: Waiting Time - Ingolstadt21

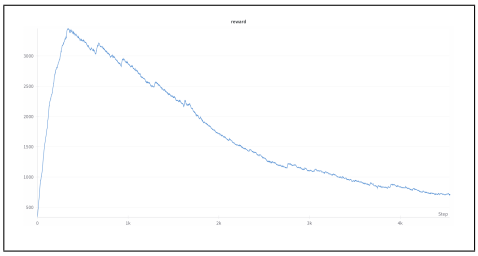

Figure 8: Reward - Ingolstadt21

The significant reduction in waiting time(Figure 7) is a success, which shows the agent's ability to optimize traffic flow across 21 intersections. However, the declining reward in Figure 8 indicates that the system is not achieving an optimal balance between minimizing waiting times and maintaining reward.

6 CONCLUSION

Based on the observations and results obtained, the A2C model demonstrates effectiveness in reducing waiting times across various traffic topologies. This indicates that Actor-Critic methods are capable of addressing congestion challenges in traffic signal control. However, the results also reveal certain limitations as the problem size increases, such as a decrease in learning efficiency, delayed convergence, and, at times, a gradual decline in the reward, which is influenced by vehicle throughput and speed.
Interestingly, we observed a contradictory scenario where waiting times decreased, yet the expected increase in vehicle throughput and speed, and consequently the reward, was not achieved. Possible explanations for this outcome include:

- Waiting time is not a sufficient metric for reward calculation. While waiting time decreased, the throughput depended on the number of vehicles processed. For example, if only one vehicle is on a lane, the throughput remains low despite reduced waiting time, whereas for a larger volume of vehicles, throughput improves irrespective of waiting times.Thus, waiting time alone is not an efficient metrics for calculating reward.
- The reduction in waiting time may favor local optimizations, which can adversely affect overall network throughput and efficiency.
- Conflicting optimization objectives: The goals of minimizing waiting time, maximizing throughput, and maintaining high speed are not always perfectly aligned, leading to trade-offs in traffic scenarios.

These findings suggest that while A2C is promising, future work should explore more comprehensive reward metrics and strategies to balance conflicting optimization objectives, particularly in large-scale traffic networks.

7 FUTURE SCOPE

To address the observed limitations and further improve the handling of traffic congestion using smart traffic lights, the following enhancements are proposed as part of future work:

- Incorporating additional Metrics such as lane density and queue length for Reward calculation. The current reward function relies primarily on waiting time, which, as observed, may not always align with throughput or overall traffic efficiency. Future work will incorporate additional metrics to provide a more comprehensive and precise measure of traffic flow

and ensure a more balanced optimization of traffic objectives.
- Increasing the training steps (>100k) to allow the model to capture the nuances of complex traffic dynamics more effectively. Longer training can provide more robust policy learning and deeper insights during visualization.
- Implementation of A3C. Observations indicate that junctions with fewer vehicles contribute less to the overall reward, thereby affecting the mean reward of the system. Implementing A3C can address this issue by enabling local agents to share their experiences with a global network. The global network updates its parameters and synchronizes the policies of all local agents, facilitating more effective coordination and scaling across the network. This approach can yield a more balanced performance across all junctions, even in scenarios with uneven traffic distribution.
- Hyperparameter Optimization for temperature parameter. Fine-tuning the temperature improved our current implementation. Thus, tuning it more with the new metrics and model will yield better and optimal results, balancing exploration and exploitation.

References

[1] Baher Abdulhai and Lina Kattan. "Reinforcement learning: Introduction to theory and potential for transport applications". In: *Canadian Journal of Civil Engineering* 30.6 (2003). Cited by: 92, 981 – 991.

[2] Mohammad Aslani, Mohammad Saadi Mesgari, and Marco Wiering. "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events". In: *Transportation Research Part C: Emerging Technologies* 85 (2017), pp. 732–752. ISSN: 0968-090X.

[3] Ana L. C. Bazzan. "Opportunities for multi-agent systems and multiagent reinforcement learning in traffic control". In: *Autonomous Agents and Multi-Agent Systems* 18.3 (2009), pp. 342–375. ISSN: 1573-7454.

[4] Basudeb Bhatta. *Analysis of urban growth and sprawl from remote sensing data*. Springer Science & Business Media, 2010.

[5] Sultan Kübra Can et al. "Traffic light management systems using reinforcement learning". In: *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE. 2022, pp. 1–6.

[6] Rudolph Joshua Candare and Junrie Matias. "Evaluating Deep Reinforcement Learning Methods to Develop an Intelligent Traffic Controller". In: *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*. IEEE. 2022, pp. 173–178.

[7] Mashrur A Chowdhury and Adel Wadid Sadek. *Fundamentals of intelligent transportation systems planning*. Artech House, 2003.

[8] Tianshu Chu et al. "Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control". In: *IEEE Transactions on Intelligent Transportation Systems* 21.3 (2020), pp. 1086–1095.

[9] Samah El-Tantawy, Baher Abdulhai, and Hossam Abdelgawad. "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto". In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013). Cited by: 391, 1140 – 1150.

[10] Omar Elharroussa et al. "Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches". In: *Journal of Artificial Intelligence and Data Science* (2023).

[11] Wade Genders and Saiedeh Razavi. "Evaluating reinforcement learning state representations for adaptive traffic signal control". In: *Procedia computer science* 130 (2018), pp. 26–33.

[12] Ivo Grondman et al. "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 1291–1307.

[13] Jaun Gu et al. "Traffic Signal Optimization for Multiple Intersections Based on Reinforcement Learning". In: *Applied Sciences* 11.22 (2021), p. 10688.

[14] S.M.A. Bin Al Islam and Ali Hajbabaie. "Distributed coordinated signal timing optimization in connected transportation networks". In: *Transportation Research Part C: Emerging Technologies* 80 (2017), pp. 272–285. ISSN: 0968-090X.

[15] Franziska Klügl and Ana LC Bazzan. "Agent-based modeling and simulation". In: *Ai Magazine* 33.3 (2012), pp. 29–29.

[16] Vijay R. Konda and John N. Tsitsiklis. “OnActor-Critic Algorithms”. In: *SIAM Journal on Control and Optimization* 42.4 (2003), pp. 1143–1166. eprint: https://doi.org/10.1137/S0363012901385691.

[17] Wanjing Ma, Kun An, and Hong K. Lo. “Multistage stochastic program to optimize signal timings under coordinated adaptive control”. In: *Transportation Research Part C: Emerging Technologies* 72 (2016), pp. 342–359. ISSN: 0968-090X.

[18] Patrick Mannion, Jim Duggan, and Enda Howley. “An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control”. In: *Autonomic Road Transport Support Systems*. Ed. by Thomas Leo McCluskey et al. Cham: Springer International Publishing, 2016, pp. 47–66. ISBN: 978-3-319-25808-9.

[19] Volodymyr Mnih et al. “Asynchronous Methods for Deep Reinforcement Learning”. In: *CoRR* abs/1602.01783 (2016). arXiv: 1602.01783.

[20] Martijn van Otterlo and Marco Wiering. “Reinforcement Learning and Markov Decision Processes”. In: *Reinforcement Learning: State-of-the-Art*. Ed. by Marco Wiering and Martijn van Otterlo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–42. ISBN: 978-3-642-27645-3.

[21] Cenk Ozan et al. “A modified reinforcement learning algorithm for solving coordinated signalized networks”. In: *Transportation Research Part C: Emerging Technologies* 54 (2015), pp. 40–55. ISSN: 0968-090X.

[22] Baher Abdulhai Samah El-Tantawy and Hossam Abdelgawad. “Design of Reinforcement Learning Parameters for Seamless Application of Adaptive Traffic Signal Control”. In: *Journal of Intelligent Transportation Systems* 18.3 (2014), pp. 227–245. eprint: https://doi.org/10.1080/15472450.2013.810991.

[23] Qi-Wei Sun et al. “Deep Reinforcement-Learning-Based Adaptive Traffic Signal Control with Real-Time Queue Lengths”. In: *Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan* (2021). Corresponding Author: Shi-Yuan Han.

[24] Richard S Sutton. “Reinforcement learning: An introduction”. In: *A Bradford Book* (2018).

[25] Christopher J. C. H. Watkins and Peter Dayan. “Q-learning”. In: *Machine Learning* 8.3 (1992), pp. 279–292. ISSN: 1573-0565.

[26] Marco Wiering et al. “Intelligent traffic light control”. In: *Institute of Information and Computing Sciences. Utrecht University* (2004).

[27] Jianyou Xu et al. “An Improved Traffic Signal Control Method Based on Multi-agent Reinforcement Learning”. In: *2021 40th Chinese Control Conference (CCC)*. 2021, pp. 6612–6616.