OPTIMIZING URBAN TRAFFIC FLOW: REINFORCEMENT LEARNING-BASED TRAFFIC LIGHT CONTROL

DEMETRE DZMANASHVILI

Thesis supervisor: ANAIS GARRELL ZULUETA (Department of Automatic Control)

Degree: Master Degree in Artificial Intelligence

Master's thesis

School of Engineering
Universitat Rovira i Virgili (URV)

Faculty of Mathematics
Universitat de Barcelona (UB)

Barcelona School of Informatics (FIB)
Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Abstract

Contents

1. Introduction

1.1 Overview

Traffic congestion is a persistent global issue, impeding daily commutes as a result of the ever-increasing urban population and transportation demands in cities worldwide [8][16]. One major contributor to this problem is the delay caused by red lights at intersections, where traffic signals typically operate on fixed-time schedules regardless of actual traffic conditions [12]. While such systems are effective in heavily congested areas, they often prove inefficient for low traffic density scenarios, resulting in unnecessary delays and fuel wastage [12].

Recent technological advancements have introduced the Adaptive Traffic Signal Control System, which utilizes sensors embedded in roads to synchronize traffic signals, thus responding to real-time traffic conditions [8]. However, this system's feasibility and cost-effectiveness have been questioned due to the need for embedded road infrastructure and power sources [8]. Additionally, optimizing traffic signal control to minimize delays while ensuring system stability remains a challenge [12].

This thesis aims to address these challenges by proposing a Traffic Control System based on reinforcement learning (RL), an artificial intelligence framework that learns optimal decision policies through continuous adaptation to real-time traffic scenarios. By moving away from fixed-time schedules and incorporating RL, we seek to develop an intelligent traffic control system that efficiently manages traffic flow, reduces environmental impact, such as air pollution and fuel wastage, and enhances road safety [12]. The research focuses on a 4-way intersection, analyzing incoming traffic density to optimize traffic signal control and improve overall transportation efficiency over time.

1.2 Motivation

My personal motivation for undertaking this thesis is deeply rooted in the traffic problems I have witnessed in my home country, Georgia. The congestion and inefficiency of traffic lights on some of the busiest streets in Georgia have long been a source of frustration for me and my fellow citizens. The resulting traffic jams not only waste valuable time but also contribute to environmental issues such as increased air pollution and fuel wastage. Additionally, the risk of accidents rises as traffic congestion worsens.

Driven by these challenges, I have chosen to focus on optimizing traffic light control, specifically targeting one of the busiest streets in Georgia, as the subject of my thesis. By harnessing the power of reinforcement learning, I aim to develop an intelligent traffic control system capable of dynamically managing traffic flows, reducing congestion, and addressing environmental concerns. Through this research, I aspire to contribute to more efficient and sustainable urban transportation systems, not only in Georgia but also as a model for cities worldwide.

2. Background

2.1 Reinforcement Learning

Reinforcement Learning (RL) is a paradigm in which an agent learns to make decisions by interacting with its environment. In the RL framework, the environment is often modeled as a Markov decision process (MDP), characterized by key components:

- S – the state space,
- A – the action space,
- $P(s_t, a, s_{t+1})$ – the transition function, mapping from state s_t and action a to the next state s_{t+1} with probabilities in the range [0, 1],
- $R(s, a)$ – the reward function, which assigns a real-valued reward to each state-action pair,
- γ – the discount factor, controlling the trade-off between immediate and future rewards.

The RL agent operates based on a policy π, which maps states to actions, i.e., $\pi : S \to A$. When the agent selects an action a_t in the current state s_t, it impacts the environment, leading to a new state s_{t+1} and an immediate reward r_t.

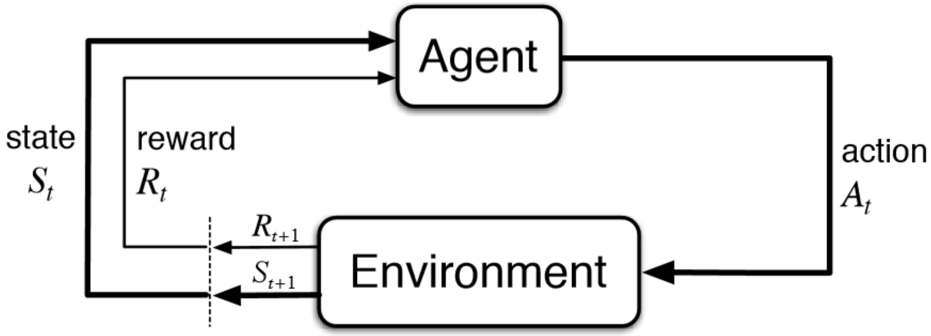

Figure 1: Reinforcement Learning Framework

The primary objective of the RL agent is to maximize the expected sum of discounted rewards, denoted as $J^\pi = \sum_{t=0}^{\infty} \gamma^t r_t$. The optimal policy, denoted as π^*, is the one that maximizes this objective.

There are various approaches for training a policy using RL:

- **Value-Based Approach:** This approach focuses on estimating the expected future utility from states (state value) or from action-state pairs (action value or q-value). The control policy is then directed towards actions or states that maximize the expected utility (J^π). A prominent example is the model-free deep Q-learning algorithm [11].

- **Policy-Gradient Approach:** In this approach, a policy is defined through a parameterized differential equation, and the parameters are updated incrementally following the policy gradient. These updates aim to achieve favorable outcomes as measured by the reward function. Estimations of state or action values are often used to define these favorable outcomes. This approach is commonly referred to as an actor-critic approach.
- **Actor-Critic Approach:** Actor-critic methods combine elements of both value-based and policy-gradient approaches. An actor (policy) learns to make decisions, while a critic (value function) evaluates these decisions. A state-of-the-art example of an actor-critic algorithm is the proximal policy optimization (PPO) algorithm [14].

These RL approaches provide a framework for training intelligent agents to make decisions in complex and dynamic environments, making them highly relevant to optimizing traffic signal control in urban settings.

2.2 Traffic Signal Control as an MDP

In the realm of traffic engineering, a signalized intersection represents a complex network of incoming and outgoing roads, each comprising one or more lanes. To efficiently manage traffic flow at such intersections, a set of phases, denoted as Φ, is defined. Each phase, $\varphi \in \Phi$, corresponds to a specific traffic movement through the intersection, as illustrated in Figure 2. It's crucial to note that two phases are considered conflicting if they cannot be simultaneously enabled due to intersecting traffic movements.

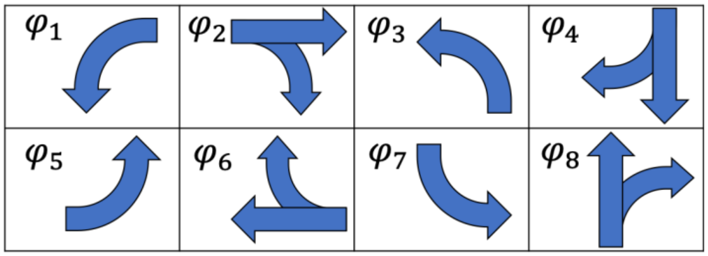

Figure 2: Example of Phases at a Signalized Intersection[3]

At each discrete time step, a signal controller is tasked with selecting a combination of non-conflicting phases to enable. The objective is to optimize a long-term objective function, which may vary depending on specific goals and constraints. In the context of Reinforcement Learning (RL)-based controllers, the signalized intersection environment is commonly modeled as a Markov Decision Process (MDP), with the following components:

- **State Space (S):** The state space encompasses the state of incoming traffic and the currently enabled phases. The definition of the state varies among studies, reflecting differing sensing capabilities. Some works assume state-of-the-art traffic sensing technologies, providing high-resolution data on incoming traffic, including information such as the number of approaching vehicles, accumulated waiting time, the number of stopped vehicles, and the average speed of approaching vehicles [4]. Others adopt less informative sensing capabilities, such as observing only the stopped queue length per lane [10] or solely the waiting time of the first vehicle in the queue [15].
- **Action Space (A):** In each time-step, the controller selects a set of non-conflicting phases to be assigned the right-of-passage (green light). If the chosen phases are different from the currently enabled ones, a mandatory yellow phase is enforced ny the system for a predefined duration. It's important to note that assigning yellow phases is not the part of the action space, it is a constraint imposed by the environment.
- **Transition Function (P):** The transition function describes the progression of traffic following the signal assignment. This progression can be defined within a simulated environment, as

commonly done in research [10], or based on real-world traffic progression in practical implementations.

- **Reward Function (R):** The reward function serves as a critical component in RL-based signal control. Different reward functions have been proposed in the literature. Commonly used reward functions include (minus) queue length summed over all incoming lanes [17], (minus) total delays imposed by the intersection [15], (minus) waiting time at the intersection [10], and (minus) traffic pressure [4]. These reward functions reflect various aspects of traffic performance and congestion alleviation.

The modeling of traffic signal control as an MDP provides a foundation for applying RL techniques to optimize signal operation, ultimately contributing to more efficient and adaptive traffic management strategies.

2.3 Evaluation environments for RL-based signal controllers

According to [3], previous research in the field of traffic signal control has often relied on custom-made scenarios tailored for evaluating specific Reinforcement Learning (RL) algorithms. For instance, Jinming and Feng (2020) utilized the well-established Simulation of Urban Mobility (SUMO) environment for their experiments. SUMO enjoys widespread acceptance within the transportation community and serves as a reasonable testbed choice for such studies. However, it's worth noting that Jinming and Feng's reported scenario, based on the real-world city of Monaco, was a modified version. This modified scenario included 18 synthetic traffic signals beyond the official "Monaco SUMO Traffic (MoST)" scenario and incorporated non-validated inflated traffic demands [6].

Another notable simulation testbed, CityFlow, was presented by Zhang et al.[18]. However, CityFlow has two primary limitations. Firstly, unlike SUMO, CityFlow lacks rigorous calibration and evaluation within the general transportation community. Although it claims to produce equivalent output as SUMO, this claim is primarily based on results from simplified grid network scenarios. Secondly, while CityFlow offers the Manhattan, New York network as a common benchmark scenario, the support for this scenario's representation of real-world city layouts and demands is limited.

Additionally, some relevant publications have conducted evaluations using the Autonomous Intersection Management (AIM) simulator. The primary drawback of the AIM simulator lies in its lack of traffic scenarios based on real-world cities. AIM typically generates simple grid networks with symmetric intersections. While one might draw parallels between such grid networks and the road layout in Manhattan, New York, a more in-depth analysis of traffic trends is needed to substantiate such claims and their relevance to the real world [13][7][10].

3. Related Work

3.1 Reinforced Signal Control (RESCO)

In this section, we review related work in the field of traffic signal control, with a focus on the Reinforced Signal Control (RESCO) toolkit, which serves as a baseline for our research.

The RESCO toolkit is a standard Reinforcement Learning (RL) traffic signal control testbed designed to achieve several key objectives:

1. Provide benchmark single and multi-agent signal control tasks based on well-established traffic scenarios.
2. Offer an OpenAI GYM interface within the testbed environment to facilitate the deployment of state-of-the-art RL algorithms.
3. Deliver a standardized implementation of state-of-the-art RL-based signal control algorithms.

RESCO is open-source and freely available under the GNU General Public License 3. It is built on top of SUMO-RL [1] and can be accessed on GitHub at `github.com/Pi-Star-Lab/RESCO`. The embedded traffic scenarios within RESCO have their own licensing, with Cologne-based scenarios under Creative Commons BY-NC-SA and Ingolstadt-based scenarios under the GNU General Public License 3.

State and Action Space

RESCO accommodates a wide range of sensing assumptions, including advanced sensing capabilities [6]. Users can select subsets of state features based on specific sensing assumptions. Features include information such as stopped vehicles' queue length, the number of approaching vehicles, total waiting time for stopped vehicles, and more, at the level of state, intersection, and lane. Additionally, users can define the effective sensing distance during initialization.

The action space in RESCO encompasses sets of non-conflicting phase combinations, following the methodology described in Section 2.2 of the RESCO documentation [6]. By default, actions are chosen for the next 10 seconds of simulation, with the first 3 seconds reserved for yellow signals, if necessary.

Reward Metrics

RESCO offers flexibility in terms of reward metrics. Users can designate any of the reward metrics defined in Section 2.2 of the RESCO documentation[6] or create custom weighted combinations of these metrics. When initializing a control task, users can pass a weight vector that assigns weights to different metrics in the reward function. These weights correspond to various aspects, such as system travel time, signal-induced delays, total waiting time at intersections, average queue length, and traffic pressure.

Benchmark Control Tasks

The signal control benchmark tasks in RESCO are based on two well-established SUMO scenarios: "TAPAS Cologne" and "InTAS" [13, 9]. These scenarios represent traffic within real-world cities, namely, Cologne and Ingolstadt in Germany. They include road network layouts and calibrated demands, making them suitable for comprehensive evaluation. RESCO defines three benchmark control tasks for each traffic scenario:

1. Controlling a single main intersection.
2. Coordinated control of multiple intersections along an arterial corridor.
3. Coordinated control of multiple intersections within a congested area (downtown).

Benchmark Algorithms

RESCO provides three baseline controllers and several RL-based controllers for comparative evaluation:

1. **Baseline Controllers**:
   (a) Fixed-time (Pre-timed) control, where phase combinations are enabled for fixed durations following predefined cycles.
   (b) Max-pressure control, which selects the phase combination with the maximum joint pressure. [4]
   (c) Greedy control, which chooses the phase combination with the maximum joint queue length and approaching vehicle count.[10]
2. **RL Controllers**:
   (a) IDQN (Independent DQN agents), employing convolutional layers for lane aggregation[2].
   (b) IPPO, which utilizes a deep neural network similar to IDQN[2].
   (c) MPLight, based on the FRAP open-source implementation, ChainerRL DQN, and pressure sensing[19].
   (d) Extended MPLight (MPLight*), an enhanced version of MPLight with additional sensing information.
   (e) FMA2C, built on top of the MA2C open-source implementation[5].

In each of the RL-based controllers, specific learning algorithms and hyperparameters are applied, allowing for a comprehensive evaluation of their performance [2, 4, 5, 10, 19].

4. State of the Art

5. Methodology

5.1 Comparison with Previous Work

Our research builds upon the RESCO toolkit, utilizing its benchmark tasks, state-of-the-art RL controllers, and well-established traffic scenarios. By conducting experiments with our own map and configuration for traffic control, we aim to contribute to the growing body of knowledge in the field of RL-based traffic signal control.

6. Experiments

7. Comparison

8. Conclusion

9. Future Work

References

[1] L. N. Alegre. Sumo-rl, 2019.

[2] J. Ault, J. Hanna, and G. Sharon. Learning an interpretable traffic signal control policy. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2020)*. International Foundation for Autonomous Agents and Multiagent Systems, May 2020.

[3] James Ault and Guni Sharon. Reinforcement learning benchmarks for traffic signal control. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track*, December 2021.

[4] Chacha Chen, Hongyu Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yilin Xiong, Kewei Xu, and Zongzhang Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3414–3421, 2020.

[5] T. Chu, J. Wang, L. Codecà, and Z. Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095, 2019.

[6] L. Codeca and J. Härri. Monaco sumo traffic (most) scenario: A 3d mobility scenario for cooperative its. In *SUMO 2018, SUMO User Conference, Simulating Autonomous and Intermodal Transport Systems, May 14-16, 2018, Berlin, Germany*, 2018.

[7] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research*, 31:591–656, 2008.

[8] D. M. Levinson. Speed and delay on signalized arterials. *Journal of Transportation Engineering*, 124(3):258–263, 1998.

[9] S. C. Lobo, S. Neumeier, E. M. Fernandez, and C. Facchi. Intas–the ingolstadt traffic scenario for sumo, 2020.

[10] Jian Ma and Fan Wu. Feudal multi-agent deep reinforcement learning for traffic signal control. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 816–824, 2020.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[12] S. S. Mousavi, M. Schukat, and E. Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7):417–423, 2017.

[13] T. T. Pham, T. Brys, M. E. Taylor, T. Brys, M. M. Drugan, P. Bosman, M.-D. Cock, C. Lazar, L. Demarchi, and D. Steenhoff. Learning coordinated traffic light control. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-13)*, volume 10, pages 1196–1201, 2013.

[14] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[15] S. M. A. Shabestary and Baher Abdulhai. Deep learning vs. discrete reinforcement learning for adaptive traffic signal control. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 286–293, 2018.

[16] A. Tirachini. Estimation of travel time and the benefits of upgrading the fare payment technology in urban bus services. *Transportation Research Part C: Emerging Technologies*, 30:239–256, 2013.

[17] Marco A. Wiering. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pages 1151–1158, 2000.

[18] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. Li. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference*, pages 3620–3624, 2019.

[19] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, and Z. Li. Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1963–1972, 2019.