Multiagent Reinforcement Learning for Traffic Signal Control

Kevyn Kelso

University of Colorado-Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80919
kkelso@uccs.edu

Abstract

Transportation is a complex problem facing developed societies. It is riddled with logistical challenges extending beyond pollution and traffic congestion. The idea of applying reinforcement learning to transportation problems is not new but has great potential for improvement and research in the field. Contrary to current approaches, reinforcement learning can make decisions in stochastic and uncertain environments characteristic of traffic problems. Many common reinforcement learning approaches have been applied, showing promise over other approaches, and will be compared below. This project aims to reproduce existing methods and attempt to improve performance concerning the traffic throughput metrics described below. An integration of various ideas from the literature, combined to study what can be learned from reinforcement learning experimentation with regards to traffic control is the primary purpose.

Introduction

Traffic Signal Control (TSC), a classic control problem affecting all drivers, holds exciting potential for reinforcement learning to address obvious societal problems including air pollution, decreases in speed, delays, and opportunity costs (de Almeida, Bazzan, and Abdoos 2022). Based on travel research conducted in urban areas, 12-55% of total commute time is caused by signalized intersections (traffic lights) (Ault and Sharon 2021). Moreover, modern reinforcement learning (RL) approaches suggest a potential 73% reduction in that time compared to traditional approaches currently deployed (Ault and Sharon 2021). The field of traffic signal control contains many ideas to increase traffic efficiency in both the fixed algorithm and RL space. However, many advanced solutions come with expensive caveats. For instance, installing advanced sensors, reworking road paths, or replacing traffic lights with roundabouts (Alegre, Bazzan, and da Silva 2020). This project aims to study reinforcement learning approaches to TSC that can be deployed to existing metropolitan areas with minimal cost and infrastructure changes. In this paper, we will discuss the TSC problem in terms of the Markov Decision Process (MDP) paradigm; and how it is formally described. Existing methods will then be explored including what is deployed traditionally in most urban environments, and what RL methods could also be used. Then, our unique solution will be proposed based on information collected comparing a wide variety of methods used previously on the TSC problem (Hafiz and Bhat 2020) (Ault and Sharon 2021) (Ghanadbashi et al. 2023). Finally, a discussion of evaluation metrics, expected challenges, and the timeline of the project will be discussed.

Problem Formulation

Multiagent approaches to TSC are Partially Observable Markov Decision Processes (POMDPs) with non-stationary dynamics, placing them in one of the most difficult classes of problems for RL to solve (de Almeida, Bazzan, and Abdoos 2022) (Alegre, Bazzan, and da Silva 2020) (Choi, Yeung, and Zhang 1999). Non-stationary problems do not come with convergence guarantees of traditional MDPs and the partial observability can cause the agent to miss nuance in the state necessary for optimal policies (Choi, Yeung, and Zhang 1999) (Lee, Ganapathi Subramanian, and Crowley 2022). State aliasing is common in POMDPs where the agent finds two states that are different to be the same, resulting in inappropriate actions. Additionally, centralized agent approaches suffer from Bellman's curse of dimensionality, require unfeasible infrastructure modifications, and have shown suboptimal performance in simulation (Alegre, Bazzan, and da Silva 2020) (Ault and Sharon 2021).

State space

The state space for traffic signal control is a vector of various traffic-related parameters described below. In the TSC problem, the state space is typically modeled as 1 where each time step t corresponds to five seconds of actual traffic dynamics, ρ_1 and ρ_2 are binary variables $\rho_1, \rho_2 \in \{0, 1\}$ indicating the state of the intersection lights, g indicates if the light has been green for the minimum specified time, L is the list of all lanes with density Δ_l which is the number of vehicles in each lane divided by the capacity of that lane. q_l is the number of queued vehicles in each lane $l \in L$ (de Almeida, Bazzan, and Abdoos 2022).

$$s_t = [\rho_1, \rho_2, g, \Delta_1, \ldots, \Delta_L, q_1, \ldots, q_L] \quad (1)$$

In reality, few intersections contain the sensors needed to make up the state space described in 1, so it will be inter-

esting to explore how state space restriction affects agent performance.

Action space

The action space is best described in figure 1 where the green paths indicate where traffic can flow and the red paths are not allowed. The agent can change the intersection into one of four modes $a_t \in \{a_1, a_2, a_3, a_4\}$. There are two direct flow modes and two turning modes. In the direct flow modes North to South and East to West, the vehicles are also permitted to take right turns. To change the intersection into a different mode, a mandatory yellow phase ϕ precedes the mode change. In most simulation scenarios, ϕ = 2 seconds.

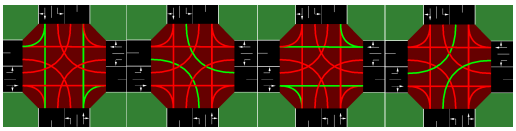

Figure 1: Traffic light modes (Alegre 2019)

Reward function

In most papers, the cumulative delay reward function 2 is used where D_t is the sum of all vehicles wait time 3. Other papers extend this idea to the sum of all vehicle wait times in the system to encourage cooperation between agents (Lee, Ganapathi Subramanian, and Crowley 2022). Furthermore, other sophisticated reward functions such as traffic pressure, augmented rewards based on intersection efficiency, and rewards based on ontology adherence have been explored (Ault and Sharon 2021) (Ghanadbashi et al. 2023). In this project, the cumulative delay reward function 2 will be used. However, the cumulative delay will be the sum of all vehicles in the system instead of at the intersection level, where V_t is the set of all vehicles present in the simulation. The idea behind this is to encourage the agents to work together rather than develop adversarial relationships.

$$r_t = D_t - D_{t+1} \tag{2}$$

$$D_t = \sum_{v \in V_t} d_t^v \tag{3}$$

Environment Simulator

The Simulated Urban MObility (SUMO) environment will be used for experimentation because it has been evaluated and calibrated by the general transportation community (Ault and Sharon 2021) (Alegre 2019) and is widely used in the literature. Additionally, SUMO was chosen for its compatibility with the OpenAI PettingZoo API which helps create a more standard environment interface for RL systems (Terry et al. 2021). A 4 x 4 grid structure 2 of traffic lights will be used as it is also widely used by other papers (de Almeida, Bazzan, and Abdoos 2022) and is easily understood. This will necessitate 16 independent agents. As a future work item, studying how agents trained on a 4 x 4 grid environment transfer to more complex environments such as the cities of Cologne or Ingolstadt will be interesting.

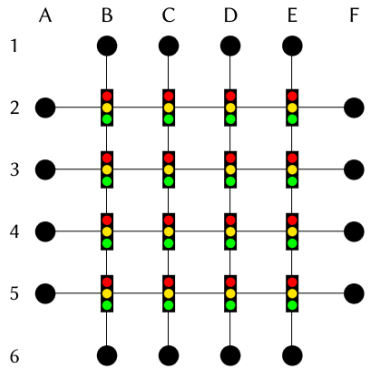

Figure 2: 4 x 4 intersection grid (Alegre, Bazzan, and da Silva 2020)

Existing Methods

Traditional Controllers

The traffic controllers currently used in most areas today fall into three categories of algorithms: fixed time, max pressure, and greedy. Fixed time controllers will set the traffic signal into one of the four modes for a fixed amount of time, cycling through each mode (Ault and Sharon 2021). Max pressure is the idea of selecting the traffic flow direction based on subtracting the number of cars after the intersection from the number of cars before the intersection (Chen et al. 2020). The greedy algorithm selects the mode that allows the direction with maximum queue length through.

RL Controllers

Most popular RL methods have been applied to TSC including Independent Deep Q-Learning Networks (IDQN), Independent Proximal Policy Optimization (IPPO), MPLight (variant of DQN) (Chen et al. 2020), Feudal Multiagent Advantage Actor-Critic (FMA2C, MA2C) (Chu et al. 2019), State Action Reward State Action Lambda $SARSA(\lambda)$, TD methods (Reza et al. 2023), Self Adaptive Systems (SAS), and ontology-based RL models (Ghanadbashi et al. 2023). Based on the results from each of these papers, IDQN appeared to have the most promising performance, and will be discussed further below.

Method	Average Cumulative Delay (seconds)
Fixed Time	90.00[2]
Max Pressure	70.00[2]
Greedy	60.00[2]
IDQN	30.74
IPPO	36.15
MPLight	54.58
MA2C	38.07[1]
FMA2C	42.26
$SARSA(\lambda)$	18.00[2]
Ontology	17.00[2]

Table 1: Average Delay Across All Scenarios

Proposed Method

Based on performance data collected in other studies (Ault and Sharon 2021) (Ghanadbashi et al. 2023), IDQN is the chosen approach to solve the TSC problem. IDQN is an off-policy, model-free RL method incorporating the idea of collecting experience tuples as the agent plays episodes. The experience is then used to train a deep neural network and is randomly shuffled to smooth the training data across many past behaviors (Mnih et al. 2013). The input to the network is the state information, and the output is a Q-value mapped to each action. In the past, non-linear Q-value function approximators (such as a deep neural network) were unsuccessful due to correlations in the sequence of states, and small updates to Q significantly changing the policy (Hafiz and Bhat 2020). Experience replay and other DQN methods were shown to significantly increase the chances of convergence and improve performance by an order of magnitude (Mnih et al. 2015). Additionally, simplifying the deep neural network to output only binary values of whether to take the action or not yielded better performance than mapping a Q-value to each action. This is called binary DQN or CS-DQN (Hafiz and Bhat 2020). Based on this information, a binary DQN agent for each traffic signal intersection will be used to solve the 4x4 grid TSC environment. An off-policy action selection method known as ϵ-greedy will be used, where ϵ is the probability the agent will select a random action instead of following the learned policy. Higher values of ϵ encourage exploration, while lower values encourage exploitation. The value of ϵ does not have to be constant throughout the learning process, and choosing an ϵ value that decays linearly from 1.0 to 0.1 provided good results (Mnih et al. 2015).

Performance Metrics

REinforced Signal COntrol (RESCO) is a testbed and benchmark environment to help judge the performance of RL-based algorithms for TSC. It comes built-in with the SUMO project and will simulate traffic in the same way it was simulated in other benchmark papers (Ault and Sharon 2021). This can be used to compare this project with what has already been done. The metrics that are relevant to this project include: total wait time for each vehicle, average delay per vehicle, average number of stops, average queue length, and average trip time (Reza et al. 2023). Studying how the agents respond to injected uncertainty such as an emergency vehicle or increased traffic demands is also an important metric for designing a robust system (Alegre, Bazzan, and da Silva 2020).

Expected Challenges

The non-stationary nature of multiagent RL problems is known to cause divergence problems due to the constantly changing dynamics. This can be somewhat mitigated by applying a scenario that is consistent across training, but the dynamics will still be changing due to the other agents' learning (Alegre, Bazzan, and da Silva 2020). Introducing context switches by changing the scenario, for example, making vehicles flow in waves to simulate rush hour is critical to designing a robust traffic system, but will be out of the scope of the initial research goals.

[1]Data was only collected in one scenario and may not reflect how the model performs in other scenarios.

[2]Data was extracted from graphs.

Timeline

Date	Milestone
2/19	4x4 grid environment setup and working
2/26	IDQN agents learning
3/4	Perform experiments and generate data
3/18	Injected uncertainty scenarios
4/1	Data compilation and writing

Table 2: Project Timeline

Conclusion

Overall, it was discussed how traffic signal control (TSC) can be formulated in the Markov Decision Process (MDP) paradigm. The state, action, and reward spaces were defined. Based on research done applying fixed algorithm and reinforcement learning to TSC, an Independent Deep Q-Learning Network approach was chosen for continued experiments, aiming to reduce the average cumulative delay D_t to 30.74 seconds or lower. The performance will be evaluated using the REinforced Signal Control RESCO toolkit, and simulations will be done on the Simulated Urban MObility SUMO environment. Given the partial and non-stationary nature of the TSC problem, it is expected to face divergence challenges. Overall, the project will be completed following the Project Timeline table 2.

References

Alegre, L. N. 2019. SUMO-RL. https://github.com/LucasAlegre/sumo-rl.

Alegre, L. N.; Bazzan, A. L. C.; and da Silva, B. C. 2020. Quantifying the Impact of Non-Stationarity in Reinforcement Learning-Based Traffic Signal Control. *CoRR*, abs/2004.04778.

Ault, J.; and Sharon, G. 2021. Reinforcement Learning Benchmarks for Traffic Signal Control. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; and Li, Z. 2020. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 3414–3421.

Choi, S.; Yeung, D.-Y.; and Zhang, N. 1999. An environment model for nonstationary reinforcement learning. *Advances in neural information processing systems*, 12.

Chu, T.; Wang, J.; Codecà, L.; and Li, Z. 2019. Multi-Agent Deep Reinforcement Learning for Large-scale Traffic Signal Control. *CoRR*, abs/1903.04527.

de Almeida, V. N.; Bazzan, A. L. C.; and Abdoos, M. 2022. Multiagent Reinforcement Learning for Traffic Signal Control: a k-Nearest Neighbors Based Approach. In *ATT@IJCAI*.

Ghanadbashi, S.; Safavifar, Z.; Taebi, F.; and Golpayegani, F. 2023. Handling uncertainty in self-adaptive systems: an ontology-based reinforcement learning model. *Journal of Reliable Intelligent Environments*.

Hafiz, A. M.; and Bhat, G. M. 2020. Deep Q-Network Based Multi-agent Reinforcement Learning with Binary Action Agents. arXiv:2008.04109.

Lee, K. M.; Ganapathi Subramanian, S.; and Crowley, M. 2022. Investigation of independent reinforcement learning algorithms in multi-agent environments. *Frontiers in Artificial Intelligence*, 5.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR*, abs/1312.5602.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Reza, S.; Ferreira, M. C.; Machado, J. J. M.; and Tavares, J. M. R. S. 2023. A citywide TD-learning based intelligent traffic signal control for autonomous vehicles: Performance evaluation using SUMO. *Expert Systems*. First published: 05 April 2023.

Terry, J.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L. S.; Dieffendahl, C.; Horsch, C.; Perez-Vicente, R.; et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 15032–15043.