Project Proposal

Building a Robust and Scalable RL-Based Traffic Control System

Tom Pecher
Department of Computer Science

USCM-AFM27: MComp(Hons)
Computer Science and Artificial Intelligence (Year 3)
University of Bath

November 7, 2024

Contents

1 Problem Description

1.1 The Evolution of Traffic Lights

Since the advent of vehicular transport, the challenge of efficiently coordinating the flow of vehicles has been a persistent one. As automated transport has advanced, so too has the imperative to adapt and manage this new superpower. Today, traffic control systems (TCSs) are omnipresent in our daily lives, as well as a frequent target of complaints. Yet strangely, few people know about the history of traffic systems, let alone how they operate in the present day.

1.1.1 Fixed-Control Time Systems

In the early 20th century, the first electronic TCSs were operated manually by humans[1], usually police officers. However, within a few years, these systems transitioned to automated versions. These early automated TCSs functioned on fixed time intervals, meaning that traffic signals would change according to predetermined durations[2]. While the deployment of such a large-scale automated TCS was a remarkable technological advancement, it introduced some notable limitations. In many ways, the system had become less intelligent; where there once was a human intelligence that could observe the traffic conditions and change their behaviour to account for any changes, this fixed-time TCS would be static and unchanging, resulting in huge congestion should the traffic conditions develop otherwise than anticipated.

1.1.2 Actuated Logic Systems

Naturally, the system architects sought to introduce some basic logic capability in TCSs. Strategically placed sensors, such as cameras and inductive loops embedded beneath the road surface, empowered the first actuated TCSs with the capability to gather environmental data and respond dynamically[3]. This advancement was particularly impactful at intersections, where vehicles were no longer required to idle at red lights on otherwise empty roads. Instead, an inductive loop buried under the road's surface at the junction's entrance would detect waiting vehicles, and if cameras confirmed no traffic within the junction, the signal would adjust to allow waiting traffic to proceed. This innovation markedly enhanced TCS efficiency, especially in localized instances like the one described. However, as road infrastructure became more dependable and the demand for personal vehicle ownership surged, an increasing load began to strain roadways, challenging the system's capacity to maintain efficiency.

1.1.3 Adaptive Traffic Systems

With the number of vehicles on roads increasing year-to-year, network-wide traffic trends began to fluctuate more violently than before: roads could be near empty one minute and hopelessly congested the next. Whilst actuated systems were adept at localized traffic coordination, they simple did not have the tools to deal with network-wide problems. Linking together multiple locations, adaptive TCSs where designed to address this problem[4]. These systems focused on synchronizing green signals (as so called "green waves") so that no vehicles with bad luck would be forced to wait at every junction along their journey.

This leaves us in the present day where adaptive systems are actively being implemented to improve overall network throughput. Nevertheless, these TCSs are still far from optimal traffic controllers as they struggle to adapt in the face of accidents and road closures. One may argue that since the days when we traded intelligence for convenience by replacing human-operated TCSs with automated ones, that "intelligence" has never truly returned as of yet. However, there may be a way to bring it back.

1.2 Reinforcement Learning

1.2.1 Reinforcement Learning Basics

In (Sutton and Barto, 2020)[5], reinforcement learning (RL) is defined as "a computational approach to goal-directed learning from interaction". Essentially, it is a framework for solving sequential decision problems in which an agent interacts with its environment. This interaction is typically described through:

- states: that the agent can be in
- actions: that the agent can choose to move between states
- rewards: the immediate benefits that an agent gains from moving to a state

If we intend to apply this framework to optimize a TCS, we need to make some adjustments. Traffic control is fundamentally a continuous-time problem and so if we intend to use RL in any useful capacity, we need to adapt the RL framework to work on continuous data. Therefore, rather than considering a set of states, actions and rewards, we should instead consider each as a continuous space. To handle this, RL algorithms typically use function approximators such as deep neural networks to generalize across the continuous state space. These approximators enable the agent to estimate values (like the value function or policy which it uses to determine what action to take) for any given state, even if it hasn't encountered that specific state before, by leveraging the structure learned from other similar states.

1.2.2 MARL

Multi-agent reinforcement learning (MARL) expands the traditional RL framework by allowing multiple agents to interact within the same environment, where each agent learns not only from its own actions but also from the actions and responses of other agents. This decentralized approach enables agents to develop cooperative, competitive, or mixed strategies to achieve individual or shared objectives, making MARL especially suitable for complex, distributed tasks like traffic control, resource management, and autonomous vehicle coordination.

1.3 The Future of Network Flow-Control

1.3.1 RL-Based Control Systems

Since its conception, RL has proven itself as an effective framework learning goal-driven sequential decision making, and so there has been much research aimed at porting this framework to solve the problem at hand[6]. Despite the RL framework being relatively straightforward, it is not so simple to decide how to formulate the RL setting in a traffic control environment (in other words, how you define the states and reward in such a way that is effective for learning). There is also the problem of credit assignment, that is understanding whether immediate rewards or losses are the result of recent actions or ones done long ago. Therefore, it is interesting to see the different approaches that papers in the field have taken to address specific limitations of this problem.

1.4 Limitations

Despite over a century's worth of development, it is clear we have still not reached a state of optimal traffic control.

Efficiency: While the efficiency of TCSs has greatly improved, there is some way to go before we can call our traffic control truly efficient. Even the latest traffic control algorithms, such as the adaptive traffic systems, fail to adapt to unforeseen circumstances such as accidents and road closures, which given the frequency of such events and the increase in road users makes this system unsustainable.

Furthermore, we must ask the question: what does it mean for our network to be efficient? In its simplest definition, we may consider it to be the throughput of the road network, however this does not consider the average waiting time or queue length, or more abstract factors such as average emissions and vehicle priority. An optimal TCS should consider all of these factors and know when to work to optimize one for another.

Robustness: A problem with earlier models such as the fixed-time TCS was the its static nature. As TCSs are growing increasingly complex and beginning to incorporate ML techniques, the result system will be little better than a black box making incomprehensible decisions. Such systems, whilst dynamic, could become too dynamic, resulting in unpredictable, erratic behaviour (to both the creators and the road users) which we may be unable to explain. Using AI in this way also brings up the question of responsibility: specifically, who is responsible for the system's decisions and who is to blame if it goes wrong. An optimal TCS should be dynamic but predictable, ideally with a decision-making process that is explainable.

Scalability: Installing a region-wide TCS is expensive. The infrastructure alone required to sustain it is already a large area of government spending and replacing an old system with a new one simple may not be viable. Alongside the financial cost of the new hardware, we must also consider the computational cost of implementing data-driven models, which are known for consuming vast amounts of resources. Additionally, TCSs that are used in or across different geographical regions will encounter different environmental and traffic conditions and so a runtime mismatch may occur where the TCSs encounter traffic patterns outside of what they were trained on, which could have possibly disastrous consequences. An optimal TCS should be lightweight but resilient and able to operate even in extreme (traffic) conditions.

2 Research Goals

2.1 Project Aim

To implement a RL-based traffic control system within a simulation environment and evaluate it against non-ML methods in environments outside of which it was trained.

2.2 Project Objectives

- Create a traffic simulation environment that realistically replicates real-life traffic flow phenomena.
- Implement the methods outlined in existing research and validate their results.
- Attempt to extend and improve upon existing research.

3 Requirements Specification

3.1 Functional Requirements

The following requirements lay out the specific features of the deliverable:

- Create a traffic simulation environment that realistically replicates real-life traffic flow phenomena (**High**).
- Create a non-ML based agent and assess its performance (**Medium**).
- Implement an RL-based system and assess its performance: compare to existing research and the simple model (**High**).
- Implement AV Integration and assess its performance: compare to no AV integration and how performance changes with percentage of vehicles being AVs (**Low**).

3.2 Nonfunctional Requirements

These requirements lay out what properties each feature should have and generally how the system should behave:

- Simulated environment should realistically replicate real-life traffic flow phenomena. (**High**)
- Non-ML system should attempt to mimic the current state of traffic light logic. (**High**)
- RL system should be robust: perform well in variable traffic conditions. (**High**)
- RL system should be scalable: perform well in road networks it was not trained on. (**Medium**)

- RL system should be based off of existing research. (**Medium**)
- System should improve upon existing research through combining ideas from different papers or from original ideas. (**Medium**)

3.3 Administrative Requirements

These requirements stipulate explicit milestones and the administration needed so that this project is submitted correctly and successfully:

- Project proposal should be completed by Friday 8th November 2024.
- Must include literature and technology survey, completed by Friday 6th December 2024.
- A basic RL-based system should be submitted as demonstration of progress by Monday 17th February 2025.
- Project should be completed by Friday 2nd May 2025.
- Project should follow the project plan unless unforeseen hindrances require a rethinking of the plan (which will be documented). Possible hindrances should be accounted for in the plan.
- Should keep a journal documenting the progress of the project.

Section 4 continues on the next page...

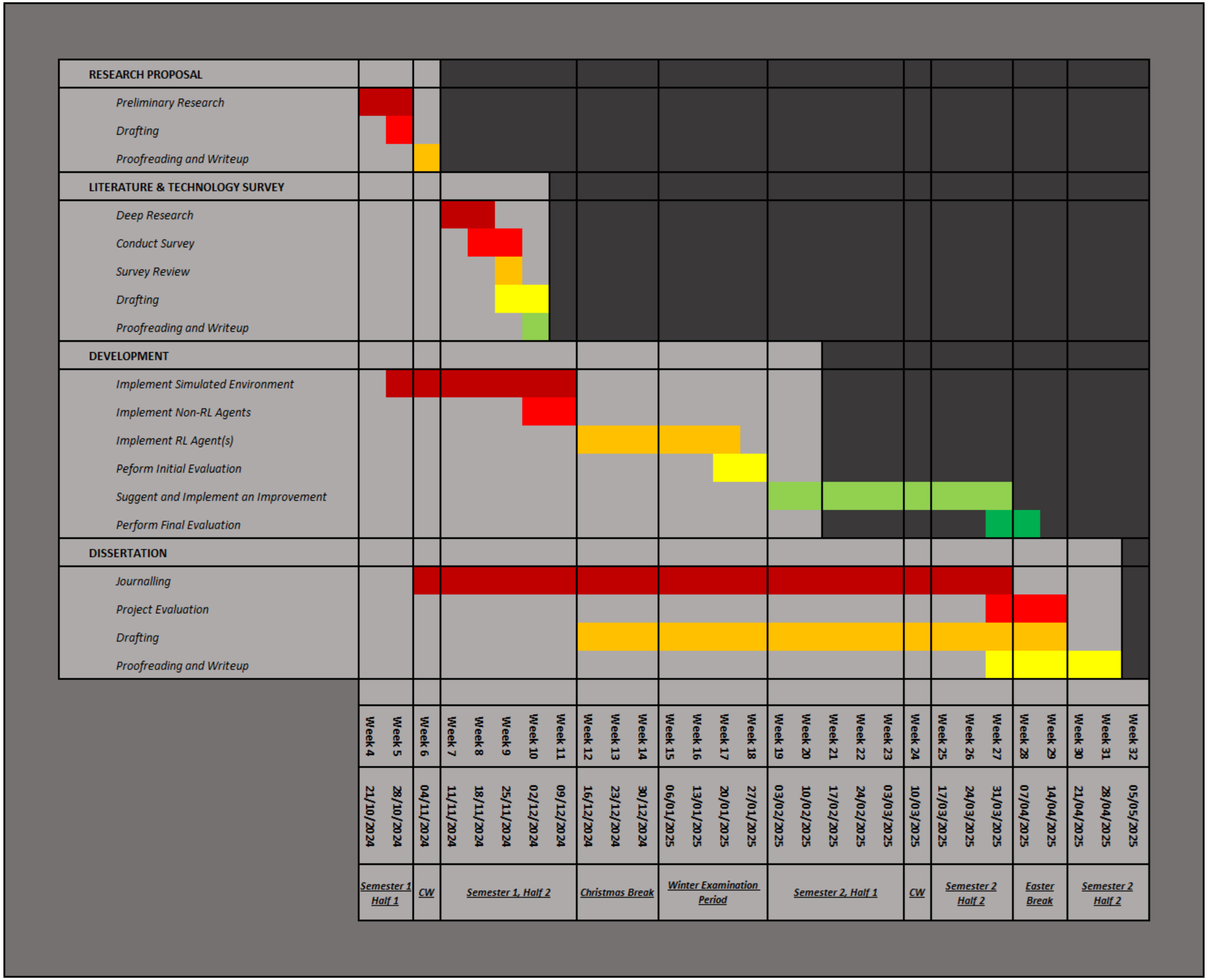

Figure 1: First iteration of project timeline as a Gantt chart

4 Project Plan

4.1 Project Timeline

APPROXIMATE TASK EFFORTS			
Task	Start Date	End Date	Duration
Research Proposal			
Preliminary Research	21/10/2024	03/11/2024	5 hours
Drafting	28/10/2024	05/11/2024	10 hours
Proofreading and Writeup	05/11/2024	08/11/2024	10 hours
Literature and Technology Survey			
Deep Research	11/11/2024	28/11/2024	20 hours
Conduct Survey	18/11/2024	01/12/2024	20 hours
Survey Review	25/11/2024	03/12/2024	5 hours
Drafting	27/11/2024	05/12/2024	10 hours
Proofreading and Writeup	18/11/2024	01/12/2024	15 hours
Development			
Implement Simulated Environment	28/10/2024	15/12/2024	50 hours
Implement Non-RL Agents	02/12/2024	18/12/2024	30 hours
Implement RL Agent(s)	16/12/2024	26/01/2025	80 hours
Perform Initial Evaluation	20/01/2025	26/01/2025	10 hours
Suggest and Implement Improvement	03/02/2025	01/04/2025	100 hours
Perform Final Evaluation	31/03/2025	13/04/2025	15 hours
Dissertation			
Journalling	04/11/2024	01/04/2025	20 hours
Project Evaluation	31/03/2025	20/04/2025	20 hours
Drafting	16/12/2024	25/04/2025	40 hours
Proofreading and Writeup	31/03/2025	30/04/2025	30 hours

4.2 Risk Assessment

To ensure that this project is successful, one must be proactive and responsive to any potential obstacles that may arise. Here is a short list of the main risks that I may encounter and how I would address each of them should they arise:

- *Unable to work (mitigating circumstance)*:
  A variety of unforeseen events may hinder my ability to work to my full capacity (such as illness, other work, etc.). I have accounted for this by leaving plenty of time for each task to be completed. The project

scope and timeline is also quite flexible (each bit builds off of the last), so a mitigating circumstance could slow my progress but never stop it.

- *Hardware failure*:
  This project is too important to not consider the risk of hardware failing and losing significant progress. As a result, I will commit and push any changes to my GitHub project repository as well as keeping a copy of the code on a separate USB.

- *Underestimation of project scope*:
  It is easy during the planning phase of any large project to greatly underestimate the difficulty of actually completing the project. As I have just mentioned, I have structured the project such that each bit of progress is incremental (as well as categorizing requirements based on priority), and so at any point in the project timeline, I will always have some results to show.

- *Unable to make progress*:
  If a certain step in the development process proves unusually difficult and I am unable to solve it with my current knowledge, I will ask for advice from my supervisor or the members of the RL lab. However, I will first revisit the research stage of the project to see if there is an alternative pathway towards completing the project.

5 Resources

5.1 Literary Resources

5.1.1 Learning RL

As has been recommended in the RL module, I have begun reading "Reinforcement Learning: An Introduction" (Sutton and Barto, 2020)[5]. I believe this will give me a deeper understanding of RL as a field (outside of research papers) that will be invaluable as I go on to suggest an extension to existing research.

5.1.2 Preliminary Research

I have begun collecting, reading and taking notes on a variety of research papers in the field. There are too many to list here (I will list them in the literature and technology survey), however here are some that I found most pertinent to this project:

- An Efficient Deep Reinforcement Learning Model for Urban Traffic Control (Lin, Y. et al., 2018)[7]
- Coordinated Deep Reinforcement Learners for Traffic Light Control (van der Pol, E. and Oliehoek, F. 2016)[8]
- Generalight: Improving environment generalization of Traffic Signal Control via meta reinforcement learning (Liu, C. et al., 2020)[9]
- Intelligent traffic light control (Wiering, M.A. et al., 2004)[10]
- Intellilight: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Wei, H. et al., 2018)[11]
- Reinforcement learning for traffic signal control: Comparison with commercial systems (Cabrejas-Egea, A., Zhang, R. and Walton, N., 2021)[12]

5.2 Computational Resources

5.2.1 Computer

I will use my computer to develop the system and the agents within it, as well as to conduct research, make notes and finally write up any deliverables.

5.2.2 GPU Cluster

In order to train my system across as many situations as possible (to ensure that it is robust), my computer may not be powerful enough to do the computation in a practical amount of time. Therefore, I may need to make use of the university's GPU cluster.

6 References

[1] Lise (2022) 1868-2019: A brief history of traffic lights, Inclusive City Maker. Available at: https://www.inclusivecitymaker.com/1868-2019-a-brief-history-of-traffic-lights/ (Accessed: 07 November 2024).

[2] Papageorgiou, M. et al. (2004) Review of Road Traffic Control Strategies, Review of road traffic control strategies. Available at: https://www.researchgate.net/publication/2986201_Review_of_road_traffic_control_strategies (Accessed: 07 November 2024).

[3] Choi, B. (2017) Traffic-actuated signals - USC Viterbi School of Engineering, USC Viterbi School of Engineering - USC Viterbi School of Engineering. Available at: https://illumin.usc.edu/traffic-actuated-signals/ (Accessed: 07 November 2024).

[4] Sims, A.G. and Dobinson, K.W. (1980) The Sydney Coordinated Adaptive Traffic (SCAT) system philosophy and benefits - IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/1622746/ (Accessed: 07 November 2024).

[5] Sutton, R.S. and Barto, A.G. (2020) Reinforcement learning: An introduction. Cambridge, MA: The MIT Press. Available at: http://incompleteideas.net/book/the-book.html (Accessed: 07 November 2024).

[6] Wei, H. et al. (2020) A survey on Traffic Signal Control Methods, arXiv.org. Available at: https://arxiv.org/abs/1904.08117 (Accessed: 07 November 2024).

[7] Lin, Y. et al. (2018) An efficient deep reinforcement learning model for Urban Traffic Control, arXiv.org. Available at: https://arxiv.org/abs/1808.01876 (Accessed: 07 November 2024).

[8] Pol, E. van der and Oliehoek, F. (2016) Publications, Frans Oliehoek: Coordinated Deep Reinforcement Learners for Traffic Light Control. Available at: https://www.fransoliehoek.net/publications/htmlfiles/b2hd-VanDerPol16LICMAS.html (Accessed: 07 November 2024).

[9] Liu, C. et al. (2020) Generalight: Improving environment generalization of Traffic Signal Control via meta reinforcement learning, arXiv.org. Available at: https://arxiv.org/abs/2009.08052 (Accessed: 07 November 2024).

[10] Wiering, M.A. et al. (2004) Intelligent traffic light control. Available at: https://www.researchgate.net/publication/2942266_Intelligent_Traffic_Light_Control (Accessed: 07 November 2024).

[11] Wei, H. et al. (2018) Intellilight: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM Other conferences. Available at: https://dl.acm.org/doi/10.1145/3219819.3220096 (Accessed: 07 November 2024).

[12] Cabrejas-Egea, A., Zhang, R. and Walton, N. (2021) Reinforcement learning for traffic signal control: Comparison with commercial systems, arXiv.org. Available at: https://arxiv.org/abs/2104.10455 (Accessed: 07 November 2024).