Deep reinforcement learning for traffic light control optimization in multi-modal simulation of SUMO

Yimin Xu

Deep reinforcement learning for traffic light control optimization in multi-modal simulation of SUMO

TIL5060

Thesis report

by

Yimin Xu

Thesis committee:

Chair:	Prof. Bart De Schutter
Supervisors:	Asst. Prof. Marco Rinaldi
	Postdoc. Dingshan Sun
Independent examiner:	Asst. Prof. Azita Dabiri
Faculty:	Faculty of Civil Engineering and Geosciences
Project Duration:	January, 2024 - September, 2024
Student number:	5696925

Faculty of Civil Engineering and Geosciences · Delft University of Technology

Preface

Two-year master life is coming to an end, and I'm glad that I finished my master thesis at this wonderful moment. Looking back on the past two years of study at TU Delft, I have gained a lot. TU Delft has greatly improved my academic level with its cutting-edge scientific theories, high-level teaching and rigorous academic attitude. This is exactly what I was pursuing before coming here.

Personally, I particularly like theories and research related to traffic control. Through the rational use and optimization of traffic control, the operation efficiency of the traffic network can be maintained at a high level, which indirectly has a profound impact on many aspects of society. That's why I choose *Deep reinforcement learning for traffic light control in multi-modal simulation of SUMO* as my master thesis topic. And I hope my research can have some positive effect on the optimization of traffic control and trigger some inspiration for future relevant research.

Finally, I want to express my gratitude to some special people who are important to me. First, I would like to thank my thesis supervisors Prof. Bart De Schutter, Asst. Prof. Marco Rinaldi, and Postdoc. Dingshan Sun for sparing valuable time to help me with my thesis. It is their patient instructions that help me finish this thesis step by step. Second, I would like to thank my classmates and friends. It is their company that adds a lot of fun and color to my life. Third, I would like to especially thank my parents. Without their emotional and financial support, I would not have achieved what I have today.

Yimin Xu
Delft, September 2024

Summary

This thesis investigates the application of different deep reinforcement learning methods for optimizing traffic light control in multi-modal urban traffic environments using the SUMO simulator.

Urban traffic congestion, with its significant economic, environmental, and social impacts, necessitates more sophisticated control strategies that can adapt to varying traffic conditions. Traditional traffic control systems, like fixed-time and adaptive methods, are often insufficient in handling the complexity of multi-modal traffic, which includes various traffic modes such as passenger cars and buses. Deep reinforcement learning, with its ability to dynamically optimize traffic light control without requiring prior knowledge of traffic patterns, is a promising method to improve traffic efficiency and achieve transit priority in multi-modal traffic.

A comprehensive literature review of the existing applications of deep reinforcement learning for traffic control in traffic simulation of SUMO at three different levels (intersection, vehicle, and combined level) and other traffic control strategies in this field is conducted, which provides an overview of various traffic control methods used in the traffic simulation of SUMO, discusses their performance, and identifies research limitations in this area. The research aims to address these limitations by employing deep reinforcement learning algorithms, particularly multi-agent deep reinforcement learning methods, to coordinate multiple traffic lights within the simulation environment.

Research experiments are conducted in three different cases, which are set in road networks of different sizes respectively. And the details of the case study including the simulation setups, the implementation of different deep reinforcement learning methods, the comparison methods (fixed, max-pressure), the computation setups as well as the experimental evaluation for these methods are fully described.

All the applied deep reinforcement methods are evaluated in terms of training process, model evaluation, and training time. And the research results demonstrate that deep reinforcement learning methods, especially multi-agent deep reinforcement learning methods, can significantly enhance traffic flow efficiency and achieve transit priority in complex urban settings in multi-modal simulation.

Finally, the conclusions, innovations, limitations of the research, and future research directions are fully discussed. This research paves the way for developing efficient deep reinforcement learning agents that consider the needs of different road users and prioritize public transport for the application of deep reinforcement learning in multi-modal simulation, which is of great significance for achieving more optimized traffic light control that considers multiple factors and realizing a more efficient and sustainable urban transportation system.

Contents

List of Figures

List of Tables

1

Introduction

1.1. Research context

Traffic congestion poses a significant challenge in many urban areas, detrimentally impacting the quality of life and environmental sustainability. Increased travel times, fuel consumption, and emissions associated with congestion contribute to economic losses, air pollution, and greenhouse gas emissions. Additionally, congestion-induced stress and reduced accessibility to essential services disproportionately affect vulnerable populations, highlighting the urgent need for effective solutions to optimize multi-modal traffic flow and mitigate the adverse consequences of traffic congestion.

Multi-modal traffic refers to the traffic flow of two or more traffic modes within a road network, which can include different major transport modes like passenger vehicles, buses, trams, trains, bicycles, pedestrians, and so on.

Achieving transit priority in multi-modal traffic is of great importance as this can help to reduce congestion and improve the efficiency of public transportation (He et al., 2014). The research conducted by B. Y. Zhang et al. indicates that prioritizing public transportation is crucial for effectively addressing the increasing problem of city traffic congestion (B. Y. Zhang et al., 2014).

Traffic control is one of the key strategies to reduce congestion, improve traffic flow, and achieve transit priority. However, it is also a complex and dynamic task that requires careful planning and coordination among different modes of transportation, such as passenger cars, and public transport. Traditional control methods like fixed traffic light control rely on handcrafted rules or fixed schedules that may not adapt well to changing traffic conditions or user preferences.

1.1.1. Adaptive traffic light control

Adaptive traffic light control systems were invented to address the limitations of fixed-time traffic signals, which do not account for real-time traffic conditions. Adaptive traffic control systems can dynamically adjust traffic light duration based on real-time traffic data gathered by sensors like inductive loops, cameras, or radar, which helps to reduce traffic congestion and enhance overall traffic throughput (Shinde, 2017). Adaptive traffic controls such as max-pressure (Varaiya, 2013) have been widely used to optimize traffic flow and improve traffic efficiency. However, adaptive traffic light control systems are not effective in handling highly dynamic and complex traffic patterns (Shinde, 2017).

1.1.2. Responsive traffic light control

Responsive traffic light control systems were developed to provide a more rapid response to current traffic conditions at intersections. Unlike fixed-time signals, responsive traffic light control systems use real-time data from sensors or cameras to adjust signal timings based on the actual traffic situation, which allows for more efficient traffic management, reducing total traffic delay (Fouladvand et al., 2004). Responsive traffic light control such as actuated traffic light control (Darroch et al., 1964) has been widely applied in urban areas. However, responsive traffic light control systems may struggle with scalability and adaptability to rapidly changing traffic conditions (Fouladvand et al., 2004).

1.1.3. Reinforcement learning

Currently, reinforcement learning (RL), especially deep reinforcement learning (DRL), has also attracted attention as a promising technique for helping optimize traffic control (Shi et al., 2023; Huang et al., 2023; S. Wang & Wang, 2023), as it can potentially overcome the limitations of traditional methods such as the need for prior knowledge, and adapt to changing traffic patterns in real-time, making them suitable for dynamic and unpredictable environments like urban traffic networks.

Reinforcement learning is a type of machine learning where agents learn from their actions and states within an environment to maximize rewards. Introduced from a computer science viewpoint by Kaelbling et al. in 1996, it involves agents developing behaviors through trial-and-error interactions in a dynamic environment (Kaelbling et al., 1996). Unlike supervised learning, which relies on labeled data, reinforcement learning lacks explicit supervision. Instead, agents learn from the outcomes of their actions, improving their strategies through trial and error. The core concept in reinforcement learning is the Markov Decision Process (MDP), which defines the environment in terms of states, actions, rewards, and state transitions. The goal for an agent is to discover an optimal policy, mapping states to actions to maximize the expected cumulative reward over time. A key challenge in reinforcement learning is balancing exploration (trying new actions to discover their effects) and exploitation (selecting actions known to yield high rewards).

The foundation of reinforcement learning is built on the seminal works of Sutton and Barto, they introduced key algorithms like Q-learning and temporal difference (TD) learning, which have been crucial in the development of reinforcement learning methodologies (Sutton & Barto, 1999).

Based on the methods used (approximating the underlying model, optimal policy, or optimal value function) in reinforcement learning, it can be divided into the following three categories (Feng & Zhong, 2023):

- **Model-based RL:** In model-based reinforcement learning, the main goal of the learner is to estimate the underlying model of the environment and then improve the policy based on this estimated model (Feng & Zhong, 2023). Algorithms like AlphaZero (Silver et al., 2017), World Models (Ha & Schmidhuber, 2018), I2A (Weber et al., 2017), MBMF (Nagabandi et al., 2018), and MBVE (Feinberg et al., 2018) are commonly used in this category.
- **Value-based RL:** In value-based reinforcement learning, attention turns towards approximating the value function, and policy updates are guided by the estimated value function (Feng & Zhong, 2023). Algorithms like Q-Learning (C. J. Watkins & Dayan, 1992), Deep Q-Network (Mnih et al., 2015), Double DQN (van Hasselt et al., 2016), Dueling DQN (Z. Wang et al., 2015), Rainbow DQN (Hessel et al., 2018), LINVIT (S. Zhang et al., 2024) are commonly used in this category.
- **Policy-based RL:** Policy-based reinforcement learning enhances the performance of the agent through direct policy updates (Feng & Zhong, 2023). Algorithms like policy gradient (Sutton et al., 1999), natural policy gradient (Kakade, 2001), Asynchronous Advantage Actor Critic (A3C) (Mnih et al., 2016), Advantage Actor Critic (A2C) (Dhariwal et al., 2017), and proximal policy optimization (PPO) (Schulman et al., 2017) are commonly used in this category.

1.1.4. Deep reinforcement learning

Deep reinforcement learning (DRL) extends traditional reinforcement learning by using deep neural networks to approximate decision-making functions like the value function, policy, or environmental model. This advancement has allowed agents to function in high-dimensional, complex environments where traditional methods fall short. A notable breakthrough in DRL is the development of Deep Q-Networks (DQN) by Mnih et al., which uses deep learning to approximate the Q-value function. This innovation enabled agents to learn to play Atari games directly from raw pixel inputs, showcasing the potential of combining deep learning with reinforcement learning (Mnih et al., 2015).

Incorporating deep learning into reinforcement learning enables the management of large state spaces and the generalization of policies across different states. However, it also brings challenges like stability and convergence issues, prompting the creation of enhancements such as experience replay and target networks, both introduced in the DQN paper by Mnih et al. Since then, deep reinforcement learning has been utilized in various fields, including robotics, gaming, and autonomous driving, expanding the capabilities of intelligent agents.

According to the number of agents involved in the learning process, deep reinforcement learning can be divided into single-agent deep reinforcement learning (SADRL) and multi-agent deep reinforcement

learning (MADRL). SADRL focuses on optimizing the decisions of one agent, while MADRL deals with the more complex problem of coordinating multiple agents (Hao et al., 2024).

In single-agent deep reinforcement learning, a single agent interacts with an environment to achieve a goal. The agent learns from the outcomes of its actions, receiving rewards or penalties. The goal is for the agent to develop a policy, which maps states to actions, to maximize cumulative rewards. This involves exploration, where the agent tests different actions to gather information, and exploitation, where it makes the best decisions based on its current knowledge (Hao et al., 2024).

In multi-agent deep reinforcement learning, several agents interact with one another and their environment to achieve goals, either cooperatively or competitively. Each agent acts independently with limited knowledge of the environment and relies on feedback from nearby agents. These agents can learn autonomously, without external critics or teachers, enabling them to solve complex problems (Hao et al., 2023). A major challenge in multi-agent deep reinforcement learning is coordinating multiple agents, particularly when their interests conflict. Another issue is the non-stationary environment, where one agent's behavior impacts the experiences of others. Despite these difficulties, multi-agent deep reinforcement learning has been successfully applied in various fields, such as game AI, autonomous vehicles, and robotics (Hao et al., 2024).

So far, deep reinforcement learning has successfully tackled numerous complex decision-making tasks that were once beyond the capabilities of machines. It has been applied to various fields, including games, robotics, natural language processing, and self-driving cars, and has demonstrated promising results in traffic control optimization in simulation studies, as discussed in Chapter 2.

1.1.5. SUMO

SUMO is an open-source microscopic traffic simulator that supports the modeling of complex, multi-modal traffic systems, including vehicles, public transport, and pedestrians. It accounts for various factors influencing traffic flow, such as road layout, vehicle behavior, and user interactions. The simulator includes several companion tools to simplify key processes like network creation, simulation execution, and performance evaluation. These tools handle tasks such as network importing, route planning, visualization, and emission analysis. SUMO also supports custom model integration and provides multiple APIs for external simulation control. A notable feature is the Traffic Control Interface (TraCI), which enables real-time manipulation of an active traffic simulation (Lopez et al., 2018). Due to these features, SUMO is employed as the simulation tool in this research.

1.2. Research Problem

Even though there are already many studies on deep reinforcement learning for traffic control optimization in traffic simulation of SUMO, there are still some limitations in the existing relevant studies, including the limited consideration of total travel time as the optimization indicator for the application of DRL in multi-modal simulation of SUMO, unexplored potential of single agent to control multiple traffic lights, unexplored performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes, and limitated evaluation of DRL-based traffic light control policies as stated in Section 2.4.

1.3. Research Objectives

This research aims to address the mentioned limitations by applying different deep reinforcement learning methods for traffic light control optimization to improve traffic efficiency and achieve transit priority in multi-modal traffic scenarios by using SUMO simulator. Based on the main objectives, the specific objectives can be described as follows:

- To set up multi-modal simulations in SUMO, including passenger cars, buses, and intersections with traffic lights.
- To implement different deep reinforcement learning algorithms for traffic light control optimization that can handle the complexity of multi-modal traffic scenarios.
- To train different deep reinforcement learning algorithms, especially MADRL, for developing good traffic light control policies.

- To conduct a performance evaluation of different DRL-based traffic light control compared with other strategy-based traffic light control.

1.4. Research Scope

The scope of this research includes the following aspects:

- The research area is set in the urban area, as urban areas typically have multi-modal traffic and more complex traffic dynamics, which makes traffic control a challenging problem and an interesting domain for applying deep reinforcement learning.
- The traffic control in this research focuses on traffic light control, as it is the main traffic control means in the urban area.
- The traffic light control in the research is formulated as a discrete-time process, as it allows for more manageable and computationally efficient modeling of the complex traffic dynamics.
- The multi-modal traffic simulation is set up by using SUMO, as SUMO is able to model different traffic modes and achieve good interactions between vehicles and traffic lights.
- DRL algorithms, especially MADRL which can handle the situation of multiple traffic lights and facilitate collaboration between traffic lights, are applied to optimize traffic light control.

1.5. Research questions

The main objective of this research is to optimize and evaluate DRL-based traffic light control for multi-modal traffic scenarios using SUMO simulator, which is achieved by answering the following main research question:

How can deep reinforcement learning algorithms be effectively applied to optimize traffic light control in multi-modal simulation of SUMO?

And the main research question is jointly answered by the following sub-questions:

1. *How to achieve transit priority in multi-modal simulation?*
2. *How to effectively represent the state space of a traffic system in multi-modal simulation of SUMO for deep reinforcement learning?*
3. *What actions are applied to improve traffic flow and reduce congestion in multi-modal simulation of SUMO?*
4. *What reward functions can effectively incentivize traffic control algorithms to improve traffic flow and reduce congestion in multi-modal simulation of SUMO?*
5. *What deep reinforcement learning algorithms can be used for traffic light control optimization in multi-modal simulation of SUMO?*
6. *How to define performance metrics and evaluate the performance of deep reinforcement learning algorithms for traffic light control optimization in multi-modal simulation of SUMO?*

1.6. Research Structure

The structure of the following thesis is arranged as follows. Chapter 2 explores the existing literature concerning deep reinforcement learning for traffic control optimization in traffic simulation of SUMO and identifies some research limitations in the relevant research so far. Chapter 3 describes the principle of the main research methods used for multi-modal simulation. Chapter 4 describes the case study of the research including the simulation setups, the implementation of different DRL algorithms, the comparison methods, the computation setups, the experimental evaluation for these methods, as well as the experimental results with analysis and discussions on different DRL algorithms in different cases. Chapter 5 summarizes the main findings, contributions, limitations, and implications of the thesis, and propose some recommendations for future research.

2

Literature review

This chapter provides a review of the application of deep reinforcement learning for traffic control optimization in traffic simulation of SUMO at three different levels (intersection level, vehicle level, and combined level) and other traffic management strategies in this field. The intersection level focuses on optimizing traffic light control to improve traffic flow and reduce congestion. The vehicle level focuses on optimizing the behavior of individual vehicles to improve traffic efficiency and safety. The combined level focuses on coordinating controls of different levels to achieve optimal traffic performance across an entire transportation network.

The literature review presented in this chapter discusses the application of DRL for traffic control optimization at each of these levels, which are shown in Table 2.1. The various DRL algorithms that have been used, the performance of these algorithms, and the challenges and opportunities for future research are discussed.

Table 2.1: Literature overview regarding deep reinforcement learning for traffic control optimization in SUMO

Literature	DRL Type	Control Method	Control Level
Genders, Razavi (2016)	DQN	traffic light	intersection
Liang et al. (2018)	3DQN	traffic light	intersection
Chen et al. (2019)	DQN	traffic light	intersection
Guo et al. (2019)	DQN	traffic light	intersection
Koh et al. (2020)	DQN	vehicle routing and navigation	vehicle
C. Li et al. (2020)	FSPPO	traffic light	intersection
D. Li et al. (2020)	DQN	traffic light	intersection
M. Li et al. (2020)	DDPG	driving stragegies	vehicle
Makantasis et al. (2020)	DDQN	driving policy	vehicle
Szőke et al. (2020)	Vanilla Policy Gradient	route navgiation	vehicle

Continued on next page

Table 2.1: Literature overview regarding deep reinforcement learning for traffic control optimization in SUMO (continued)

Literature	DRL Type	Control Method	Control Level
Zhao et al. (2020)	DDQN	decision making and interactions between vehicles	vehicle
Bouktif et al. (2021)	P-DQN	traffic light	intersection
Z. Li, Xu, & Zhang (2021)	MADDPG	traffic light	intersection
Z. Li, Yu, et al. (2021)	KS-DDPG	traffic light	intersection
Ma et al. (2021)	PPO	traffic light	intersection
Bagwe et al. (2022)	PPO	automatic merging maneuvers	vehicle
Codeca & Cahill (2022)	MAPPO	route planning	vehicle
Han et al. (2022)	3DQN	traffic light	intersection
Kumar et al. (2022)	DQN	traffic light	intersection
Louw et al. (2022)	PPO, DQN	traffic light	intersection
J. Li et al. (2022)	Distributional DQN	traffic light	intersection
B. Wang et al. (2022)	EP-D3QN	traffic light	intersection
C. Wang et al. (2022)	DDPG, TD3	ramp metering and variable speed limit	intersection and road
Mei et al. (2023)	PPO, DQNs	traffic light	intersection
Shabab et al. (2023)	DQN	traffic light	intersection
Shen et al. (2023)	DQN	traffic light	intersection
Yu et al. (2023)	DQN	traffic light	intersection
Zhong et al. (2023)	DQN	traffic light	intersection

2.1. Deep reinforcement learning for traffic control optimization in single-modal simulation of SUMO

Currently, the majority of the relevant research about deep reinforcement learning for traffic control optimization using SUMO is conducted in single-modal simulation, and the utilized traffic control varies a lot including at intersection, vehicle, and combined level for different optimization objectives.

2.1.1. Deep reinforcement learning for traffic control optimization at intersection level in single-modal simulation of SUMO

In terms of the studies about deep reinforcement learning for traffic control optimization at intersection level in single-modal simulation of SUMO, DRL of different categories are applied.

Many related studies use value-based DRL, Deep Q-Network (DQN) or its variants, as the main research methods to optimize traffic light control in single-modal simulation. Genders and Razavi employed DQN to construct an adaptive traffic light control agent in SUMO. This system effectively reduced average cumulative delay, average queue length, and average travel time (Genders & Razavi, 2016). Chen et al. introduced a new adaptive signal control method utilizing DQN to coordinate traffic light controls on arterial roads. Through traffic data detected in real time, the hybrid global and local reward functions were optimized, showcasing the effectiveness and efficiency of the method over actual and fix-time traffic light control methods in SUMO simulations (Chen et al., 2019). Guo et al. utilized a DQN DRL approach for traffic light control optimization at urban intersections. Their simulation results demonstrated that the approach converged well and generalized effectively, showing notable improvements in wait time and queue length compared to some benchmarking traffic light control approaches (Guo et al., 2019). Bouktif et al. adopted a Parameterized Deep Q-Networks (P-DQN) DRL, considering both continuous and discrete decisions for traffic light control optimization. Their SUMO simulation results illustrated that the presented framework using P-DQN markedly reduced average queue length and travel time compared to the alternative traffic light control systems based on deep reinforcement learning (Bouktif et al., 2021). B. Wang et al. introduced EP-D3QN, a DRL method for optimizing traffic light timing that utilizes double dueling deep Q-network (3DQN), self-organizing traffic light control, and max pressure traffic light control. The simulation results using SUMO demonstrated that EP-D3QN is better than the other four methods in the scenarios of heavy and light traffic flow respectively, reducing the travel time and waiting time of vehicles (B. Wang et al., 2022). Yu et al. introduced an innovative approach to enhance the service level of both the bus system and car traffic within a multi-modal road network. By integrating bus priority and holding strategies with traffic light control using decentralized DQN controllers, the proposed approach surpasses the performance of model-based adaptive traffic light control approaches and the centralized reinforcement learning approach in terms of bus stability and traffic efficiency (Yu et al., 2023). Shabab et al. applied DQN for traffic light control optimization in the road network of multiple intersections. Simulations conducted in SUMO demonstrate that the suggested deep reinforcement learning model, by optimizing traffic signal timing across multiple intersections, greatly decreases both waiting time and traffic conflict when compared with the benchmark and is beneficial for both safety and mobility (Shabab et al., 2023). J. Li et al. presented an enhanced Distributional DQN to develop a traffic light control optimization decision-making model , which effectively leverages intersection environment information for each phase action to predict the distribution of future total returns. Their experiment results show that the Distributional DQN achieves a quicker rate of convergence compared to DQN and has a much lower cumulative intersection delay and higher mean driving velocity (J. Li et al., 2022). Liang et al. used Dueling Double Deep Q-Network (3DQN) for traffic light control optimization by using the states of small grids discretized from the intersection and the reward calculated by the difference in cumulative waiting time between two consecutive time steps. The simulation results using SUMO demonstrate the effectiveness of the model in managing traffic lights (Liang et al., 2018). D. Li et al. developed an adaptive traffic light control model in SUMO using the Deep Q-Network algorithm. Real-time traffic data, including the number of vehicles and mean speed at one or more intersections, are utilized as the states of the model. To minimize the mean waiting time, an optimal traffic signal phase and duration are determined by the agents for both single-intersection and multi-intersection scenarios. Testing the model with datasets from three different traffic situations shows that it outperforms three other methods including Q-learning, Webster fixed timing control, and longest queue first method in terms of the mean travel time and waiting time (D. Li et al., 2020).

There are also some studies applying policy-based DRL for traffic light control optimization in single-modal simulation of SUMO. Z. Li, Xu, and Zhang introduced a multi-agent deep deterministic policy gradient (MADDPG) method with centralized learning and decentralized execution, which builds upon the actor-critic policy gradient algorithms. The simulation results of the model demonstrate that this method can efficiently manage traffic lights (Z. Li, Xu, & Zhang, 2021). C. Li et al. presented the Fairness Scheduling Proximal Policy Optimization (FSPPO), a DRL algorithm that integrates the Proximal Policy Optimization (PPO) algorithm with a fairness criterion. The algorithm aimed to reduce the longest waiting time for drivers during a traffic light cycle, and the results indicated its efficient optimization for the fairness criterion (C. Li et al.,

2020). Ma et al. introduced a traffic light timing optimization strategy utilizing Proximal Policy Optimization (PPO), which enables traffic lights to select proper phases based on the traffic conditions for each direction of the corresponding intersection. and to dynamically adjust the duration of these phases. Experiments conducted on actual traffic data using SUMO demonstrate that this approach significantly reduces the queue length and vehicle waiting times under different traffic scenarios compared to conventional traffic light control methods (Ma et al., 2021). Kumar et al. suggested a traffic-light scheduling scheme using SDDRL, incorporating the dynamics of vehicles from real-time traffic environments. Their SUMO simulation results demonstrate that the suggested method has improved multiple performance indicators including throughput, mean speed, mean waiting time, and mean queue length compared to several state-of-the-art methods including DQN, NFM, FLTC, FTA, and MPA (Kumar et al., 2022). Z. Li, Yu, et al. introduced Knowledge Sharing Deep Deterministic Policy Gradient (KS-DDPG), a multi-agent DRL approach for optimizing traffic light control through enhanced agent cooperation. By enabling knowledge sharing, every agent is able to access the combined traffic environment data gathered by all agents. Experiments with synthetic and real-world data show that KS-DDPG outperforms traditional traffic light control methods and state-of-the-art RL-based methods in efficiently managing large-scale road networks and handling traffic flow fluctuations (Z. Li, Yu, et al., 2021).

Morever, some studies utilize both value-based and policy-based DRL for traffic light control optimization in single-modal simulation. Louw et al. applied two DRL algorithms PPO and DQN to improve urban traffic management on a simulated intersection in SUMO. The experiment results show that both methods showcase significant enhancements over conventional traffic light control methods (Louw et al., 2022). Mei et al. applied two DRL algorithms PPO and DQNs to establish a model in SUMO simulation in an area with two signalized intersections and two railroad crossings. The results of SUMO simulation underscore the superior performance of the presented DRL-based traffic light control method over the fixed traffic light control (Mei et al., 2023).

In conclusion, this review underscores the diverse and effective applications of deep reinforcement learning for optimizing traffic control at the intersection level within single-modal SUMO simulations. The studies discussed demonstrate that value-based DRL methods, particularly Deep Q-Networks (DQN) and its variations, have consistently improved traffic efficiency by reducing delays, queue lengths, and travel times across various traffic scenarios. Additionally, policy-based DRL approaches, such as those utilizing Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), have proven effective in more complex, multi-agent environments, further enhancing traffic signal control. Moreover, hybrid approaches combining both value-based and policy-based methods have shown significant promise, underscoring the adaptability and robustness of DRL in managing urban traffic at intersections. These findings collectively suggest that DRL offers a powerful and flexible solution for optimizing traffic signal control, paving the way for more intelligent and responsive urban traffic management systems.

2.1.2. Deep reinforcement learning for traffic control optimization at vehicle level in single-modal simulation of SUMO

In terms of the studies about deep reinforcement learning for traffic control optimization at vehicle level in single-modal simulation of SUMO, value-based or policy-based DRL is also applied in the existing relevant research.

Many related studies use value-based DRL, Deep Q-Network (DQN) or its variants, as the main research methods to optimize traffic control at vehicle level. Koh et al. introduced an improved DQN approach for constructing a smart vehicle routing and navigation system that operates in real time. The experimental results show that the introduced approach showcased superior performance over the benchmark algorithms for routing optimization (Koh et al., 2020). Makantasis et al. used Double Deep Q-Network (DDQN) DRL for developing a driving policy under the traffic situations both involving manual and autonomous driving vehicles on the road. The simulation results of SUMO show that the developed DRL-based driving policy has better performance over SUMO policies, under the scenarios with and without the introduction of uncertainties (Makantasis et al., 2020). In a different study, Zhao et al. utilized the DDQN DRL to model the decision-making process and interactions among vehicles during highway driving. The agent vehicle, as per the simulation results, demonstrated the capability to accomplish the highway driving task easily, approximating the maximum safe driving velocity that avoids collisions (Zhao et al., 2020).

There are also some studies applying policy-based DRL for traffic control optimization at vehicle level

in single-modal simulation of SUMO. Bagwe et al. introduced an innovative method for ensuring the robust on-ramp merge of Connected and Autonomous Vehicles through Augmented and Multi-modal DRL considering comfortable driving behavior, driving safety, and traffic efficiency. To enhance the reliability of the merging operations, surveillance images and basic safety messages are both used at the same time for multi-modal observation to train a policy model using PPO with augmented data. The simulation results of SUMO demonstrate the effectiveness and efficiency of their robust on-ramp merging design in two typical merging scenarios (Bagwe et al., 2023). In another study, M. Li et al. used Deep Deterministic Policy Gradient (DDPG) to develop a driving strategy for individual vehicles with the purpose of reducing oscillations and enhancing traffic safety in stop-and-go waves. Their SUMO simulation results demonstrate that the developed strategy outperforms the adaptive cruise control and jam-absorbing driving strategies, showcasing its effectiveness in lowering the risk of accidents (M. Li et al., 2020). Szőke et al. employed the Vanilla Policy Gradient method to develop an agent for navigation and driving behavior optimization. The experimental results reveal that the developed agent can safely navigate through varying highway traffic conditions and successfully traverse the specified section while maintaining the reference velocity in a preset highway situation (Szőke et al., 2020).

In summary, the application of deep reinforcement learning (DRL) for traffic control optimization at the vehicle level within single-modal SUMO simulations has yielded promising results. Both value-based and policy-based DRL approaches have been effectively employed to enhance vehicle routing, driving policies, and decision-making processes. Studies utilizing Deep Q-Networks (DQN) and its variants have demonstrated significant improvements in vehicle navigation, route optimization, and highway driving scenarios, showcasing the ability of these methods to adapt to dynamic traffic conditions and avoid congestion. Additionally, policy-based DRL methods, such as Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), have proven effective in addressing challenges like safe merging of autonomous vehicles and mitigating traffic oscillations. Collectively, these findings highlight the potential of DRL to optimize vehicle-level traffic control, enhancing both safety and efficiency in various driving environments.

2.1.3. Deep reinforcement learning for traffic control optimization at combined level in single-modal simulation of SUMO

In terms of the studies about deep reinforcement learning for traffic control optimization at combined level, much fewer studies are conducted and the existing study of C. Wang et al. uses policy-based DRL methods to learn optimal policies.

C. Wang et al. presented a centralized traffic control system featuring a unique double-layer structure designed to synchronize a plurality of variable speed limit and ramp metering traffic controllers. The policy-based DRL approaches are incorporated in the system to learn coordinated actions within a high-dimensional traffic environment on the freeway. The simulation results indicate the superiority of policy-based methods over alternative approaches including TD3, DDPG, fixed-time ramp metering, and integrated feedback control in terms of the total travel time savings in the scenarios of a single-ramp interweaving area and a large freeway corridor with a plurality of off-ramps and on-ramps (C. Wang et al., 2022).

In conclusion, research on deep reinforcement learning for traffic control optimization at the combined level in single-modal SUMO simulations is relatively sparse, with only a few studies exploring this complex area. The work by C. Wang et al. stands out by employing policy-based DRL to develop a centralized traffic control system that synchronizes multiple traffic management strategies, such as ramp metering and variable speed limits. This approach has demonstrated the potential to significantly enhance freeway mobility and reduce congestion, particularly in challenging traffic environments. The results of the study underscore the effectiveness of policy-based DRL in managing high-dimensional traffic scenarios, offering a promising direction for future research in integrated traffic control systems.

2.2. Deep reinforcement learning for traffic control optimization in multi-modal simulation of SUMO

Compared to the research about deep reinforcement learning for traffic control optimization in multi-modal simulation, much fewer studies conduct the research in multi-modal simulation. And it is often the case that only one kind of traffic control at intersection or vehicle level is used in each study.

2.2.1. Deep reinforcement learning for traffic control optimization at intersection level in multi-modal simulation of SUMO

As for the research regarding deep reinforcement learning for traffic control optimization at intersection level, most studies focus on using DQN as their training algorithms to optimize traffic light control.

Shen et al. utilized DQN for managing bus signal priority at intersections within a bus network, which considers the needs of both general traffic and pedestrians while ensuring priority access for each bus flow. The simulation experiments using SUMO demonstrate that the proposed method significantly decreases bus waiting times and the mean waiting time of all vehicles in comparison to the actuated traffic light control and the fixed-time traffic light control methods (Shen et al., 2023).

A system that uses connected vehicle technology to give priority to transit signals is proposed by Zhong et al., employing DQN for traffic light control optimization. The system utilizes Vehicle-to-Infrastructure (V2I) communication technology to gather real-time data on vehicle movement, traffic light phase information, and states of the road traffic. Experiments conducted using SUMO demonstrate that the proposed system has significant reductions in both vehicle delay and cumulative delay per passenger in comparison to the conventional traffic light control under the scenarios with low and medium traffic densities (Zhong et al., 2023).

Han et al. proposed a traffic light control method considering pedestrian behavior based on the Dueling Double Deep Q-Network (3DQN) algorithm, which integrates the strengths of Double DQN and Dueling DQN algorithms. The method simultaneously considers traffic safety, traffic efficiency, pedestrian waiting times, and vehicle waiting times. Simulation experiments conducted using SUMO on real intersection scenarios demonstrate that the proposed approach significantly reduces both pedestrian waiting times and the number of pedestrians waiting at crossroads compared with the Dueling DQN method (Han et al., 2022).

In summary, the application of deep reinforcement learning (DRL) for traffic control optimization at the intersection level in multi-modal SUMO simulations has primarily focused on enhancing traffic light management using DQN-based approaches. The studies reviewed have shown significant advancements in reducing vehicle and pedestrian waiting times while improving overall traffic flow. By incorporating considerations for various traffic participants, such as buses and pedestrians, these DRL methods have demonstrated superior performance compared to traditional control techniques. The success of these approaches highlights the potential of DRL to create more efficient and responsive traffic control systems in multi-modal urban environments.

2.2.2. Deep reinforcement learning for traffic control optimization at vehicle level in multi-modal simulation of SUMO

As for the research regarding deep reinforcement learning for traffic control optimization at vehicle level, only the existing study of Codeca and Cahill has been known so far.

Codeca and Cahill assessed the potential of employing multi-agent PPO (MAPPO) to coordinate journey plans in large-scale events like concerts and sports events. They utilize multi-agent MAPPO to develop synchronized plans that consider the availability, needs, and constraints of various traffic modes with the objective of increasing just-in-time arrival. The findings indicate that the devised plan effectively enhances the mean travel time and punctuality rates compared to a simplistic decision-making algorithm relying on estimated travel times (Codeca & Cahill, 2022).

In conclusion, the study by Codeca and Cahill offers valuable insights into the use of multi-agent PPO for optimizing vehicle-level traffic control in multi-modal SUMO simulations, particularly in the context of large-scale events. Their research demonstrates the effectiveness of coordinated transportation planning, taking into account various transportation modes and infrastructure constraints. The results highlight the potential of advanced DRL methods like MAPPO to significantly enhance travel efficiency and punctuality, providing a promising approach to managing complex traffic scenarios during major events.

2.3. Other traffic management strategies for traffic control optimization in multi-modal simulation of SUMO

In addition to deep reinforcement learning being used to optimize traffic control in multi-modal simulation of SUMO, other traffic management strategies have also been applied in this field.

Plenty of studies utilize different traffic management strategies to optimize traffic light control in multi-modal simulation of SUMO. Schmidt et al. proposed a framework to prioritize buses according to their occupancy and delay in a SUMO simulation set in Ingolstadt. They implemented four levels of prioritization interventions, adjusting traffic light cycles to advance preferred green phases based on bus priority levels. Simulation results under various conditions show that the proposed framework effectively reduces the number of stops and travel times for prioritized buses (Schmidt et al., 2024). Zeng et al. applied fixed cycle and induction control in SUMO simulation respectively to achieve bus priority. Their comparative experiment results show that the delay of buses under two controls reduces a lot while the parking frequency of social vehicles is almost not influenced (Zeng et al., 2023). Ying-Chuan Ni and Huang proposed a traffic light control optimization model to coordinate signal offsets along the Bus Rapid Transit (BRT) routes to reduce BRT vehicle delays without significantly impacting other traffic. Experiments set in the Taichung BRT System in Taiwan show that the proposed passive transit signal priority control significantly reduces transit delays, moderately affects other traffic, and enhances system capacity with reliable service (Ying-Chuan Ni & Huang, 2022). Nesmachnow et al. developed a parallel evolutionary algorithm to optimize public transport by coordinating traffic lights for Bus Rapid Transit systems. A case study on Garzón Avenue in Montevideo, Uruguay demonstrates that the developed algorithm significantly improves service quality, increasing the mean velocity of buses and other vehicles (Nesmachnow et al., 2019). Ali et al. introduced an adaptive traffic light control system using fuzzy logic combined with Webster and modified Webster's formulas. The experimental results show that the introduced methods have a better performance than traditional fuzzy logic-based and fixed-time traffic light control systems on the average delay of vehicles, velocity, and travel time (Ali et al., 2021). Colombaroni et al. proposed a model-based optimization procedure to design a control system that includes real-time bus priority, traffic light control coordination, and green light speed advisory for car drivers. Simulations on a main street with a tram line in Rome show that offline signal optimization and online signal priority can significantly reduce travel times for bus riders and overall traffic delays, and incorporating speed advisory for drivers into traffic light control optimization can improve delays for both drivers and transit passengers by enabling more efficient road use (Colombaroni et al., 2020). Lee and Wang presented a Person-based Adaptive traffic light control approach with Cooperative Transit signal priority, in which on-board units offer in-vehicle velocity advice to reduce delays and road-side units optimize traffic light control. Simulation results of SUMO show that the proposed approach significantly decreases delays for transit and auto passengers compared with preoptimized signal plans, especially when the occupancy weight factor is high, indicating its potential for signal preemption (Lee & Wang, 2022).

There are also some studies applying traffic management strategies for other traffic control optimization in multi-modal simulation of SUMO. Rakkesh et al. suggested a method to balance multi-modal traffic environments after roadblocks through Vehicular ad-hoc network scenario simulations by using the VEINS architecture to combine OMNeT++ and SUMO. The experiment results of the simulations set in Kandy and Colombo show that this method decreases the waiting time of vehicles and trip durations, and improves traffic flow compared to the method not using the equilibrating module (Rakkesh et al., 2017). Lu et al. presented a Green Light Optimal Speed Advisory (GLOSA) system in multi-modal traffic environments to reduce vehicle fuel consumption by integrating SUMO and OMNET++ using the VEINS architecture. The results of the experiment set in the Haidian District of Beijing demonstrate that the proposed system can largely reduce waiting time and CO2 emissions in multi-modal traffic settings (Lu et al., 2020).

In summary, a variety of other traffic management strategies have also been applied to optimize traffic control in multi-modal simulation of SUMO. These strategies range from bus prioritization frameworks and adaptive traffic light control systems to algorithms designed for public transport coordination and fuel consumption reduction. Whether focusing on prioritizing buses, coordinating signal offsets, or balancing traffic environments after disruptions, each method demonstrates significant improvements in traffic performance like delays, waiting times, and emissions. These approaches collectively highlight the versatility and effectiveness of traffic control optimization strategies within the SUMO simulation platform for diverse traffic conditions and multi-modal environments.

2.4. Discussion

The reviewed literature has shown that the potential of DRL for traffic control optimization is significant, and has provided valuable insights into the application of DRL for traffic control optimization. However, there are still several limitations existing in the relevant literature.

One of the limitations is the limited consideration of using total travel time as the optimization indicator for the application of DRL in multi-modal simulation of SUMO. Most existing studies focus on scenarios with only passenger vehicles, while real-world traffic systems also include other traffic modes. Even though there are few relevant studies considering multiple traffic modes, they did not use total travel time as the optimization indicator. Total travel time includes not only the time spent waiting at traffic lights but also the time spent moving through the network. And total travel time can also capture the impact of various traffic conditions, including congestion, traffic light delays, and road capacity, which makes it a more holistic indicator than other factors like total waiting time, total delay, and so on.

The second limitation is that the potential of using single agent to control multiple traffic lights remains unexplored. So far, for the case of multiple intersections, related studies have used multi-agent deep reinforcement learning methods but have not considered using a single agent to control multiple intersections.

Another limitation is that the performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes is not explored. Exploring the performance of deep reinforcement learning for traffic light control optimization in road networks of different sizes is crucial because it allows for a comprehensive understanding of the scalability and adaptability of DRL algorithms. Different road network sizes present unique challenges and complexities, such as varying traffic patterns, congestion levels, and infrastructure constraints. Applying DRL only to a single road network cannot well show the strengths, limitations, and its applicability for different situations.

In addition, the evaluation of DRL-based traffic light control policies is very limited in the reviewed literature. Most of the related studies only mention the training process and do not conduct a reasonable and diversified evaluation of the DRL training model. Therefore, the effect and applicability of DRL on traffic control can not be fully reflected.

Based on the findings of the literature review, several points are presented to address the current limitations:

- **Expand the scope of DRL research to use total travel time as the optimization indicator in multi-modal simulation.**
  Total travel time accounts for the entire journey of a vehicle from start to finish, providing a more comprehensive measure of traffic efficiency.
- **Explore the use and effectiveness of using different single-agent DRL to control multiple traffic lights in multi-modal simulation.**
  A single-agent DRL system has the potential to optimize traffic flow holistically by considering the states and the impact of decisions on the entire network rather than making independent decisions for each intersection, which might lead to better training performance compared to the multi-agent DRL system.
- **Explore the performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes.**
  By exploring the performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes, the effectiveness of different DRL in different situations can be fully demonstrated.
- **Evaluate different DRL in terms of different aspects.**
  By evaluating different deep reinforcement learning methods considering more factors, the advantages and disadvantages of each deep reinforcement learning can be analyzed from a more comprehensive perspective, showing the effect and applicability of each method.

By addressing these research limitations, DRL has the potential to revolutionize traffic control and significantly improve traffic efficiency, safety, and sustainability.

2.5. Conclusion

This literature review provides a comprehensive overview of the application of deep reinforcement learning for traffic control optimization in multi-modal simulation of SUMO at three different levels (intersection level, vehicle level, and combined level) as well as other traffic management strategies in this field. Plenty of studies on DRL-based traffic control have demonstrated promising results in improving traffic efficiency and reducing congestion. However, there are still some limitations in the existing relevant literature to be addressed. Several points are presented to solve the limitations existing in the relevant research.

3

Research methodology

This chapter illustrates the research methodology used for this research in detail. In this research, the process of traffic light control is modeled as a Markov Decision Process. And different single-agent and multi-agent deep reinforcement learning methods including PPO, A2C, MAPPO, MAA2C, and MADQN are applied to optimize traffic light control in multi-modal simulation as all of them support multiple action choices for one agent at each step. In order to test the effectiveness and performance of these DRL algorithms, fixed traffic light control and max-pressure traffic light control are also implemented for comparison.

3.1. Markov Decision Process

A Markov Decision Process (MDP) is introduced by Puterman, which provides a framework for representing and analyzing the interaction between an agent and its environment. It is a mathematical model used for sequential decision-making, where the cost and transition functions depend solely on the current state and action (Puterman, 1990). The goal of an MDP model is to determine an optimal policy that maximizes expected rewards over time (Puterman, 1990).

An MDP is defined by a collection of states, a set of actions available to the agent in each state, a transition model that specifies the probability of moving from one state to another given a particular action, and a reward function that assigns a numerical reward to each state-action pair as shown below:

1. **States:** The possible configurations of the environment.
2. **Actions:** The possible choices for the agent to execute.
3. **Transition probabilities:** The probabilities of transitioning from one state to another after taking a specific action.
4. **Rewards:** The immediate incentives received for taking actions in particular states.

An MDP model is formulated as follows:

$$M = (S, A, T, R) \tag{3.1}$$

where S is a set of states, A is a set of actions, T: S x $A \rightarrow$ S is a transition probability function that maps from a state-action pair to the probability of transitioning to a new state, R: S x $A \rightarrow R$ is a reward function that maps from a state-action pair to the immediate reward received after taking that action in that state.

Traffic light control in multi-modal simulation using SUMO can be effectively formulated as an MDP for several reasons:

1. **State Dependence:** MDPs rely on the principle of Markov property, where the future state depends on the current state and chosen action. This aligns well with traffic light control, where the current traffic conditions fully determine the potential outcomes of the next action.
2. **Discrete States and Actions:** Even though the real traffic environment is continuous, SUMO discretizes both states and actions for simulation purposes. This discrete nature makes it compatible with the discrete framework of MDPs.

3. **Sequential Decision Making:** Traffic light control involves making a series of decisions over time, where each decision affects the subsequent state and possible future decisions. This sequential decision-making process mirrors the core concept of MDPs.
4. **Reward Signal Definition:** MDPs require a reward signal to evaluate the effectiveness of chosen actions. In traffic light control simulations, various relevant indicators can be defined as rewards, such as travel time, queue length, and fuel consumption.
5. **Reinforcement Learning Compatibility:** MDPs offer a solid foundation for applying reinforcement learning algorithms, which are particularly suited to multi-agent environments. By learning from reward feedback, the control system can adapt its actions dynamically to optimize chosen performance metrics.

3.1.1. State design

In deep reinforcement learning, states are the core representation of the environment, encapsulating all the essential information an agent requires to make decisions. A state is like a snapshot of the environment at a specific moment, and it is vital for the agent to learn and improve its behavior over time. The significance of states lies in their role in reflecting the current condition of the environment, which directly impacts the actions of the agent and subsequent learning process (AlMahamid & Grolinger, 2021).

Currently, there are different kinds of states being used for the research regarding deep reinforcement learning for traffic light control optimization. Among them, the queue length of the incoming lanes of the intersection is commonly used as an essential observation element (Chen et al., 2019; Guo et al., 2019; Bouktif et al., 2021; Louw et al., 2022; Shen et al., 2023). Some research chooses to discretize the space to collect more detailed information about vehicle states like presence, location, speed, and direction in which the vehicle passes through the intersection, at each section of the road (Guo et al., 2019; Bouktif et al., 2021; B. Wang et al., 2022; D. Li et al., 2020; Ma et al., 2021; Shen et al., 2023; Han et al., 2022). In addition, different factors like mean speed (Chen et al., 2019; D. Li et al., 2020), traffic volume (D. Li et al., 2020; Z. Li, Xu, & Zhang, 2021), the current green phase of traffic light (Bouktif et al., 2021; Ma et al., 2021; Shen et al., 2023; Han et al., 2022), the ratio of the current green time to the maximum green time, lane occupancy (Yu et al., 2023; Ma et al., 2021), number of phase switches during the last ten action steps, bus space headway, actions taken by the immediate neighborhood traffic light (Yu et al., 2023), the release time of the vehicle phase, the red light time of the vehicle phase, and the red light waiting time of pedestrian phase (Shen et al., 2023) are also applied to represent the states of the traffic.

In this research, the traffic information of each edge in the road network is assumed to be known. The weighted traffic volume (i.e. number of vehicles) and weighted queue length at each intersection are used as the states for observation, as they can well reflect the traffic conditions on the road and help realize the transit priority.

The weighted traffic volume of the incoming lanes of each green phase for each traffic light at each time step is calculated by the sum of the results of multiplying the total number of each vehicle type in the incoming lanes of each green phase for each traffic light at each time step by the weight of the corresponding vehicle type as shown in Equation 3.2.

$$Vol_t^{mp} = \sum_{v \in V, l \in L^{mp}} \alpha_v * nv_t^{mlv} \quad \forall t \in T, m \in M, p \in P^m \tag{3.2}$$

where Vol_t^{mp} is the weighted traffic volume of the incoming lanes of the green phase p for the traffic light m at the time step t, nv_t^{mlv} is the number of vehicles of vehicle type v of the incoming lane l of the traffic light m at the time step t, V is the vehicle type set, α_v is the weight for the vehicle type v, M is the traffic light set, P^m is the green phase set of the traffic light m, and L^{mp} is the lane set of the phase p of the traffic light m, T is the time step set, the length of a time step δ is set to be 5 seconds, because the length of a time step has to be larger than yellow time (3 seconds) to ensure that the new green phase chosen at the last time step can be activated before the new green phase. (δ is the number of seconds between consecutive actions of an agent. When an agent selects a new green phase different from the green phase selected at the last time step, the corresponding traffic light will first shift to the yellow phase for 3 seconds. If δ is no larger than 3 seconds, the traffic light would skip the green phase selected at the last time step to execute the new selected green phase.)

The weighted queue length of the incoming lanes of each green phase for each traffic light at each time step is calculated by the sum of the results of multiplying the total queue length of each vehicle type in the incoming lanes of each green phase for each traffic light at each time step by the weight of the corresponding vehicle type as shown in Equation 3.3.

$$Que_t^{mp} = \sum_{v \in V, l \in L^{mp}} \alpha_v * ns_t^{mlv} \quad \forall t \in T, m \in M, p \in P^m \tag{3.3}$$

where Que_t^{mp} is the weighted queue length of the incoming lanes of the green phase p for the traffic light m at the time step t, ns_t^{mlv} is the number of stopping vehicles of vehicle type v of the incoming lane l of the traffic light m at the time step t.

According to the research conducted by Network for Transport Measures (NTM) which is a non-profit organization striving to develop a unified framework for assessing the environmental impact of diverse transportation modes encompassing both freight and passenger travel, the usual passenger occupancy of the bus is 15-20 (NTM, 2024). Thus, in this research, the weight for bus α_{bus} is set to be 20, and the weight for passenger car $\alpha_{passenger}$ is set to be 1.

As for multi-agent DRL, each agent controls one traffic light and the state of the agent is the state of the controlled traffic light, including the normalized weighted volume and the normalized weighted queue length, as shown in Equation 3.4.

$$s_t^m = [\frac{Vol_t^{mp}}{200}, \frac{Que_t^{mp}}{200}] \quad \forall t \in T, m \in M, p \in P^m, s_t^m \in S^m \tag{3.4}$$

where s_t^m is the state for the traffic light m at the time step t, S^m is the state set of the traffic light m, P^m is the green phase set of the traffic light m.

As for single-agent DRL, one single agent controls all the traffic lights and the state of the agent is the combination of the states of all the traffic lights as shown in Equation 3.5.

$$s_t = [s_t^1, s_t^2, ..., s_t^m] \quad \forall t \in T, m \in M, p \in P^m, s_t \in S \tag{3.5}$$

3.1.2. Action design

In deep reinforcement learning, an action is a decision made by the agent that causes a change of the current state of the environment, and it results in a new state. The action space can be either discrete, consisting of a finite set of actions, or continuous, comprising an infinite set of actions within certain limits. The action selected in each state is dictated by the policy, which is a mapping from states to actions (Mohan et al., 2024).

As for the action space, the most prevailing and widely used action is to choose one green phase at each step (Genders & Razavi, 2016; Chen et al., 2019; Guo et al., 2019; B. Wang et al., 2022; D. Li et al., 2020; Z. Li, Xu, & Zhang, 2021; Ma et al., 2021; Louw et al., 2022; Shen et al., 2023; Zhong et al., 2023; Han et al., 2022). Among these research as mentioned above, some of them also determine the associated phase duration on the basis of green phase selection (Guo et al., 2019; Bouktif et al., 2021), while C. Li et al. choose to decide the green phase time under the framework of a fixed green phase sequence (C. Li et al., 2020). In particular, Chen et al. adopt two action sets with both choosing the green phase, and extending or shortening the green phase duration difference (Chen et al., 2019).

In this research, the action for each traffic light is assumed to be which green phase to activate at each time step so that the number of actions for each episode is fixed, which is beneficial to the learning process. Every time a green phase change occurs, the next phase is preceded by a corresponding yellow phase with a duration of 3 seconds.

For multi-agent DRL, each agent controls one traffic light, so the action space of each agent has to be Discrete, which allows each agent to determine the action of the controlled traffic light as shown in Equation 3.6.

$$action_t^m = p \quad \forall t \in T, m \in M, action_t^m \in A^m, p \in P^m \tag{3.6}$$

where A^m is the action set of the traffic light m. At each time step t, each agent chooses the action $action_t^m$ for the controlled traffic light m from its green phase set P^m. The number of actions of the traffic light m is equal to the number of green phases of it. That is, $|A^m| = |P^m|$.

For single-agent DRL, one single agent has to control the actions of all the traffic lights. Therefore, the action space of the agent in this research has to be MultiDiscrete, which allows one single agent to determine the action for each traffic light at each time step as shown in Equation 3.7.

$$a_t = [action_t^1, action_t^2, ..., action_t^m] \quad \forall t \in T, m \in M, a_t \in A, action_t^m \in A^m \tag{3.7}$$

where A is the action set of the agent. At each time step t, the agent chooses the action $action_t^m$ for the traffic light m from its green phase set P^m.

3.1.3. Reward design

In deep reinforcement learning, the reward function is essential for guiding the learning process of the agent, as the reward can directly reflect how well the agent executes an action under a certain set of states in the environment, which in turn influences the future actions of the agent.

Regarding the reward design of DRL for traffic light control optimization, by reviewing the relevant literature, it can be found that waiting time and queue length are the most commonly used factors. Some related studies use the difference of waiting time for reward computation (B. Wang et al., 2022; Liang et al., 2018; Ma et al., 2021; Zhong et al., 2023) while others use the difference of queue length instead (Guo et al., 2019; Bouktif et al., 2021). There are also some studies that directly use the waiting time as the reward (C. Li et al., 2020; Louw et al., 2022) while others directly use queue length as the reward (Bouktif et al., 2021). Z. Li, Xu, and Zhang use both waiting time and queue length as the reward (Z. Li, Xu, & Zhang, 2021). Different from previous mentioned research, D. Li et al. use the ratio of waiting time to the set waiting time threshold in the reward design (D. Li et al., 2020). Moreover, there is also some research combining one of the two factors with another one. Chen et al. consider both the net outflow (global reward), and the absolute negative difference between queue lengths (local reward) in the whole reward function (Chen et al., 2019). Han et al. also consider the vehicle throughput in their reward design (Han et al., 2022). Particularly, Genders and Razavi adopt the change in cumulative vehicle delay between actions as the reward (Genders & Razavi, 2016).

In this research, the weighted travel time of vehicles in all the incoming lanes of the intersection controlled by each traffic light at each time step is calculated by multiplying the weighted traffic volume of the incoming lanes of each green phase for each traffic light at each time step by the length of a time step as shown in Equation 3.8.

$$TTS_t^m = \sum_{p \in P^m} Vol_t^{mp} * \delta \quad \forall t \in T, m \in M \tag{3.8}$$

where TTS_t^m is the weighted travel time of vehicles in all the incoming lanes of the intersection controlled by the traffic light m at the time step t.

And the total weighted travel time of vehicles in all the incoming lanes of all the intersections at each time step is the sum of the weighted travel time of vehicles in all the incoming lanes of the intersection controlled by each traffic light at each time step as shown in Equation 3.9.

$$TTS_t = \sum_{m \in M} TTS_t^m \quad \forall t \in T, m \in M \tag{3.9}$$

where TTS_t is the total weighted travel time of vehicles in all the incoming lanes of all the intersections at the time step t.

For multi-agent DRL, the reward at each time step for each agent r_t^m in this research is defined as the normalized negative total weighted travel time of vehicles in all the incoming lanes of the traffic light controlled by each agent at the time step t as shown in Equation 3.10.

$$r_t^m = -\frac{TTS_t^m}{1000} \quad \forall t \in T, m \in M, r_t^m \in R^m \tag{3.10}$$

where r_t^m is the reward of the traffic light m at the time step t, R^m is the reward set of the traffic light m.

And the total reward at each time step is defined as the sum of reward of all the agents as shown in Equation 3.11.

$$r_t = \sum_{m \in M} r_t^m \quad \forall t \in T, m \in M, r_t \in R \tag{3.11}$$

where r_t is the total reward at the time step t, R is the total reward set.

For single-agent DRL, the reward at each time step r_t of the agent in this research is defined as the negative total weighted travel time of vehicles in all the incoming lanes of all the traffic lights at the time step t as shown in Equation 3.12.

$$r_t = -TTS_t \quad \forall t \in T, r_t \in R \tag{3.12}$$

3.2. Q-Learning

As stated in the paper of Neufeld and Sester, Q-Learning is well-suited for solving Markov Decision Process (MDP) (Neufeld & Sester, 2023). Q-Learning, developed by C. J. C. H. Watkins, is a reinforcement learning algorithm that learns an action-value table (Q-table) to map states to the expected value of taking a particular action in that state (C. J. C. H. Watkins, 1989).

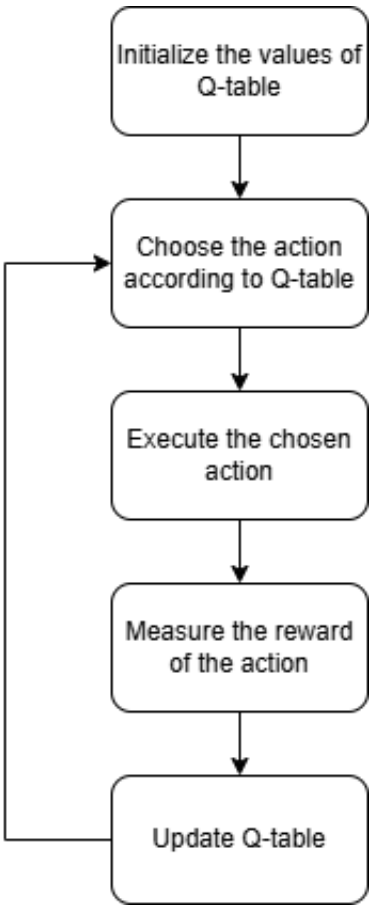


Figure 3.1: Architecture of Q-Learning

As shown in Figure 3.1, the Q-Learning algorithm mainly works in the following way:

1. Initialize an empty Q-table with zeros
2. For each episode, do:
   (a) Initialize an empty state vector with zeros
   (b) Repeat until episode termination
      i. For each step, do:
         A. Observe the current state vector s_t
         B. Choose an action a_t.
         C. Execute action a_t on traffic light s_t
         D. Observe the next state vector s_{t+1} after executing action a_t
         E. Calculate a reward r(s_t|s_{t+1}, a_t)
         F. Update Q-table with r(s_t|s_{t+1}, a_t)
      ii. If episode termination condition is met, break out of loop
3. Return Q-table as output.

However, Q-Learning faces several challenges when applied to traffic control:

- **High-Dimensional State Space:** The state space of a traffic system is very large and complex, making it difficult for Q-Learning to learn an accurate Q-table. The state can be represented by a combination of traffic volume, traffic density, signal phases, and other factors. This large number of variables makes it challenging for Q-Learning to learn the Q-values effectively.
- **Sparse Rewards:** Traffic control often involves delayed rewards, making it difficult for Q-Learning to reinforce desired behaviors. The reward in traffic control is typically the overall traffic efficiency, which is a long-term consequence of the actions taken. This delayed reward makes it difficult for Q-Learning to learn the optimal policy.
- **Exploration vs. Exploitation Tradeoff:** Q-Learning must balance exploration (trying new actions to discover better policies) with exploitation (taking the best known action based on the current Q-table). In traffic control, it is crucial to strike a balance between these two strategies to avoid inefficient traffic flow. Over-exploration can lead to poor traffic performance, while over-exploitation can prevent the agent from discovering better policies.

3.3. Deep Q-Network

As can be seen from Table 2.1, many relevant studies apply Deep Q-Network for traffic light optimization, which shows that DQN has already been a commonly used DRL method for traffic control optimization in traffic simulation of SUMO.

The Deep Q-Network (DQN) algorithm was first introduced by Mnih et al. to address the limitations of Q-Learning (Mnih et al., 2013). It is a pioneering algorithm in the field of deep reinforcement learning that utilizes a convolutional neural network (CNN) to approximate the Q-values, enabling the agent to generalize across similar states and actions based on their representations in the neural network. This architecture allows DQN to handle complex and high-dimensional state spaces more effectively than traditional Q-Learning (Mnih et al., 2015).

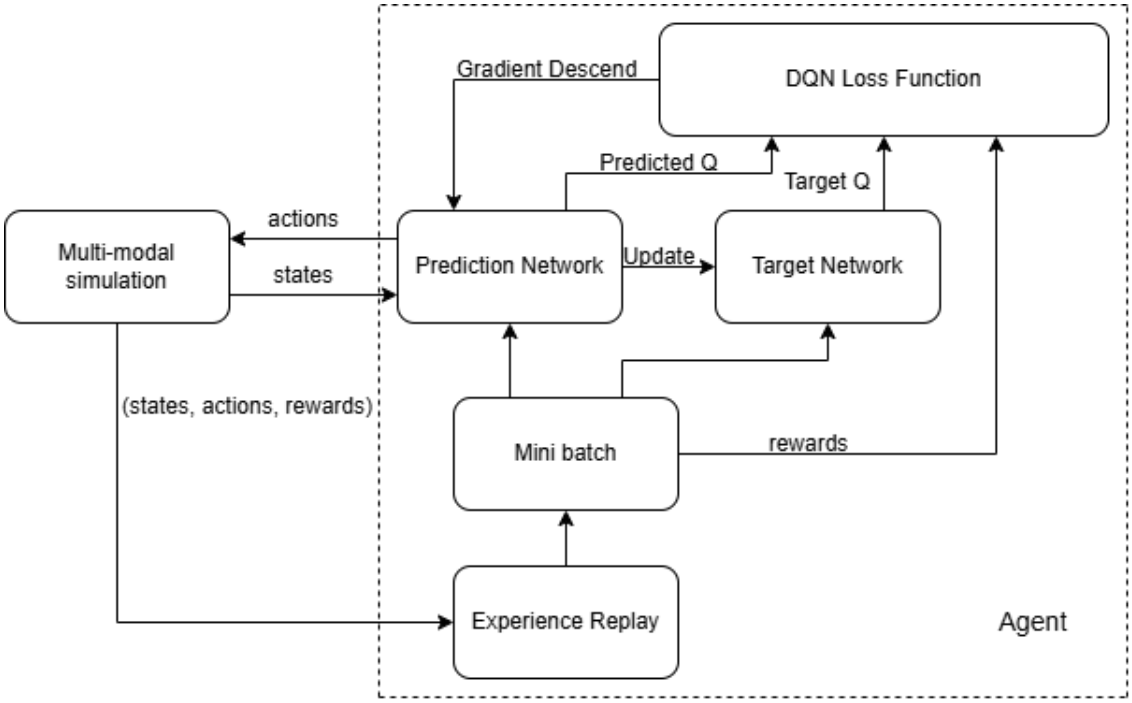


Figure 3.2: Architecture of a Deep Q-Network

As can be seen from Figure 3.2, the DQN agent interacts with the environment and stores its experiences (state, action, reward, next state) in an experience replay buffer. From this buffer, random mini-batches of experiences are sampled and passed through the prediction network, which estimates Q-values for various actions. To stabilize learning, a target network, updated less frequently, is used to generate target Q-values. These target values are compared with the predicted Q-values using the DQN loss function, and the resulting error is used to update the prediction network, gradually improving the performance of the agent.

So far, DQN has been successfully applied in various tasks including traffic light control and has shown significant effects in improving traffic flow, reducing congestion, and optimizing traffic light control as stated in Chapter 2.

3.4. Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a deep reinforcement learning algorithm introduced by Schulman et al. and is part of the broader category of policy gradient methods, where the policy directly maps states to actions.

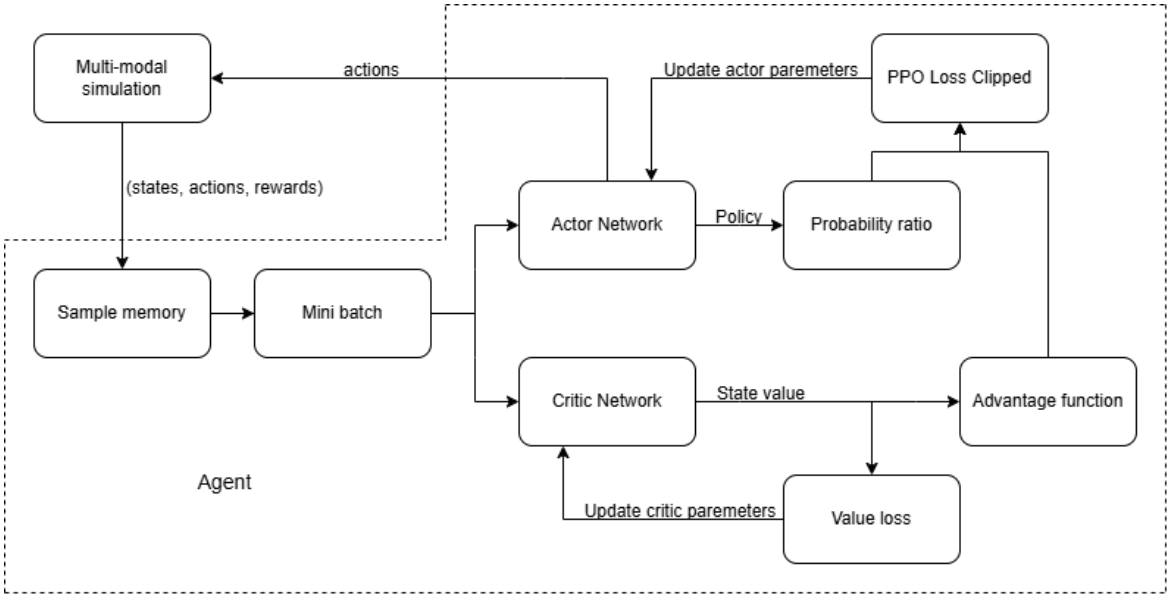


Figure 3.3: Architecture of Proximal Policy Optimization

As can be seen from Figure 3.3, the PPO agent interacts with the environment, storing its experiences in a sample memory. Mini-batches of these experiences are drawn from memory and passed to both the actor and critic networks. The actor network calculates the probability ratios, which are then clipped to ensure stable policy updates. Meanwhile, the critic network computes the value function, providing an estimate of the expected returns. The advantage function is used to guide the policy improvement by comparing actual returns with the estimated value. The PPO loss is a combination of the clipped probability ratio (for stable policy optimization) and the value loss (for accurate state-value predictions), and this loss is minimized to improve both the actor and critic networks.

So far, PPO has gained widespread popularity for its ability to train complex policies in a stable and efficient manner and has been successfully applied in numerous domains, including robotics, game playing, and autonomous systems. Its ability to handle high-dimensional action spaces and complex environments has made it a go-to algorithm for many reinforcement learning practitioners. Additionally, the balance of PPO between computational efficiency and training stability makes it suitable for both research and practical applications.

3.5. Advantage Actor Critic

Advantage Actor-Critic (A2C) is a synchronous version of the Asynchronous Advantage Actor-Critic (A3C) algorithm which is a deep reinforcement learning method introduced by Mnih et al. that enhances the stability and efficiency of training by running multiple actors in parallel, each interacting with separate environments (Mnih et al., 2016; Dhariwal et al., 2017).

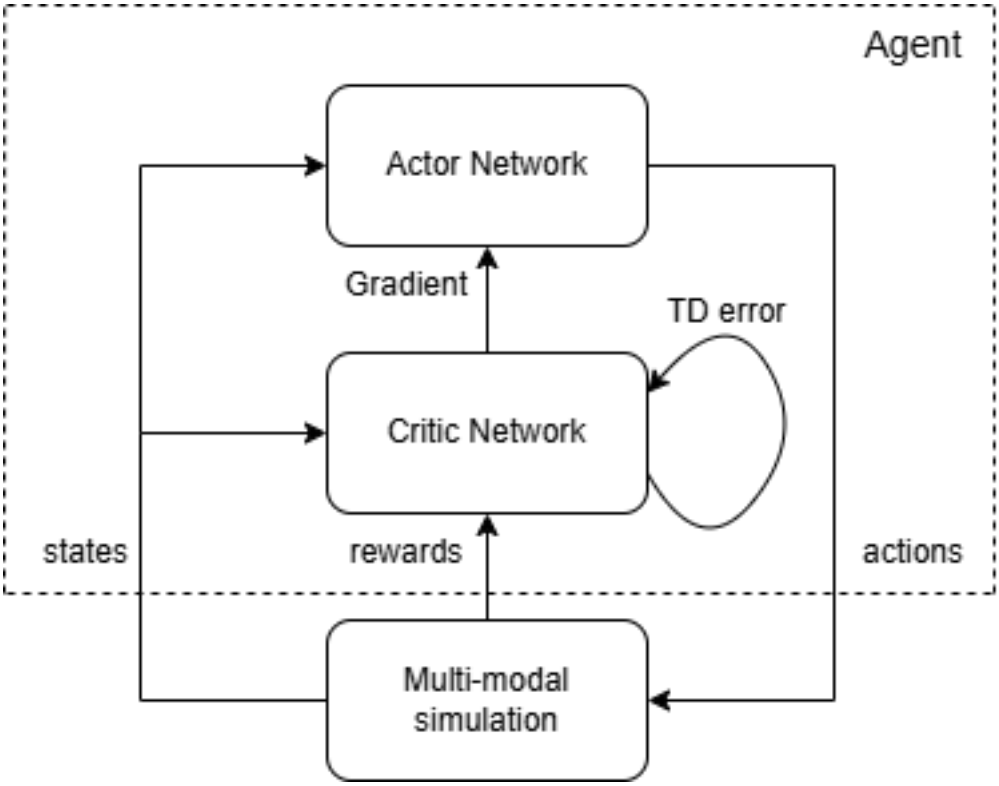


Figure 3.4: Architecture of Advantage Actor Critic

As can be seen from Figure 3.4, the A2C agent interacts with the environment, gathering information about the current state. The actor network uses this state to determine the optimal action policy, deciding which actions the agent should take. Simultaneously, the critic network evaluates the state by estimating the value function, which assesses the expected rewards for the given state. The actor and critic networks work together: the actor optimizes the policy, and the critic helps guide the actor by providing feedback on how good or bad the chosen actions are based on the expected future rewards.

While A3C uses multiple independent neural networks to generate trajectories and update parameters asynchronously, A2C achieves this synchronously. This synchronous update mechanism is more cost-effective and performs better than asynchronous implementations and can leverage GPUs efficiently (OpenAI, 2017). So far, A2C has been widely used in reinforcement learning tasks, including Atari games and continuous control (Mnih et al., 2016; Dhariwal et al., 2017).

3.6. Multi-agent deep reinforcement learning

Multi-agent deep reinforcement learning (MADRL) builds on traditional DRL by applying it to settings with multiple agents that may have individual goals or work towards a common objective. In these scenarios, agents make decisions through interactions with their environment and each other, refining their strategies based on the rewards or feedback they receive. Based on the way of learning, multi-agent deep reinforcement learning can be classified into two categories: centralized multi-agent DRL and decentralized multi-agent DRL.

In centralized multi-agent deep reinforcement learning, agents either follow a shared policy or are directed by a central controller with access to the global state of the environment. This centralized method enables better coordination among agents, as the central controller can evaluate the actions and states of all agents at once. This can result in more optimized collective strategies, especially in situations where agent interactions are intricate and closely linked (Lowe et al., 2017).

However, centralized multi-agent deep reinforcement learning can be demanding in terms of computation and may struggle to scale with an increasing number of agents or a larger environment. It can also face communication bottlenecks and delays, as the central controller must process and respond to

data from all agents in real time. This can be a significant disadvantage in dynamic settings where quick decision-making is crucial.

In contrast, decentralized multi-agent deep reinforcement learning enables each agent to function and learn on its own, relying solely on its local observations and experiences. Each agent formulates its own policy, striving to enhance its performance while taking into account the actions of nearby agents as part of the environment. Decentralized methods are generally more scalable and robust, as they reduce the reliance on a single point of failure and are better equipped to manage the complexity of environments with numerous agents.

In summary, decentralized multi-agent deep reinforcement learning is used in this research due to its scalability and robustness in complex urban environments.

3.7. Fixed traffic light control

Fixed traffic light control is the most common and straightforward method for managing traffic flow at intersections. It operates on a predetermined schedule, cycling through green, yellow, and red lights according to a fixed timing plan which is based on average traffic conditions and remains unchanged regardless of real-time traffic variations. The simplicity and predictability of fixed traffic light control make it an easily implemented and widely used approach.

However, fixed traffic light control also has some disadvantages. The most significant is its lack of adaptability. Because the timing of the lights does not change in response to real-time traffic conditions, fixed traffic light control can lead to unnecessary delays during off-peak hours or when traffic is unusually light. Additionally, it can struggle to effectively manage traffic during unexpected surges or incidents.

Despite these limitations, fixed traffic light control remains a viable option for managing traffic in many situations, particularly in areas with predictable traffic patterns or limited resources for more advanced control systems. The research conducted by Allsop illustrates the effectiveness of fixed light control in situations where traffic demand is relatively predictable (Allsop, 1974).

3.8. Max-pressure traffic light control

Max-pressure traffic light control, designed by Varaiya, is an advanced adaptive traffic light control method for managing traffic at signalized intersections in real time (Varaiya, 2013). Unlike fixed traffic light control, max-pressure traffic light control dynamically adjusts the phase based on the current traffic state, aiming to minimize congestion and delays.

The core idea behind max-pressure control is to calculate the pressure of each phase, which is a measure of the imbalance between the incoming and outgoing traffic on links of each phase connected to the intersection. By prioritizing movements that relieve the most pressure, this approach helps to prevent the buildup of queues and smooths traffic flow across the network (X. Wang et al., 2022).

So far, max-pressure traffic light control has showed its effectiveness in traffic light control optimization. Comparative experiments conducted by Varaiya demonstrate that the max-pressure control system offers a more reliable and efficient solution for managing traffic at intersections compared to other local controllers such as priority service and fully actuated control (Varaiya, 2013).

4

Case study

This chapter describes the case study of the research in detail including the simulation setups, the implementation of different DRL algorithms (PPO, A2C, MAPPO, MAA2C, MADQN), the comparison methods (fixed, max-pressure), the computation setups as well as the experimental evaluation for these methods.

4.1. Simulation setups

4.1.1. Network setups

Network setups are conducted by using *netedit*, which is a graphical network editor in SUMO to create the road network for simulation (Lopez et al., 2018).

This research conducted training under three different cases including a road network of single intersection, a road network of 2*2 intersections, and a road network of 3*2 intersections as shown in Figure 4.1, Figure 4.2, and Figure 4.3 respectively. For all the cases, each edge has three lanes and the length of each edge is 1 km. At the end of the three lanes, only the left turn, straight ahead, or right turn is allowed respectively. And there are traffic lights controlling the traffic flow at each intersection of both road networks.

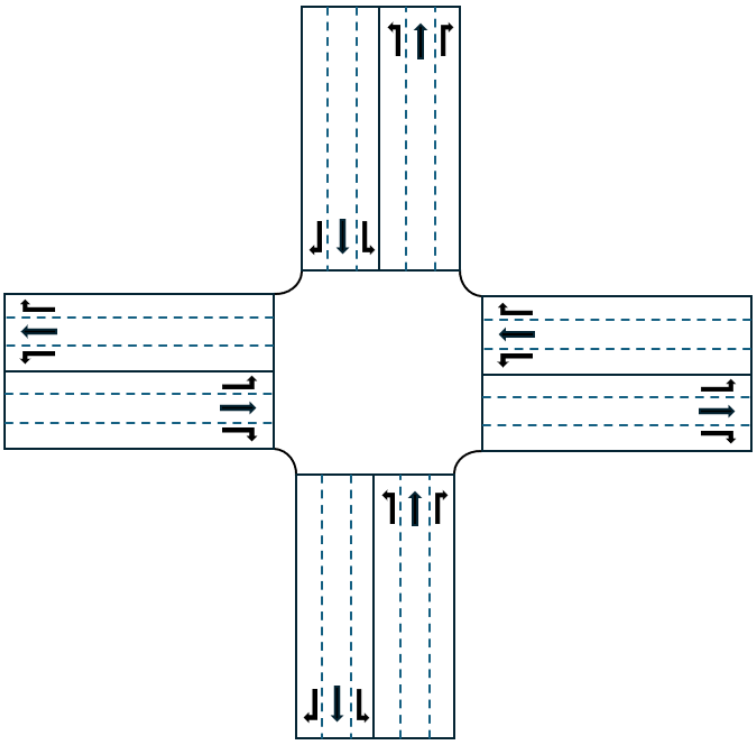


Figure 4.1: Case 1: road network of single intersection

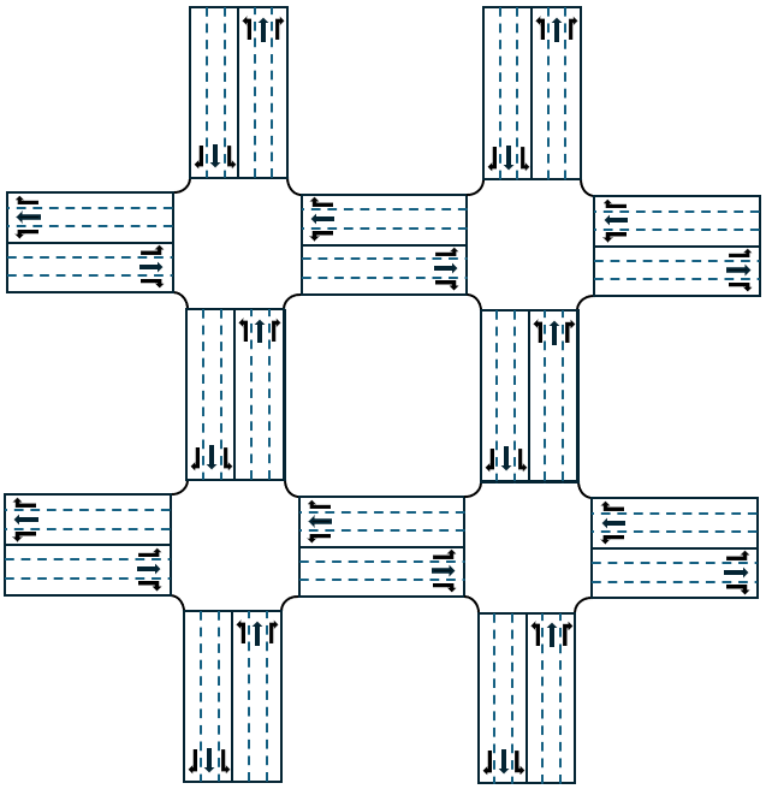


Figure 4.2: Case 2: road network of 2*2 intersections

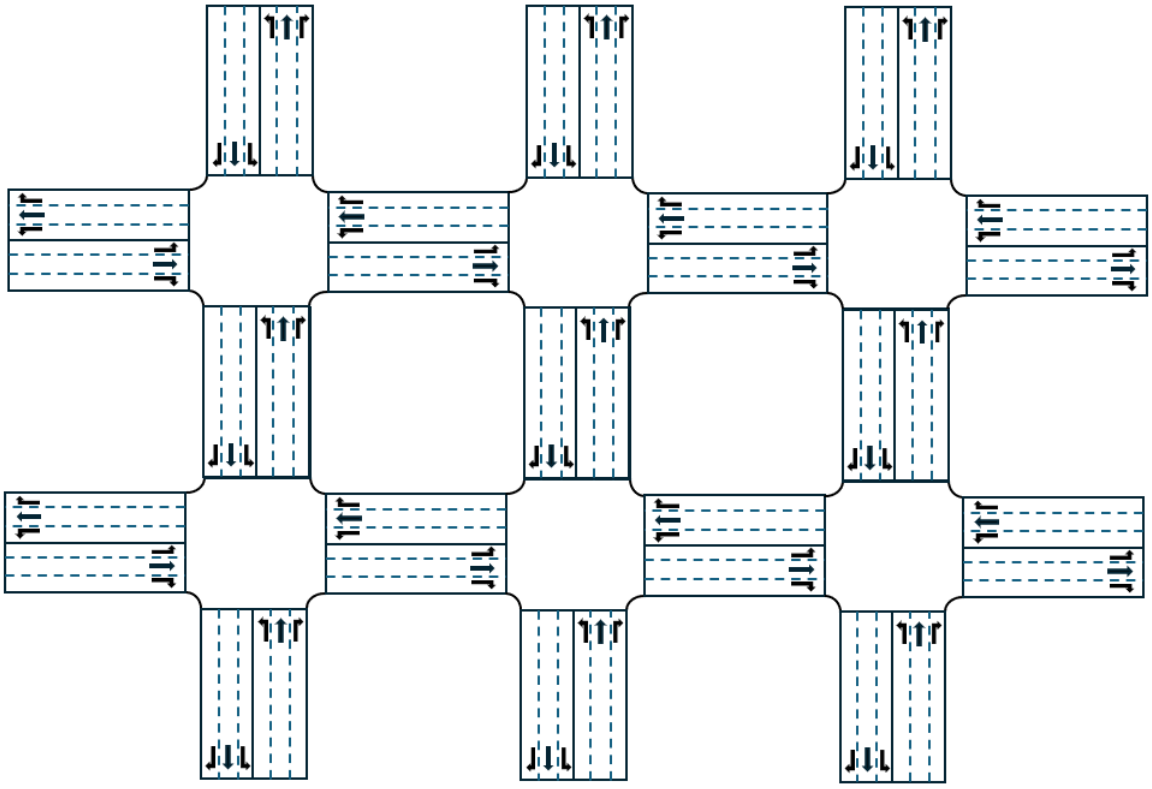


Figure 4.3: Case 3: road network of 3*2 intersections

4.1.2. Phase setups

As for the phase setups, the typical phase settings are applied in the case study. As shown in Figure 4.4, the straight and right turns of each pair of opposite roads are designed in one separate phase, and the left turns of each pair of opposite roads are also designed in one separate phase to ensure smooth passage of left-turning traffic.

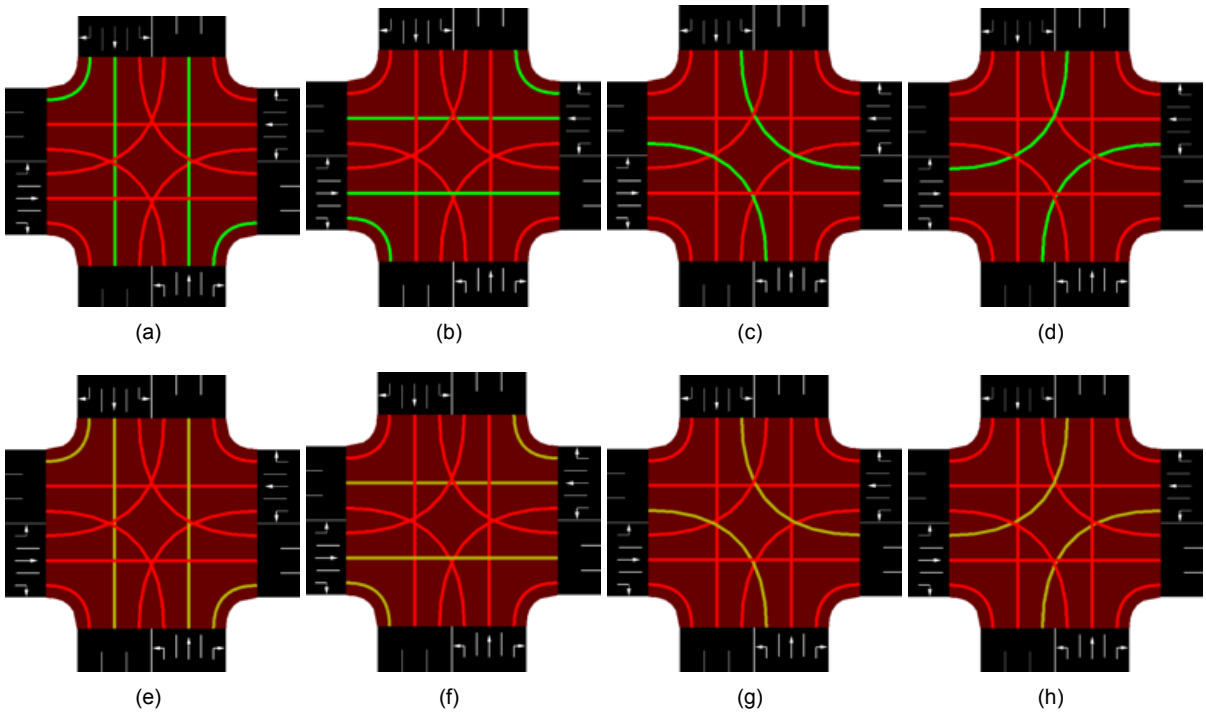


Figure 4.4: Phase settings

4.1.3. Public transport setups

For Case 1, there is a bus line with two opposite routes designed as shown in Figure 4.5.

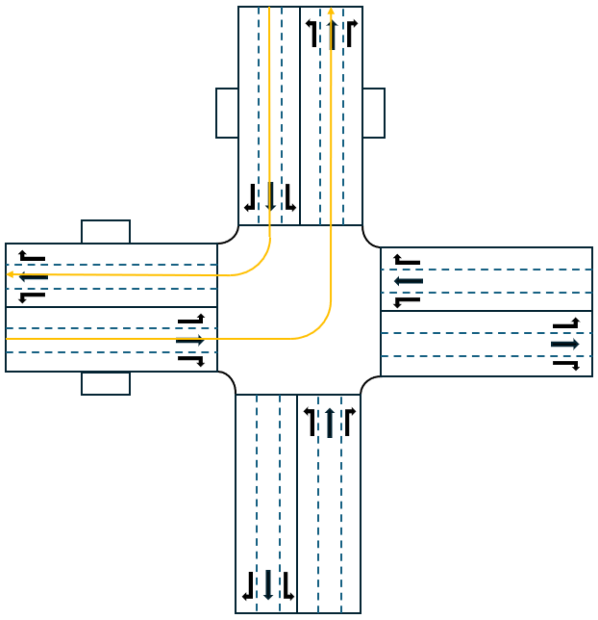


Figure 4.5: Bus routes of road network of single intersection

For Case 2, there are two bus lines with two opposite routes designed as shown in Figure 4.6.

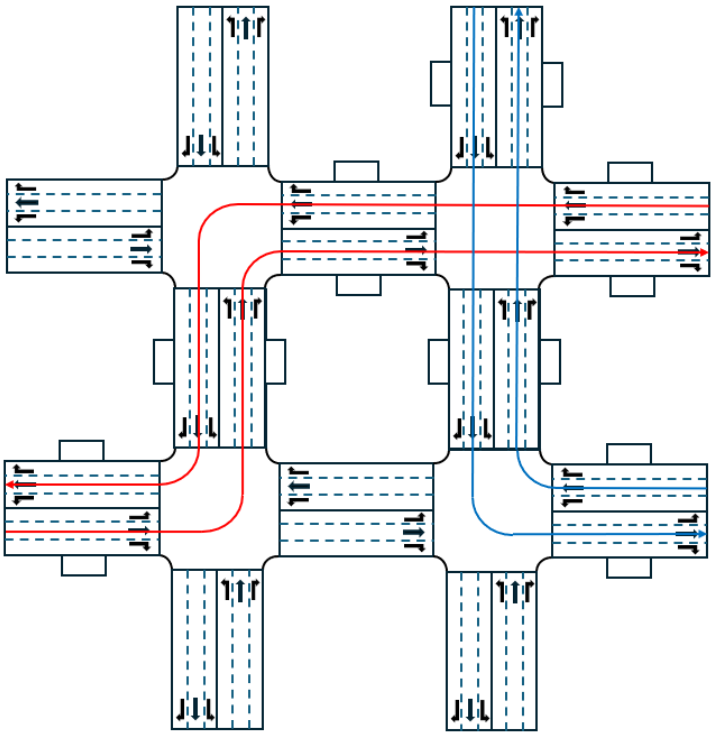


Figure 4.6: Bus routes of road network of 2*2 intersections

For Case 3, there are also two bus lines with two opposite routes designed as shown in Figure 4.7.

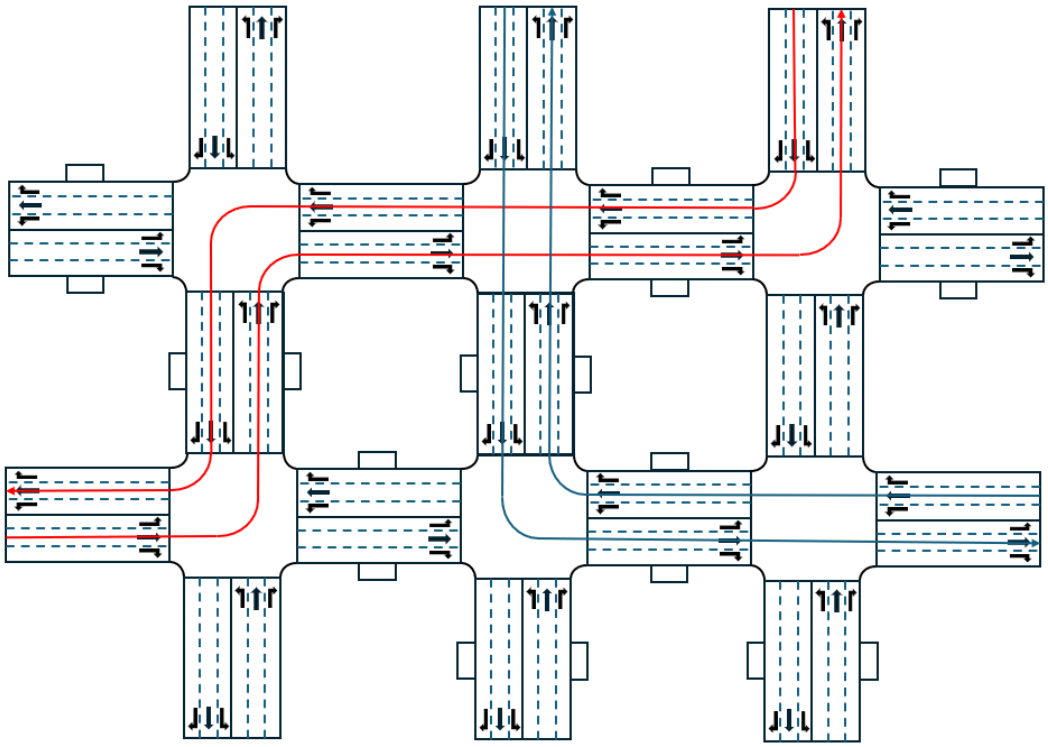


Figure 4.7: Bus routes of road network of 3*2 intersections

To simplify, one bus stop is set in the middle of each edge that bus routes pass through. And at each bus stop in these routes, the bus is set to stop for 15 seconds to imitate the behavior of a bus picking up and dropping off passengers.

4.1.4. Demand setups

As there are two kinds of traffic modes, passenger cars and buses, considered in this research, the demands of both are set up.

As for the demands of vehicle passengers, *randomTrips* of SUMO is used to generate a set of random trips starting from and ending towards the marginal edges of the road network for passenger cars. For Case 1, the demand for passenger cars is set to be 1500 veh/h. For Case 2, the demand for passenger cars is set to be 2500 veh/h. For Case 3, the demand for passenger cars is set to be 3000 veh/h. The values of these demands are set to cause bottlenecks in each case and avoid the complete blockage of the road network. And the generated random demands are uneven for each direction in the road network.

As for the demands for buses, 6 buses per hour are set for each bus route for all 3 cases.

4.2. DRL algorithm implementation

For DRL algorithm implementation, there are some kinds of available sources to help realize the application in multi-modal simulation of SUMO: (1) an open source repository *SUMO-RL* in GitHub to provide an interface between RL and SUMO; (2) an open source repository *sumolights* in GitHub to provide the function of reading road network information of SUMO and obtaining the controlled lanes of each phase for a traffic light based on the paper of Genders and Razavi (Genders & Razavi, 2019); (3) a Python package *stable_baselines3* providing PPO, A2C, and DQN algorithms (Raffin et al., 2021); (4) an open source repository *RLTrafficManager* providing examples of applying multi-agent PPO, A2C, and DQN algorithms from stable_baselines3 (YanivHacker, 2022). This research takes advantage of these codes and adapts them for the implementation of different DRL algorithms in multi-modal simulation of SUMO.

The implementation process of the single-agent DRL algorithm is shown in Figure 4.8.

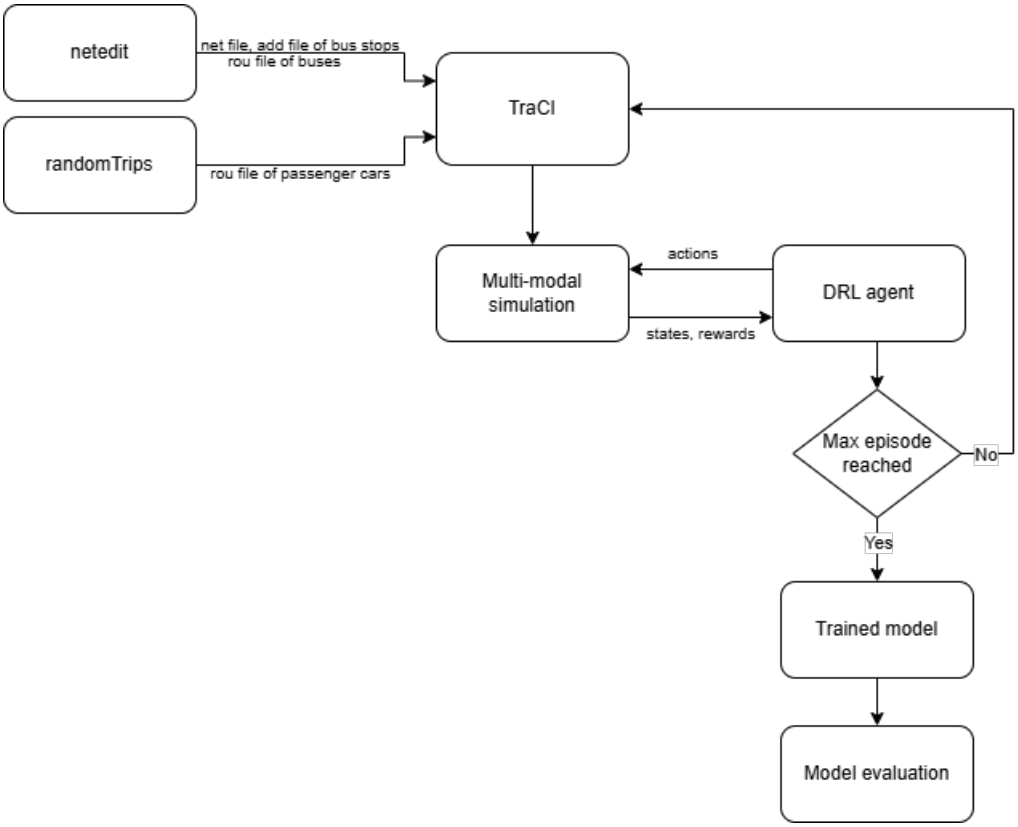


Figure 4.8: Flow chart of the single-agent DRL algorithm implementation process

netedit and *randomTrips* are first used to create the network file, additional file of bus stops, route file of passenger cars, and route files of buses respectively. Then, these files are used by *TraCI* for initializing and running the simulation. During the simulation process, the agent in the DRL algorithm assigns the actions for the corresponding traffic lights, and then the simulation continues by conducting the actions and feeding the states and rewards back to the agent in the DRL algorithm. This process is repeated until the maximum number of training episodes is reached, and the trained model is obtained. After the training process of the DRL algorithm is finished, the obtained trained model is evaluated by using two new multi-modal simulations set up by adding 10% random demand and deleting 10% random demand for passenger cars to the original multi-modal simulation.

The implementation process of the multi-agent DRL algorithm is shown in Figure 4.9.

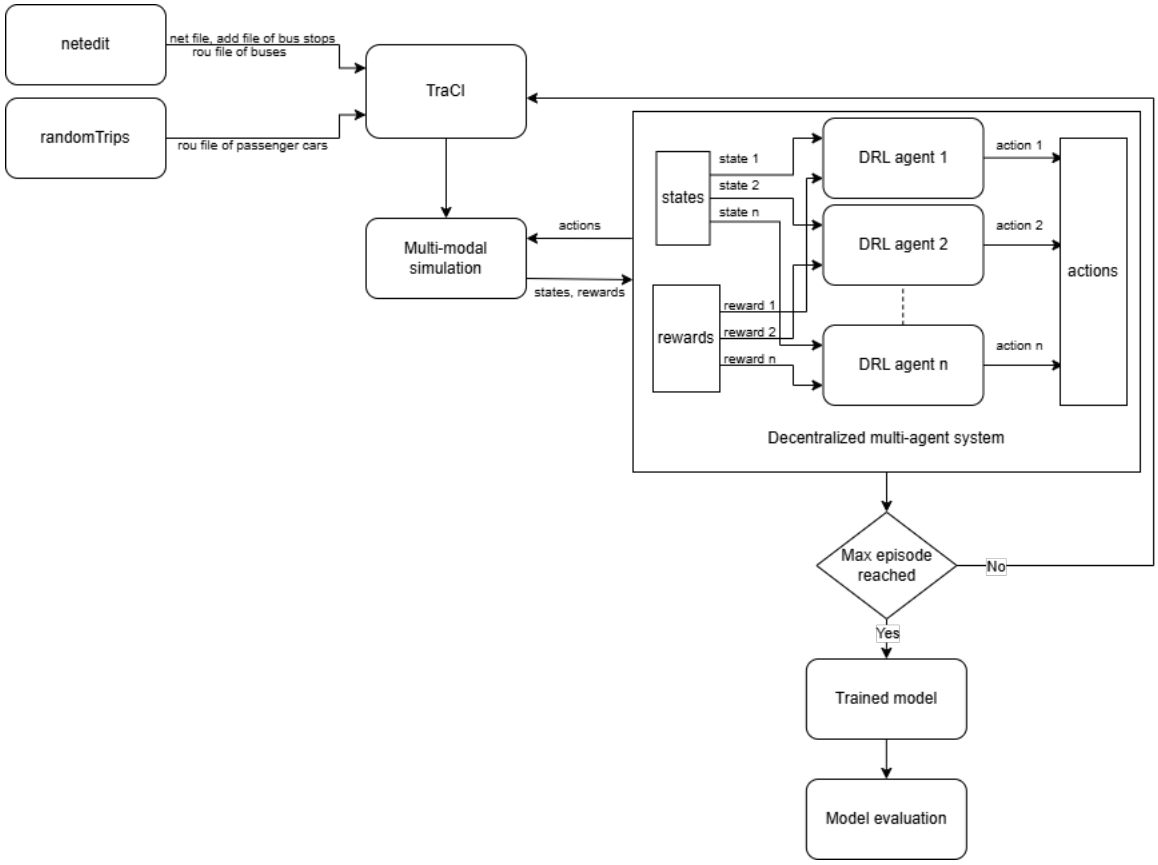


Figure 4.9: Flow chart of the multi-agent DRL algorithm implementation process

The implementation process of multi-agent DRL algorithm is roughly the same as that of single-agent DRL, but the implementation of multi-agent DRL algorithm utilizes a multi-agent system containing multiple DRL agents to interact with the multi-modal simulation instead of making the DRL agent to interact with multi-modal simulation directly in the implementation process of single-agent DRL algorithm. The system transfers the states and rewards of each traffic light to each corresponding agent and collects the actions of each agent, which ensures that all the agents can obtain their own observation and reward information and execute their actions in a certain order at the same time step to avoid confusion.

In the implementation process of multi-agent DRL algorithm, all the agents interact through a shared environment, which is the multi-modal simulation. The actions taken by each agent affect the states of the shared environment, which in turn influences the decisions of each agent.

For Case 1, only the two single-agent DRL algorithms (PPO, A2C) are implemented due to the fact that there is only one intersection with one traffic light in the road network. For Case 2 and 3, both the two single-agent DRL algorithms (PPO, A2C) and the three multi-agent DRL algorithms (MAPPO, MAA2C, MADQN) are implemented.

4.2.1. PPO implementation

The parameter settings of PPO and MAPPO for training are shown in Table 4.1, 4.2 and 4.3 respectively.

Table 4.1: PPO training hyperparameters

Parameters	Value	Parameters	Value
Policy	MlpPolicy	N episodes	10
Learning rate	0.0001	Gae lambda	0.95
N steps	128	Clip range	0.2
Batch size	256	Vf coef	0.5
γ	0.99	Max grad norm	0.5

Table 4.2: MAPPO training hyperparameters of Case 2

Parameters	Value	Parameters	Value
Policy	MlpPolicy	N episodes	20
Learning rate	0.001	Gae lambda	0.95
N steps	128	Clip range	0.2
Batch size	64	Vf coef	0.5
γ	0.99	Max grad norm	0.5

Table 4.3: MAPPO training hyperparameters of Case 3

Parameters	Value	Parameters	Value
Policy	MlpPolicy	N episodes	20
Learning rate	0.0005	Gae lambda	0.95
N steps	128	Clip range	0.2
Batch size	64	Vf coef	0.5
γ	0.99	Max grad norm	0.5

4.2.2. A2C implementation

The parameter settings of A2C and MAA2C for training are shown in Table 4.4, 4.5 and 4.6 respectively.

Table 4.4: A2C training hyperparameters

Parameters	Value	Parameters	Value
Policy	MlpPolicy	Gae lambda	1.0
Learning rate	0.0001	Vf coef	0.5
N steps	5	Max grad norm	0.5
γ	0.99	Rms prop eps	0.00001

Table 4.5: MAA2C training hyperparameters of Case 2

Parameters	Value	Parameters	Value
Policy	MlpPolicy	Gae lambda	1.0
Learning rate	0.001	Vf coef	0.5
N steps	5	Max grad norm	0.5
γ	0.99	Rms prop eps	0.00001

Table 4.6: MAA2C training hyperparameters of Case 3

Parameters	Value	Parameters	Value
Policy	MlpPolicy	Gae lambda	1.0
Learning rate	0.0005	Vf coef	0.5
N steps	128	Max grad norm	0.5
γ	0.99	Rms prop eps	0.00001

4.2.3. DQN implementation

The parameter settings of MADQN for training are shown in Table 4.7 and 4.8 respectively.

Table 4.7: MADQN training hyperparameters of Case 2

Parameters	Value	Parameters	Value
Policy	MlpPolicy	Gradient steps	1
Learning rate	0.001	τ	1.0
Buffer Size	1000000	Target update interval	10000
Learning starts	100	Exploration fraction	0.1
Batch size	32	Exploration initial eps	1.0
γ	0.99	Exploration final eps	0.05
Train freq	4	Max grad norm	10

Table 4.8: MADQN training hyperparameters of Case 3

Parameters	Value	Parameters	Value
Policy	MlpPolicy	Gradient steps	1
Learning rate	0.0005	τ	1.0
Buffer Size	1000000	Target update interval	10000
Learning starts	100	Exploration fraction	0.1
Batch size	32	Exploration initial eps	1.0
γ	0.99	Exploration final eps	0.05
Train freq	4	Max grad norm	10

In this research, the majority of hyperparameters use the default value in the python package *stable_baselines3*, while the learning rate and number of steps per update vary a little bit.

With more agents involved in the simulation, the interactions and dependencies between them become more complex and the amount of noise in the learning process also increases. A lower learning rate is used

as the number of agents increases to stabilize the learning process, improve coordination, and reduce noise. And a larger number of steps per update is also needed with the increase in the number of agents to capture the intersections between agents more accurately, better understand the impact of the actions of one agent on other agents, and have a stable and reliable learning.

In the setup of using single agent to control multiple traffic lights, the agent has to manage a much larger state and action space because it needs to consider the status of all traffic lights and their interactions, which requires more careful and gradual updates. Therefore, the single-agent DRL in this research needs a lower learning rate compared to decentralized multi-agent DRL to manage the larger state and action spaces, coordinate actions, and filter out noise for stable updates.

4.3. Fixed traffic light control implementation

In the case study, the implementation of fixed traffic light control is set to shift to the next green phase every 15 seconds, and the next green phase is determined by a fixed green phase sequence schedule, which is the sequence as shown in Figure 4.4.

4.4. Max-pressure traffic light control implementation

As the max-pressure traffic light control is very complicated and difficult to program and implement directly, many studies use a variant of max-pressure for research instead. A variant of max-pressure used by some research (Genders & Razavi, 2019; Wei et al., 2019) is implemented in this case study. In the case study, the implementation of max-pressure traffic light control is set to choose the next green phase every 15 seconds to guarantee the release of the traffic pressure in time and avoid wasting traffic efficiency due to too frequent green phase changes, and the next green phase is determined by max pressure green phase, which is the green phase having the most difference between the total weighted incoming traffic volume and the total weighted outgoing traffic volume.

4.5. Computation setups

In this research, the computation of the learning process is fulfilled with one CPU by Delftblue, which is the supercomputer of Delft University of Technology (DHPC, 2024). The environment variable $SUMO\_HOME$ is set to the path of the Python package $Libsumo$, and the variable $LIBSUMO\_AS\_TRACI$ is declared to be 1 for the purpose of obtaining a large performance boost (Alegre, 2019).

4.6. Experimental evaluation

The experimental evaluation compares the results of fixed traffic light control and max-pressure traffic light control with the learning results of different DRL methods in multi-modal simulation of SUMO: (1) PPO; (2) A2C; (3) MAPPO; (4) MAA2C; (5) MADQN.

In the evaluation process, different performance indicators including total reward, total travel time of passenger cars and buses are measured, and different relevant experiment results including training results, model evaluation results, and average training time of an episode for each DRL algorithm are presented. These results can reflect how well each method can be applied to improve traffic flow efficiency and achieve transit priority compared to conventional methods in multi-modal simulation. And Python is used to plot the line plots, box plots, and violin plots of these results to facilitate demonstrating the performance of different control methods.

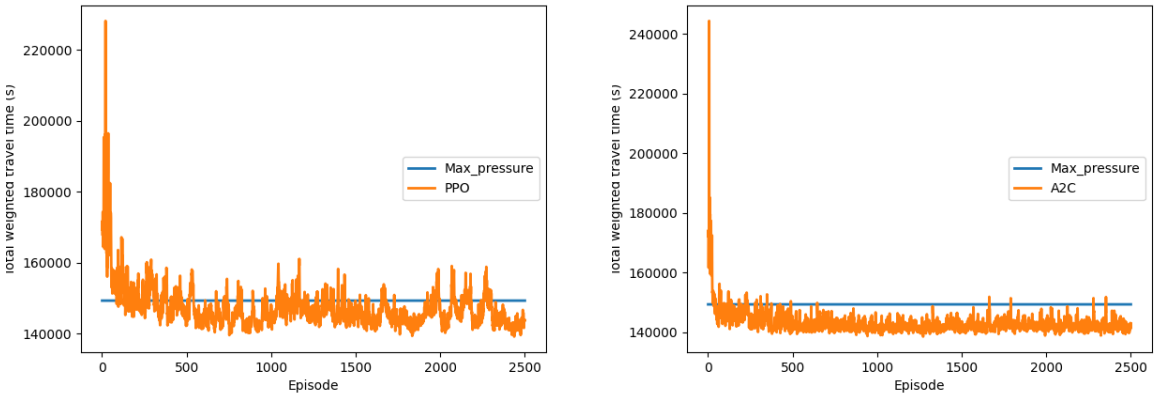
4.7. Results

This section provides a description and discussion of the training results, model evaluation results, and training time results of different DRL algorithms in different cases.

4.7.1. Training results

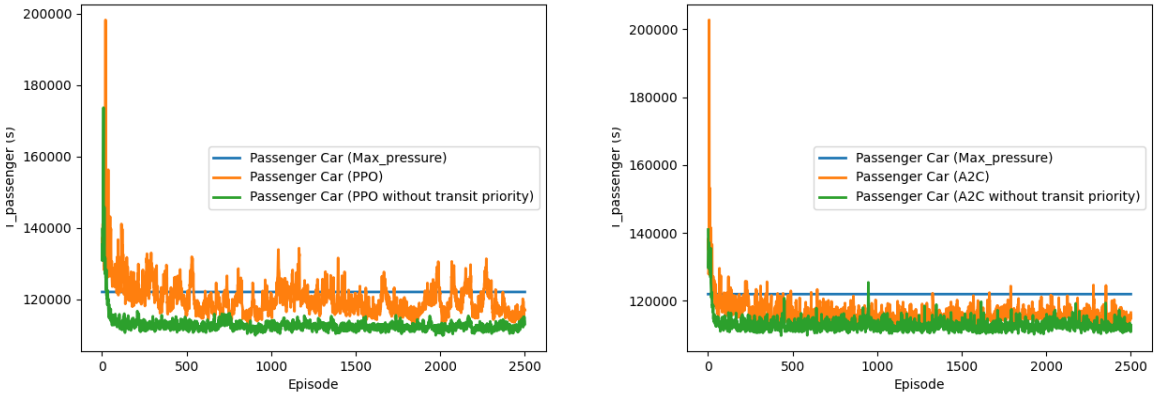
The training results use line plots to show the change, trend, and training performance of different DRL algorithms in different cases.

Training results of Case 1



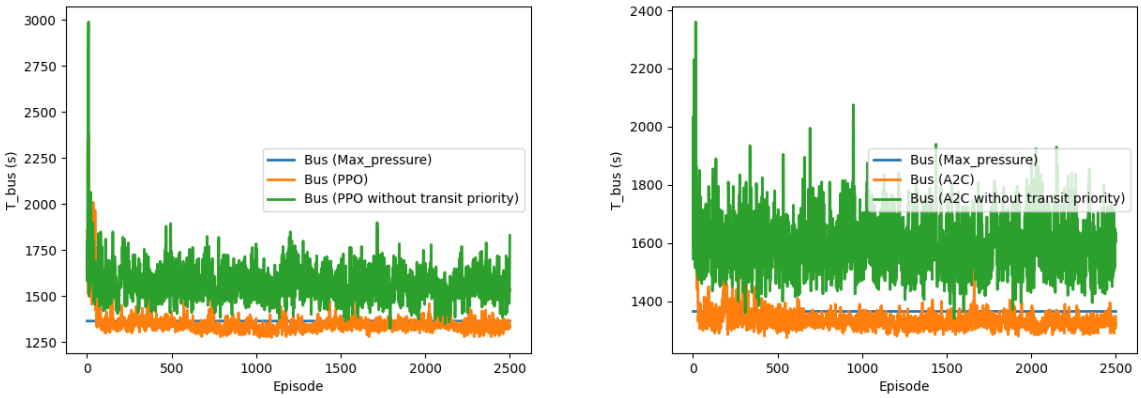
(a) Training process of PPO (b) Training process of A2C

Figure 4.10: Training Process of different DRL algorithms of Case 1



(a) Total travel time of passenger cars for PPO (b) Total travel time of passenger cars for A2C

Figure 4.11: Total travel time of passenger cars of different algorithms of Case 1



(a) Total travel time of buses for PPO

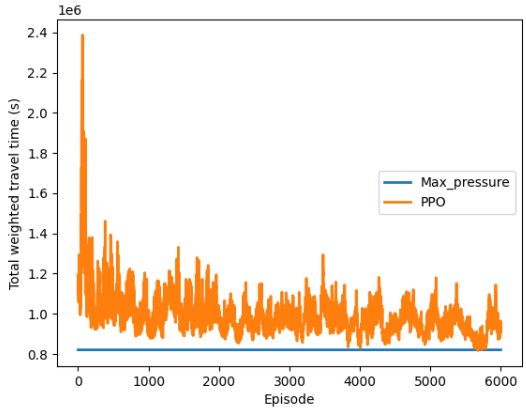
(b) Total travel time of buses for A2C

Figure 4.12: Total travel time of buses of different algorithms of Case 1

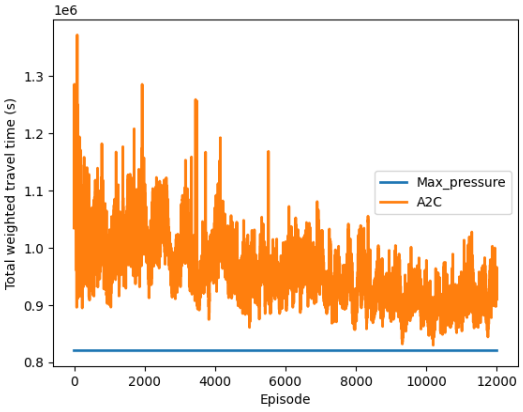
In Case 1, the total weighted travel time, the total travel time of passenger cars, and the total travel time of buses of fixed traffic light control are 712355 s, 445855 s, and 13325 s respectively.

As can be seen from Figure 4.10, 4.11, and 4.12, both PPO and A2C perform slightly better than max-pressure after the convergence of the training process, and A2C shows smaller fluctuations than PPO. Compared to the deep reinforcement learning methods without considering transit priority, the deep reinforcement learning methods considering transit priority have a higher total travel time of passenger cars but a lower total travel time of buses, which proves the achievement of transit priority in Case 1.

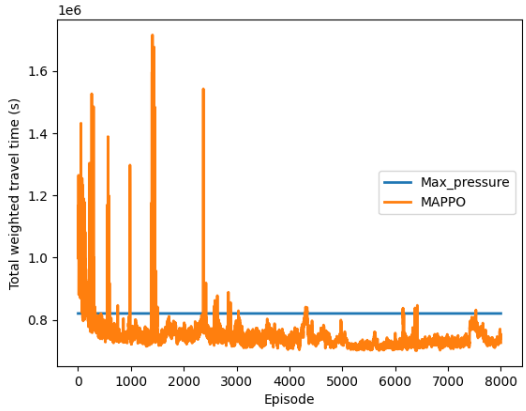
Training results of Case 2



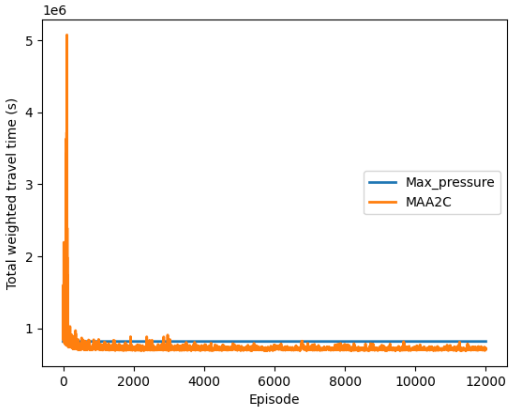
(a) Training process of PPO



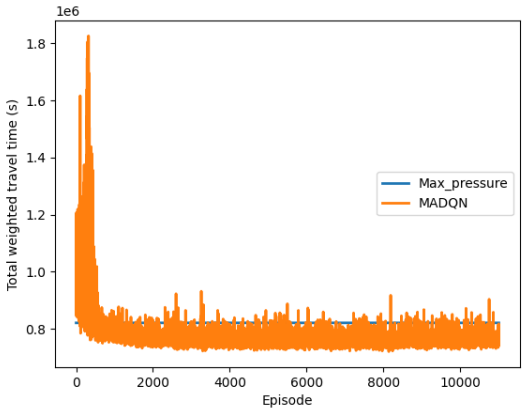
(b) Training process of A2C



(c) Training process of MAPPO

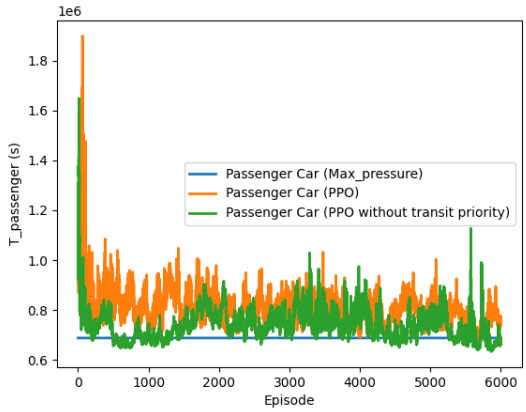


(d) Training process of MAA2C

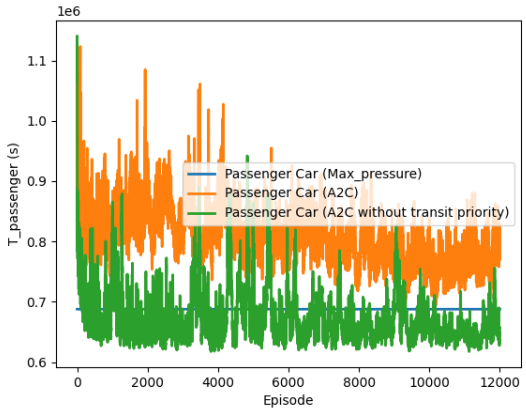


(e) Training process of MADQN

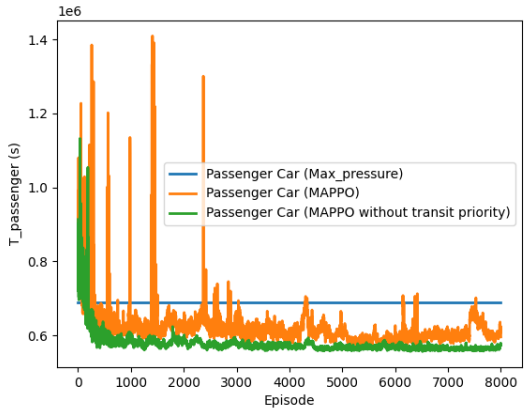
Figure 4.13: Training Process of different DRL algorithms of Case 2



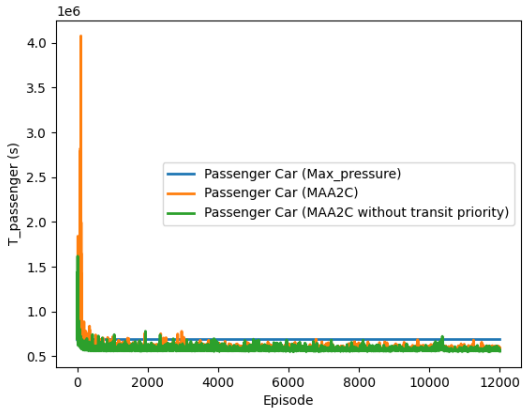
(a) Total travel time of passenger cars for PPO



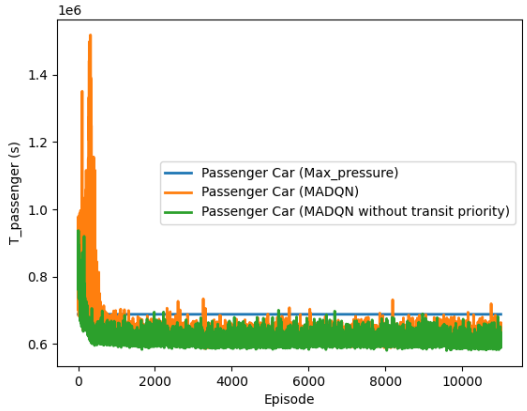
(b) Total travel time of passenger cars for A2C



(c) Total travel time of passenger cars for MAPPO

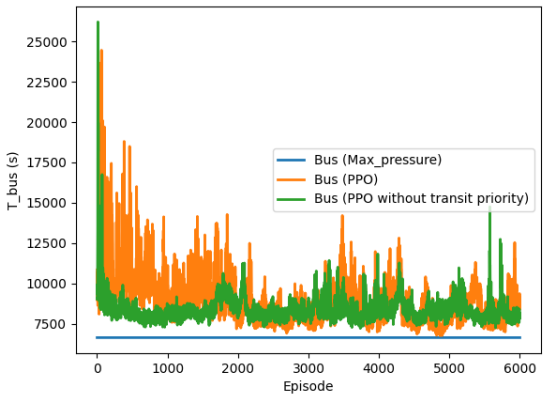


(d) Total travel time of passenger cars for MAA2C

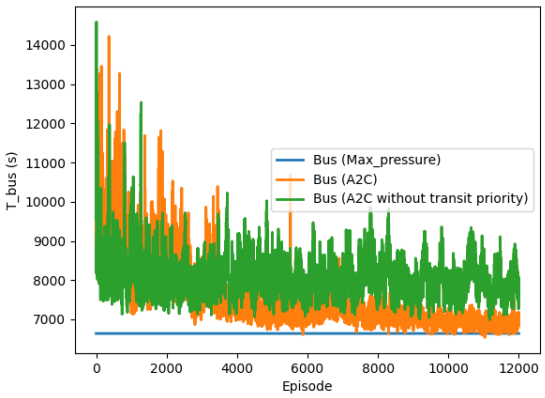


(e) Total travel time of passenger cars for MADQN

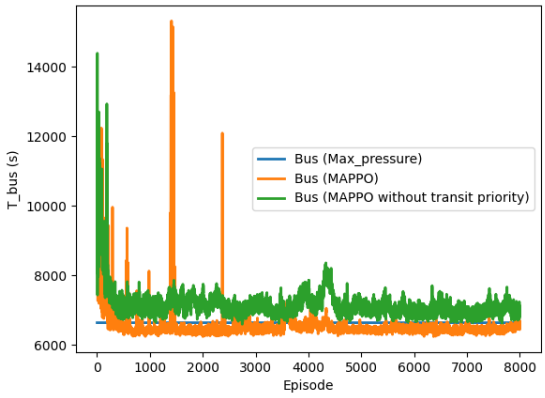
Figure 4.14: Total travel time of passenger cars of different algorithms of Case 2



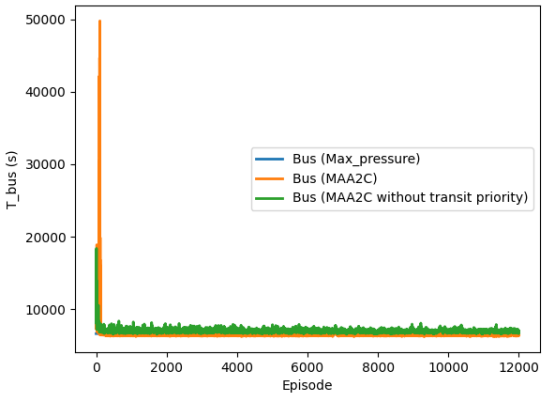
(a) Total travel time of buses for PPO



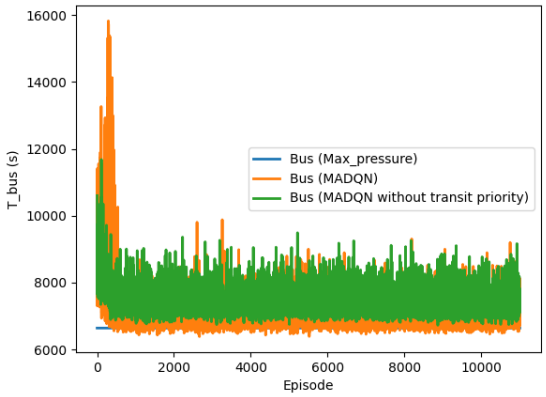
(b) Total travel time of buses for A2C



(c) Total travel time of buses for MAPPO



(d) Total travel time of buses for MAA2C



(e) Total travel time of buses for MADQN

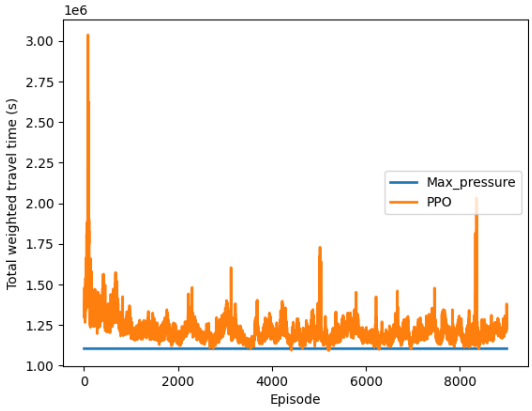
Figure 4.15: Total travel time of buses of different algorithms of Case 2

In Case 2, the total weighted travel time, the total travel time of passenger cars, and the total travel time of buses of fixed traffic light control are 2514635 s, 1951935 s, and 28135 s respectively.

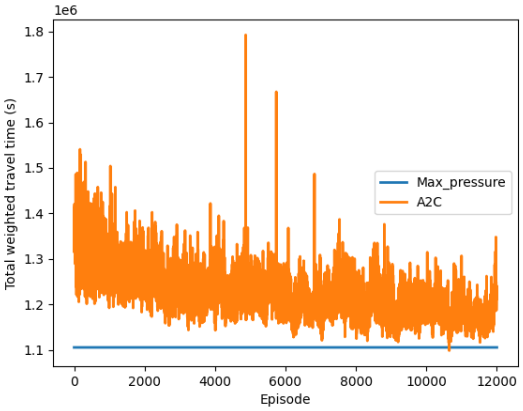
As can be seen from Figure 4.13, 4.14, and 4.15, the single-agent deep reinforcement learning methods PPO and A2C become more volatile and perform worse than max-pressure in Case 2, while

multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MAPPO perform better than max-pressure and demonstrate smaller fluctuations than single-agent deep reinforcement learning methods after convergence. Among the three multi-agent deep reinforcement learning methods, MAA2C performs best after convergence, MAPPO needs more episodes to converge, and MADQN shows the largest fluctuations after convergence. Compared to the deep reinforcement learning methods without considering transit priority, the deep reinforcement learning methods considering transit priority have a higher total travel time of passenger cars but a lower total travel time of buses, which proves the achievement of transit priority in Case 2.

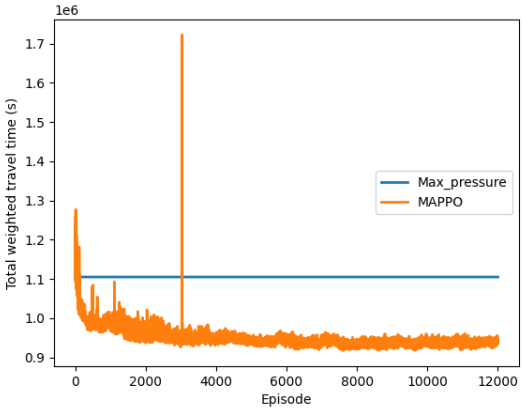
Training results of Case 3



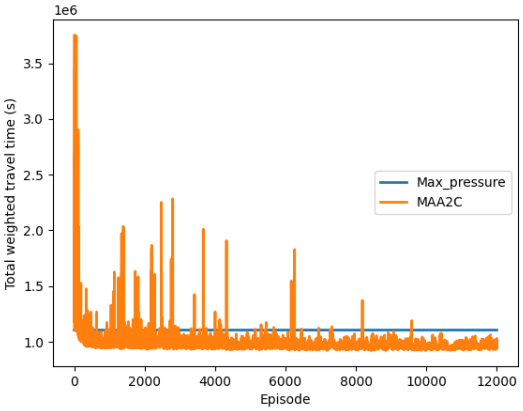
(a) Training process of PPO



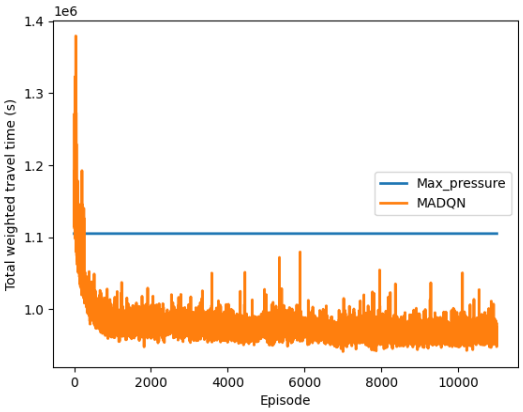
(b) Training process of A2C



(c) Training process of MAPPO

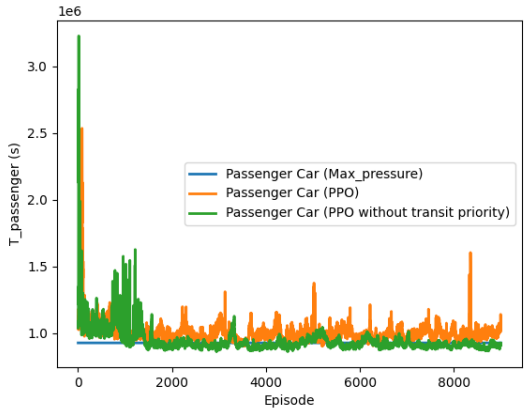


(d) Training process of MAA2C

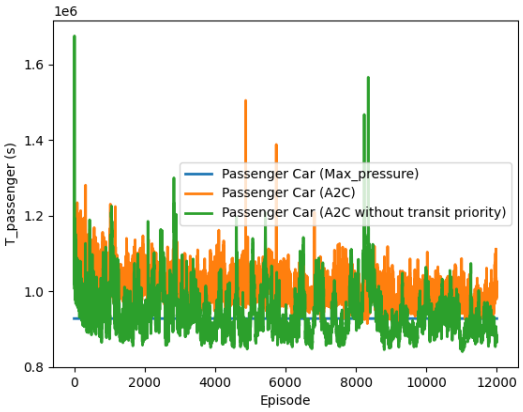


(e) Training process of MADQN

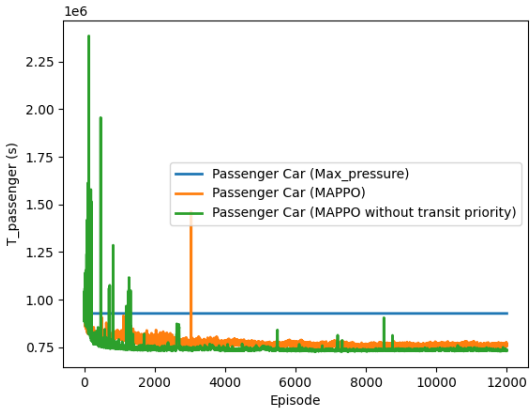
Figure 4.16: Training Process of different DRL algorithms of Case 3



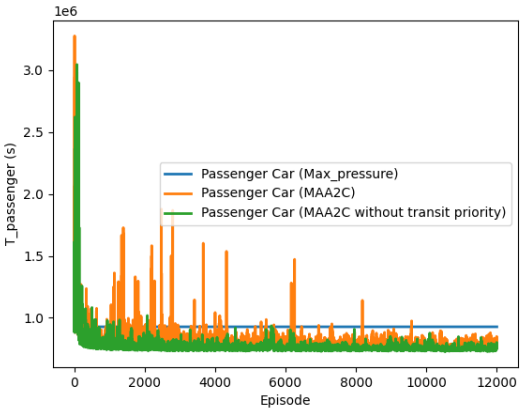
(a) Total travel time of passenger cars for PPO



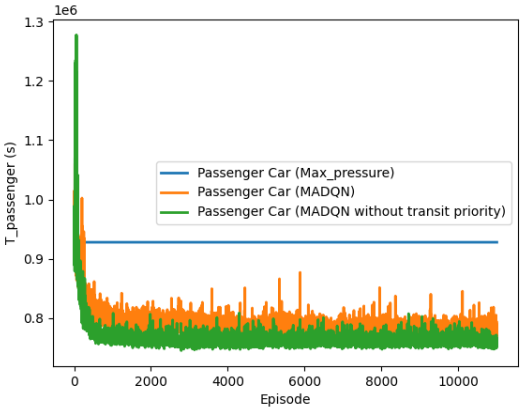
(b) Total travel time of passenger cars for A2C



(c) Total travel time of passenger cars for MAPPO

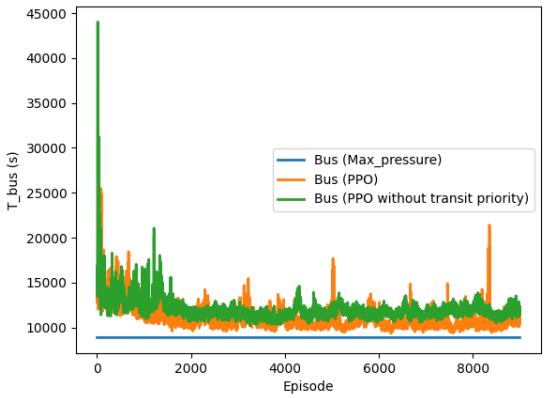


(d) Total travel time of passenger cars for MAA2C

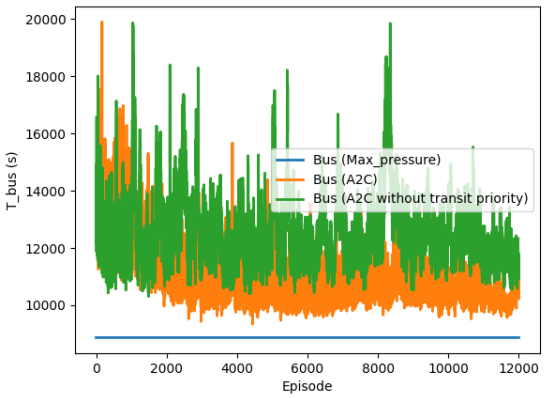


(e) Total travel time of passenger cars for MADQN

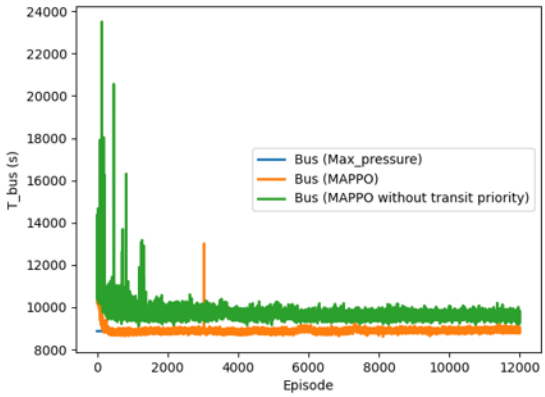
Figure 4.17: Total travel time of passenger cars of different algorithms of Case 3



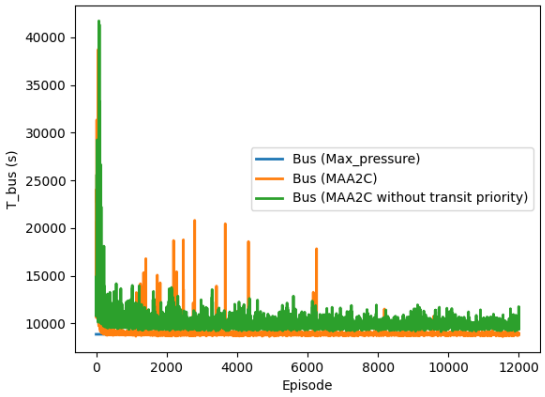
(a) Total travel time of buses for PPO



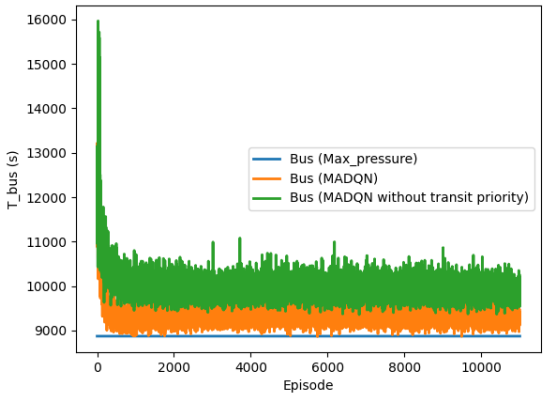
(b) Total travel time of buses for A2C



(c) Total travel time of buses for MAPPO



(d) Total travel time of buses for MAA2C



(e) Total travel time of buses for MADQN

Figure 4.18: Total travel time of buses of different algorithms of Case 3

In Case 3, the total weighted travel time, the total travel time of passenger cars, and the total travel time of buses of fixed traffic light control are 3126960 s, 2533160 s, and 29690 s respectively.

As can be seen from Figure 4.16, 4.17, and 4.18, the single-agent deep reinforcement learning methods PPO and A2C become more volatile and perform worse than max-pressure in 3, while multi-

agent deep reinforcement learning methods MAPPO, MAA2C, and MAPPO perform better than max-pressure and demonstrate smaller fluctuations than single-agent deep reinforcement learning methods after convergence. Among the three multi-agent deep reinforcement learning methods, MAPPO performs best after convergence, MAA2C needs more episodes to converge, and MADQN shows the largest fluctuations after convergence. Compared to the deep reinforcement learning methods without considering transit priority, the deep reinforcement learning methods considering transit priority have a higher total travel time of passenger cars but a lower total travel time of buses, which proves the achievement of transit priority in Case 3.

4.7.2. Model evaluation results

In this research, the model evaluation experiments are conducted under two scenarios with 10% less demand of passenger cars and 10% more demand of passenger cars. In each scenario, the evaluation experiments are repeated for 10 times. The model evaluation results use not only tables to demonstrate the mean and standard error of the three indicators of DRL performance for different DRL algorithms in different cases, but also box plots and violin plots to show how the evaluation results of different DRL algorithms in different cases are spread out with outliers identification and compare the distribution of evaluation results between different DRL algorithms respectively as shown in Appendix B, thus explaining the values of the standard error in model evaluation results.

Model evaluation results of Case 1

The statistics of model evaluation results of Case 1 are shown in Table 4.13 and 4.14.

Table 4.9: Statistics of model evaluation results with 10% less demand of Case 1

Method	$\overline{T}_{total}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	665095.0	-	398395.0	-	13335.0	-
Max-pressure	135125.0	-	108425.0	-	1335.0	-
PPO	131690.5	633.9	105290.5	477.1	1320.0	13.6
A2C	129384.0	442.5	103164.0	417.8	1311.0	6.6

Table 4.10: Statistics of model evaluation results with 10% more demand of Case 1

Method	$\overline{T}_{total}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	753430.0	-	486830.0	-	13330.0	-
Max-pressure	158260.0	-	131260.0	-	1350.0	-
PPO	157036.0	843.2	129716.0	950.7	1366.0	26.9
A2C	154631.0	694.3	127561.0	713.9	1353.5	25.3

As can be seen from Table 4.9 and 4.10, both PPO and A2C perform better than max-pressure with a small standard error in terms of total weighted travel time and total travel time of passenger cars under the scenarios with 10% less demand and 10% more demand. As for total travel time of buses, PPO and A2C perform similarly to max-pressure. As for standard errors, A2C has lower standard errors for evaluation results than PPO, which means A2C has lower variance than PPO. Overall, A2C performs best with lower total travel time of passenger cars, similar total travel time of buses to max-pressure, and lower variance among all the deep reinforcement learning methods in Case 1.

Model evaluation results of Case 2

Table 4.11: Statistics of model evaluation results with 10% less demand of Case 2

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	2296885.0	-	1734685.0	-	28110.0	-
Max-pressure	736890.0	-	606690.0	-	6510.0	-
PPO	819064.0	20272.4	655684.0	15746.6	8169.0	316.2
A2C	816489.0	6043.8	679729.0	5683.3	6838.0	101.0
MAPPO	663550.5	4359.4	533970.5	3236.2	6479.0	67.9
MAA2C	641319.0	2794.6	514089.0	3166.9	6361.5	35.4
MADQN	675208.5	6119.8	540848.5	5961.2	6718.0	88.7

Table 4.12: Statistics of model evaluation results with 10% more demand of Case 2

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	2721580.0	-	2158680.0	-	28145.0	-
Max-pressure	886675.0	-	757775.0	-	6445.0	-
PPO	1078274.5	32353.1	883344.5	22847.8	9746.5	546.1
A2C	1047563.0	16627.0	905913.0	16078.0	7082.5	107.8
MAPPO	805418.0	7477.1	672558.0	7118.7	6643.0	50.3
MAA2C	780059.0	4990.4	650749.0	4933.8	6465.5	57.6
MADQN	851752.5	27663.1	704052.5	18478.0	7385.0	496.0

As can be seen from Table 4.11 and 4.12, both the single-agent deep reinforcement learning methods PPO and A2C perform worse than max-pressure, but still much better than fixed traffic control. The multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MADQN perform better than max-pressure in terms of total weighted travel time and total travel time of passenger cars and they all have a similar total travel time of buses to max-pressure. MAPPO and MADQN perform worse than MAA2C in terms of total weighted travel time in both scenarios and have quite large standard errors for evaluation results which means that the performance of MAPPO and MADQN have large variance. Overall, MAA2C performs best with the lowest travel time of each mode and variance among all the deep reinforcement learning methods in Case 2.

Model evaluation results of Case 3

The statistics of model evaluation results of Case 3 are shown in Table 4.13 and 4.14.

Table 4.13: Statistics of model evaluation results with 10% less demand of Case 3

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	2836885	-	2245185	-	29585	-
Max-pressure	1007545	-	831145	-	8820	-
PPO	1125164.0	14061.4	917664.0	13030.7	10375.0	175.5
A2C	1116447.5	20277.9	912877.5	18981.8	10178.5	203.3
MAPPO	857570.5	3508.6	680670.5	2737.3	8845.0	77.4
MAA2C	869156.5	9406.5	693766.5	9277.3	8769.5	45.7
MADQN	879149.0	7454.0	695799.0	5926.4	9167.5	147.4

Table 4.14: Statistics of model evaluation results with 10% more demand of Case 3

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	3396745.0	-	2804445.0	-	29615.0	-
Max-pressure	1214005.0	-	1035305.0	-	8935.0	-
PPO	1374221.5	25523.1	1156591.5	22259.8	10881.5	251.3
A2C	1379712.0	40784.1	1168222.0	39914.3	10574.5	390.1
MAPPO	1026981.5	4987.7	847811.5	4829.3	8958.5	63.1
MAA2C	1034160.0	12214.5	855570.0	12007.7	8929.5	59.3
MADQN	1064534.5	10186.8	876634.5	8238.0	9395.0	136.5

As can be seen from Table 4.13 and 4.14, both the single-agent deep reinforcement learning methods PPO and A2C perform worse than max-pressure, but still much better than fixed traffic control. The multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MADQN perform much better than max-pressure in terms of total weighted travel time and total travel time of passenger cars and they all have a similar total travel time of buses to max-pressure. MAA2C and MADQN perform worse than MAPPO in terms of total weighted travel time in both scenarios and have quite large standard errors for evaluation results which means that the performance of MAA2C and MADQN have large variance. Overall, MAPPO performs best with the lowest total weighted travel time and variance among all the deep reinforcement learning methods in Case 3.

4.7.3. Training time results

The training time results present the average training time of different DRL algorithms in different cases under different scenarios.

Table 4.15: Average training time of Case 1

Method	Average training time of an episode (s)
PPO	1.96
A2C	1.99

Table 4.16: Average training time of Case 2

Method	Average training time of an episode (s)
PPO	4.87
A2C	5.28
MAPPO	6.14
MAA2C	6.38
MADQN	5.76

Table 4.17: Average training time of Case 3

Method	Average training time of an episode (s)
PPO	14.26
A2C	9.62
MAPPO	8.88
MAA2C	9.81
MADQN	8.27

As can be seen from the above three tables, single-agent deep reinforcement learning methods PPO and A2C have less average training time than multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MADQN in Case 2 while single-agent deep reinforcement learning methods PPO and A2C have more average training time than multi-agent deep reinforcement learning methods MAA2C and MADQN, which shows that single-agent deep reinforcement learning methods become less time-efficient with the increase in the number of actions to be determined. As for the policy-based multi-agent deep reinforcement learning methods, MAPPO need less average training time than MAA2C in both Case 2 and 3. As for the value-based multi-agent deep reinforcement learning method MADQN, it needs less time for training than the policy-based multi-agent deep reinforcement learning methods MAPPO, MAA2C in both Case 2 and 3.

4.7.4. Discussion

By reviewing all the results, it can be seen that A2C, MAA2C, MAPPO are the deep reinforcement learning methods with the best training performance, evaluation performance, lowest variance, and fast training time in the situations of single traffic light, 2*2 traffic lights, and 3*2 traffic lights respectively. A2C achieves the best traffic light controls on total travel time of each traffic mode in Case 1. Even though MAA2C and MAPPO perform slightly worse than max-pressure in terms of total travel time of max-pressure in the evaluation with 10% more demand of passenger cars in Case 2 and 3 respectively, they have a much better performance in total weighted travel time and total travel time of passenger cars, which proves their best general performance under the premise of considering both traffic efficiency and transit priority.

Both single-agent deep reinforcement methods PPO and A2C perform well in the training and evaluation with fast average training time in the situation of single traffic light while A2C performs slightly better than PPO. The reason can be that the clipping mechanism of PPO to prevent large updates might inadvertently cause it to overfit to suboptimal policies in simple environments while less constrained updates of A2C allow it to explore and generalize better in straightforward tasks. Therefore, both PPO and A2C are recommended for traffic light control optimization in the road network of single traffic light in multi-modal simulation.

Even though single-agent deep reinforcement learning methods PPO and A2C perform well in the situation of single traffic light, they struggle to achieve a good performance in the situations of multiple traffic lights. This is due to the fact that the single-agent DRL integrates the states of each traffic light to form a large state set and the formed large state set can not directly show what states belong to which traffic light, which makes the training process more difficult with the increase in the number of traffic lights. Therefore, it is recommended to use multi-agent deep reinforcement learning methods instead of single-agent deep reinforcement learning methods in the situation of multiple traffic lights in multi-modal simulation to avoid too large size of state set and help the better learning process for DRL algorithms.

As for value-based deep reinforcement learning method MADQN, it performs worse than policy-based deep reinforcement learning methods MAPPO and MAA2C in both Case 2 and 3. The reason for this is probably that policy-based deep reinforcement learning methods MAPPO and MAA2C directly learn a policy, which is a probability distribution over actions, and generate actions from it, allowing them to explore different strategies and action combinations more effectively for better discovery of optimal policies in environments where multiple agents interact while the value-based deep reinforcement learning method MADQN focuses on estimating the Q-value function for each possible action and optimizing it, which can lead to overfitting to specific action values, potentially limiting exploration. Therefore, it is advisable to use policy-based deep reinforcement learning methods for traffic light control optimization in multi-modal simulation.

As for policy-based deep reinforcement learning methods, MAA2C performs better than MAPPO in the road network of small sizes but performs worse in the road network of large sizes. This is probably due to the reason that the advantage-based updates in MAA2C can lead to more aggressive policy changes, which might be beneficial in simpler, less crowded environments. However, in more complex environments with more agents, this aggressive updating can lead to poor performance due to over-exploitation, which is adverse to the coordination between agents. The conservative updates of MAPPO based on the clipping mechanism encourage more consistent exploration across episodes. This becomes an advantage in scenarios with more agents, where the environment is more complex, and the need for consistent policy updates is crucial to maintain performance. Therefore, it is recommended to use MAA2C in road networks

of multiple traffic lights with small sizes and use MAPPO in road networks with large sizes for more optimized traffic light control.

5

Conclusions

5.1. Conclusions

In conclusion, the research applied both single-agent deep reinforcement learning methods PPO, A2C and multi-agent deep reinforcement learning methods MAPPO, MAA2C, MADQN for traffic light control optimization in multi-modal simulation. Experiments are conducted in 3 different cases, which are set in road networks of different sizes respectively. And all the deep reinforcement methods used are evaluated in terms of training process, model evaluation, and training time.

The results of the research show that A2C, MAA2C, and MAPPO perform best in road networks of single traffic light, 2*2 traffic lights, and 3*2 traffic lights respectively. Both PPO and A2C are recommended for traffic light optimization in the road network of single traffic light in multi-modal simulation. But for road networks of multiple traffic lights, it is recommended to use multi-agent deep reinforcement learning methods instead of single-agent deep reinforcement learning methods in multi-modal simulation to avoid too large size of state set and help the better learning process for DRL algorithms. And it is advisable to use policy-based deep reinforcement learning methods for traffic light control optimization in multi-modal simulation. As for road networks of multiple traffic lights, it is recommended to use MAA2C in road networks of multiple traffic lights of small sizes and use MAPPO in road networks of multiple traffic lights of large sizes for more optimized traffic light control.

The innovations of the research are mainly reflected in the following three points.

1. The research innovatively considers both traffic efficiency and transit priority for traffic light control optimization by using the way of assigning a weight for each traffic mode in the computations of queue length, traffic volume, and travel time for states and rewards in the application of deep reinforcement learning methods for traffic light control optimization in multi-modal simulation, which can guide people to choose more environmentally friendly traffic modes and ultimately achieve the goal of forming a more sustainable transportation system in the urban area.
2. The research explores the use and effectiveness of using different single-agent deep reinforcement learning methods to control multiple traffic lights in multi-modal simulation.
3. The research evaluates different deep reinforcement learning methods in terms of different aspects including training curves, performance of each traffic mode, training time, performance of model evaluation, and performance in road networks of different sizes under the topic of the application of different deep reinforcement learning methods for traffic light control optimization in multi-modal simulation.

In summary, this research discovers the use and effects of different deep reinforcement learning methods in multi-modal simulation, evaluates their performance in road networks of different sizes from different aspects, and paves the way for developing efficient DRL agents that consider the needs of different road users and prioritize public transport for the application of DRL in multi-modal simulation, which is of great significance for achieving more optimized traffic light control that considers multiple factors and realizing a more efficient and sustainable urban transportation system.

As the research is simplified to some extent, some further measures can be further applied to transform the research into a mature product in the real world.

1. Different types of sensors like loops and cameras should be installed on each edge of the study area to obtain the traffic information including weighted traffic volume and weighted queue length and identify the appearance of buses on the road.
2. Enough historical traffic demand data of the study area can be used as the input for training.
3. The weight assigned for each traffic mode can be determined based on the actual importance of each traffic mode defined by the traffic management sector.

However, there are still some limitations existing in this research as shown below:

1. Only two traffic modes, passenger cars and buses, are considered.
   In modern urban traffic, the participation and interaction of multiple traffic modes have become the norm. Therefore, covering more traffic modes in multimodal simulation is of great significance for the optimal control of traffic lights in complex scenarios.
2. The action is only limited to selecting the green phase without involving the corresponding green phase duration.
   In the current framework, the duration for a green phase is just a multiple of 5 seconds, which is less flexible than also considering the corresponding green phase duration in the action selection directly.
3. Only the travel time is considered as the reward and performance indicator.
   Actually, the control of urban traffic is a large and complicated task considering plenty of factors like efficiency, safety, energy consumption, level of public transport service, sustainability, and so on. A more comprehensive reward design and evaluation can be of great significance to optimizing the DRL-based traffic light control in multi-modal simulation for achieving a balance of multiple goals.
4. The road network used is simpler than the real road network.
   The road network used in this research is simple, which may not adequately reflect the complexity of real road network scenarios.

5.2. Recommendations for future work

Based on the limitations mentioned in Section 5.1, further research regarding the application of DRL algorithms in multi-modal simulation of SUMO can focus on the following several points:

1. **Incorporate more traffic modes.**
   This can enhance the realism and complexity of the simulated environment, better approximating real-world scenarios with diverse traffic compositions, and enable the exploration of complex interactions and dependencies between different traffic modes.
2. **Use both the green phase and the corresponding green phase duration as the action choice.**
   The application of both kinds of actions can achieve more accurate traffic light control. But this has to be considered carefully as this could increase the difficulty for the learning process of DRL algorithms.
3. **Consider multiple factors in reward design.**
   A more comprehensive reward function allows for promoting the system performance from a more holistic perspective.
4. **Consider multiple factors in model evaluation.**
   Evaluating trained models based on a variety of performance metrics provides a more robust assessment of their capabilities, identifying potential trade-offs and enabling the selection of agents that excel in specific operational conditions or prioritize particular performance criteria.
5. **Use a part of a real road network as the case study.**
   This provides a more realistic and complex environment compared to synthetic networks, allowing for the validation of algorithm performance under realistic traffic conditions. This also enables the direct comparison of DRL-based policies with existing traffic management systems, providing insights into the potential benefits and challenges of implementing such algorithms in real-world applications.

References

He, Q., Head, K., & Ding, J. (2014). Multi-modal traffic signal control with priority, signal actuation and coordination. *Transportation Research Part C: Emerging Technologies*, *46*, 65–82. https://doi.org/10.1016/j.trc.2014.05.001

Zhang, B. Y., Yue, H., & Wang, S. (2014). Experience and reference of implementing public transport priority strategy dominated by urban rail transit in tokyo. *Advances in Transportation*, *505*, 813–819. https://doi.org/10.4028/www.scientific.net/AMM.505-506.813

Shinde, S. M. (2017). Adaptive traffic light control system. *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 300–306. https://doi.org/10.1109/ICISIM.2017.8122189

Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, *36*, 177–195. https://doi.org/10.1016/j.trc.2013.08.014

Fouladvand, M. E., Sadjadi, Z., & Shaebani, M. R. (2004). Optimized traffic flow at a single intersection: Traffic responsive signalization. *Journal of Physics A: Mathematical and General*, *37*(3), 561. https://doi.org/10.1088/0305-4470/37/3/002

Darroch, J., Newell, G. F., & Morris, R. (1964). Queues for a vehicle-actuated traffic light. *Operations Research*, *12*(6), 882–895. https://doi.org/10.1287/opre.12.6.882

Shi, T., Devailly, F.-X., Larocque, D., & Charlin, L. (2023). Improving the generalizability and robustness of large-scale traffic signal control. https://doi.org/10.48550/arXiv.2306.01925

Huang, X., Wu, D., & Boulet, B. (2023). Traffic signal control using lightweight transformers: An offline-to-online rl approach. https://doi.org/10.48550/arXiv.2312.07795

Wang, S., & Wang, S. (2023). A novel multi-agent deep rl approach for traffic signal control. https://doi.org/10.48550/arXiv.2306.02684

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237–285. https://doi.org/10.1613/jair.301

Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning: An introduction. *Robotica*, *17*(2), 229–235. https://doi.org/10.1017/S0263574799271172

Feng, G., & Zhong, H. (2023). Rethinking model-based, policy-based, and value-based reinforcement learning via the lens of representation complexity.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T. P., Simonyan, K., & Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, *abs/1712.01815*. https://doi.org/10.48550/arXiv.1712.01815

Ha, D., & Schmidhuber, J. (2018). World models. *CoRR*, *abs/1803.10122*. https://doi.org/10.48550/arXiv.1803.10122

Weber, T., Racanière, S., Reichert, D. P., Buesing, L., Guez, A., Rezende, D. J., Badia, A. P., Vinyals, O., Heess, N., Li, Y., Pascanu, R., Battaglia, P. W., Silver, D., & Wierstra, D. (2017). Imagination-augmented agents for deep reinforcement learning. *CoRR*, *abs/1707.06203*. https://doi.org/10.48550/arXiv.1707.06203

Nagabandi, A., Kahn, G., Fearing, R. S., & Levine, S. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 7559–7566. https://doi.org/10.1109/ICRA.2018.8463189

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., & Levine, S. (2018). Model-based value estimation for efficient model-free reinforcement learning. *CoRR*, *abs/1803.00101*. https://doi.org/10.48550/arXiv.1803.00101

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*, 279–292. https://doi.org/10.1007/BF00992698

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1). https://doi.org/10.1609/aaai.v30i1.10295

Wang, Z., de Freitas, N., & Lanctot, M. (2015). Dueling network architectures for deep reinforcement learning. *CoRR*, *abs/1511.06581*. http://arxiv.org/abs/1511.06581

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.11796

Zhang, S., Zheng, S., Ke, S., Liu, Z., Jin, W., Yuan, J., Yang, Y., Yang, H., & Wang, Z. (2024). How can LLM guide rl? A value-based approach. *CoRR*, *abs/2402.16181*. https://doi.org/10.48550/ARXIV.2402.16181

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Proceedings of the 12th International Conference on Neural Information Processing Systems*, 1057–1063.

Kakade, S. M. (2001). A natural policy gradient. In T. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems* (Vol. 14). MIT Press. https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *CoRR*, *abs/1602.01783*. http://arxiv.org/abs/1602.01783

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., & Zhokhov, P. (2017). Openai baselines.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, *abs/1707.06347*. http://arxiv.org/abs/1707.06347

Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., Liu, P., & Wang, Z. (2024). Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/tnnls.2023.3236361

Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., Liu, P., & Wang, Z. (2023). Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/TNNLS.2023.3236361

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic traffic simulation using sumo. *The 21st IEEE International Conference on Intelligent Transportation Systems*. https://elib.dlr.de/124092/

Szőke, L., Aradi, S., Bécsi, T., & Gáspár, P. (2020). Driving on highway by using reinforcement learning with cnn and lstm networks. *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*, 121–126. https://doi.org/10.1109/INES49302.2020.9147185

Genders, W., & Razavi, S. N. (2016). Using a deep reinforcement learning agent for traffic signal control. *CoRR*, *abs/1611.01142*. https://doi.org/10.48550/arXiv.1611.01142

Chen, P., Zhu, Z., & Lu, G. (2019). An adaptive control method for arterial signal coordination based on deep reinforcement learning. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 3553–3558. https://doi.org/10.1109/ITSC.2019.8917051

Guo, M., Wang, P., Chan, C.-Y., & Askary, S. (2019). A reinforcement learning approach for intelligent traffic signal control at urban intersections. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 4242–4247. https://doi.org/10.1109/ITSC.2019.8917268

Bouktif, S., Cheniki, A., & Ouni, A. (2021). Traffic signal control using hybrid action space deep reinforcement learning. *Sensors*, *21*(7). https://doi.org/10.3390/s21072302

Wang, B., He, Z., Sheng, J., & Chen, Y. (2022). Deep reinforcement learning for traffic light timing optimization. *Processes*, *10*(11). https://doi.org/10.3390/pr10112458

Yu, J., Laharotte, P.-A., Han, Y., & Leclercq, L. (2023). Decentralized signal control for multi-modal traffic network: A deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, *154*, 104281. https://doi.org/10.1016/j.trc.2023.104281

Shabab, K. R., Ali, S. M., & Zaki, M. H. (2023). Deep reinforcement learning-based short-term traffic signal optimizing using disaggregated vehicle data. *Data science for transportation*, *5*(2), 13. https://doi.org/10.1007/s42421-023-00074-x

Li, J., Chen, T., Zhou, F., Lv, X., & Peng, F. (2022). Research on signal control method of deep reinforcement learning based on value distribution. *Journal of Physics: Conference Series*, *2330*(1), 012019. https://doi.org/10.1088/1742-6596/2330/1/012019

Liang, X., Du, X., Wang, G., & Han, Z. (2018). Deep reinforcement learning for traffic light control in vehicular networks. *CoRR*, *abs/1803.11115*. https://doi.org/10.48550/arXiv.1803.11115

Li, D., Wu, J., Xu, M., Wang, Z., & Hu, K. (2020). Adaptive traffic signal control model on intersections based on deep reinforcement learning. *Journal of Advanced Transportation*, *2020*(1), 6505893. https://doi.org/https://doi.org/10.1155/2020/6505893

Li, Z., Xu, C., & Zhang, G. (2021). A deep reinforcement learning approach for traffic signal control optimization. https://doi.org/10.48550/arXiv.2107.06115

Li, C., Ma, X., Xia, L., Zhao, Q., & Yang, J. (2020). Fairness control of traffic light via deep reinforcement learning. *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 652–658. https://doi.org/10.1109/CASE48305.2020.9216899

Ma, Z., Cui, T., Deng, W., Jiang, F., & Zhang, L. (2021). Adaptive optimization of traffic signal timing via deep reinforcement learning. *Journal of Advanced Transportation*, *2021*, 1–14. https://doi.org/10.1155/2021/6616702

Kumar, N., Mittal, S., Garg, V., & Kumar, N. (2022). Deep reinforcement learning-based traffic light scheduling framework for sdn-enabled smart transportation system. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2411–2421. https://doi.org/10.1109/TITS.2021.3095161

Li, Z., Yu, H., Zhang, G., Dong, S., & Xu, C.-Z. (2021). Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *125*, 103059. https://doi.org/https://doi.org/10.1016/j.trc.2021.103059

Louw, C., Labuschagne, L., & Woodley, T. (2022). A comparison of reinforcement learning agents applied to traffic signal optimisation. *SUMO Conference Proceedings*, *3*, 15–43. https://doi.org/10.52825/scp.v3i.116

Mei, X., Fukushima, N., Yang, B., Wang, Z., Takata, T., Nagasawa, H., & Nakano, K. (2023). Reinforcement learning-based intelligent traffic signal control considering sensing information of railway. *IEEE Sensors Journal*, *23*(24), 31125–31136. https://doi.org/10.1109/JSEN.2023.3327696

Koh, S., Zhou, B., Fang, H., Yang, P., Yang, Z., Yang, Q., Guan, L., & Ji, Z. (2020). Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing*, *96*, 106694. https://doi.org/10.1016/j.asoc.2020.106694

Makantasis, K., Kontorinaki, M., & Nikolos, I. (2020). Deep reinforcement-learning-based driving policy for autonomous road vehicles. *IET Intelligent Transport Systems*, *14*(1), 13–24. https://doi.org/10.1049/iet-its.2019.0249

Zhao, J., Qu, T., & Xu, F. (2020). A deep reinforcement learning approach for autonomous highway driving [3rd IFAC Workshop on Cyber-Physical and Human Systems CPHS 2020]. *IFAC-PapersOnLine*, *53*(5), 542–546. https://doi.org/10.1016/j.ifacol.2021.04.142

Bagwe, G., Yuan, X., Chen, X., & Zhang, L. (2023). Ramrl: Towards robust on-ramp merging via augmented multimodal reinforcement learning. *2023 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*, 23–33. https://doi.org/10.1109/MOST57249.2023.00011

Li, M., Li, Z., Xu, C., & Liu, T. (2020). Deep reinforcement learning-based vehicle driving strategy to reduce crash risks in traffic oscillations. *Transportation Research Record*, *2674*(10), 42–54. https://doi.org/10.1177/0361198120937976

Wang, C., Xu, Y., Zhang, J., & Ran, B. (2022). Integrated traffic control for freeway recurrent bottleneck based on deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 15522–15535. https://doi.org/10.1109/TITS.2022.3141730

Shen, W., Zou, L., Deng, R., Wu, H., & Wu, J. (2023). A bus signal priority control method based on deep reinforcement learning. *Applied Sciences*, *13*(11). https://doi.org/10.3390/app13116772

Zhong, N., Liu, K., Li, Y., et al. (2023). Deep q-learning network model for optimizing transit bus priority at multiphase traffic signal controlled intersection. *Mathematical Problems in Engineering*, *2023*. https://doi.org/10.1155/2023/9137889

Han, G., Zheng, Q., Liao, L., Tang, P., Li, Z., & Zhu, Y. (2022). Deep reinforcement learning for intersection signal control considering pedestrian behavior. *Electronics*, *11*(21). https://doi.org/10.3390/electronics11213519

Codeca, L., & Cahill, V. (2022). Using deep reinforcement learning to coordinate multi-modal journey planning with limited transportation capacity. *SUMO Conference Proceedings*, *2*, 13–32. https://doi.org/10.52825/scp.v2i.89

Schmidt, K. J., Steinmetz, N., & Margreiter, M. (2024). Bus priority procedure for signalized intersections based on bus occupancy and delay. *SUMO Conference Proceedings*, *5*, 127–145. https://doi.org/10.52825/scp.v5i.1111

Zeng, X., Wu, J., Wang, S., Zhang, X., & Ding, H. (2023). Bus priority hardware-in-the-loop simulation system based on sumo. *2023 5th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 534–537. https://doi.org/10.1109/ICFTIC59930.2023.10455802

Ying-Chuan Ni, Y.-T. H., Hsien-Hao Lo, & Huang, H.-J. (2022). Exploring the effects of passive transit signal priority design on bus rapid transit operation: A microsimulation-based optimization approach. *Transportation Letters*, *14*(1), 14–27. https://doi.org/10.1080/19427867.2020.1805681

Nesmachnow, S., Massobrio, R., Arreche, E., Mumford, C., Olivera, A. C., Vidal, P. J., & Tchernykh, A. (2019). Traffic lights synchronization for bus rapid transit using a parallel evolutionary algorithm. *International Journal of Transportation Science and Technology*, *8*(1), 53–67. https://doi.org/10.1016/j.ijtst.2018.10.002

Ali, M. E. M., Durdu, A., Celtek, S. A., & Yilmaz, A. (2021). An adaptive method for traffic signal control based on fuzzy logic with webster and modified webster formula using sumo traffic simulator. *IEEE Access*, *9*, 102985–102997. https://doi.org/10.1109/ACCESS.2021.3094270

Colombaroni, C., Fusco, G., & Isaenko, N. (2020). A simulation-optimization method for signal synchronization with bus priority and driver speed advisory to connected vehicles [Transport Infrastructure and systems in a changing world. Towards a more sustainable, reliable and smarter mobility.TIS Roma 2019 Conference Proceedings]. *Transportation Research Procedia*, *45*, 890–897. https://doi.org/10.1016/j.trpro.2020.02.079

Lee, W.-H., & Wang, H.-C. (2022). A person-based adaptive traffic signal control method with cooperative transit signal priority. *Journal of Advanced Transportation*, *2022*(1), 2205292. https://doi.org/10.1155/2022/2205292

Rakkesh, S. T., Weerasinghe, A., & Ranasinghe, R. (2017). Equipoising multi-modal traffic environments using vehicular ad-hoc networks. *The International Journal on Advances in ICT for Emerging Regions*, *10*(2). https://journal.icter.org/index.php/ICTer/article/view/232

Lu, G., Ge, Y., Wang, M., Wang, J., Wei, D., Kang, C., & Yu, R. (2020). Green light optimal speed advisory systems under multi-modal traffic environments for reducing fuel consumption. *2020 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 1470–1474. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00221

Puterman, M. L. (1990). Chapter 8 markov decision processes. In *Stochastic models* (pp. 331–434, Vol. 2). Elsevier. https://doi.org/10.1016/S0927-0507(05)80172-0

AlMahamid, F., & Grolinger, K. (2021). Reinforcement learning algorithms: An overview and classification. *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1–7. https://doi.org/10.1109/CCECE53047.2021.9569056

NTM. (2024). Passenger occupancy. https://www.transportmeasures.org/en/wiki/manuals/11-road-passenger-bus-transport/11-8-passenger-occupancy/

Mohan, A., Zhang, A., & Lindauer, M. (2024). Structure in deep reinforcement learning: A survey and open problems. *Journal of Artificial Intelligence Research*, *79*, 1167–1236. https://doi.org/10.1613/jair.1.15703

Neufeld, A., & Sester, J. (2023). Robust Q-learning algorithm for markov decision processes under wasserstein uncertainty. https://doi.org/10.48550/arXiv.2210.00898

Watkins, C. J. C. H. (1989). Learning from delayed rewards.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. A. (2013). Playing atari with deep reinforcement learning. *CoRR*, *abs/1312.5602*. http://arxiv.org/abs/1312.5602

OpenAI. (2017). Openai baselines: Acktr & a2c. https://openai.com/index/openai-baselines-acktr-a2c/

Lowe, R., WU, Y., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf

Allsop, R. E. (1974). Some possibilities for using traffic control to influence trip distribution and route choice. *Transportation and traffic theory, proceedings*, *6*.

Wang, X., Yin, Y., Feng, Y., & Liu, H. X. (2022). Learning the max pressure control for urban traffic networks considering the phase switching loss. *Transportation Research Part C: Emerging Technologies*, *140*, 103670. https://doi.org/10.1016/j.trc.2022.103670

Genders, W., & Razavi, S. (2019). An open-source framework for adaptive traffic signal control. https://doi.org/10.48550/arXiv.1909.00395

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, *22*(268), 1–8. http://jmlr.org/papers/v22/20-1364.html

YanivHacker. (2022). Rltrafficmanager. https://github.com/YanivHacker/RLTrafficManager

Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., & Li, Z. (2019). Presslight: Learning max pressure control to coordinate traffic signals in arterial network. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1290–1298. https://doi.org/10.1145/3292500.3330949

DHPC. (2024). Delftblue supercomputer (phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2

Alegre, L. N. (2019). Sumo-rl. https://github.com/LucasAlegre/sumo-rl

A

Scientific article

Deep reinforcement learning for traffic light control optimization in multi-modal simulation of SUMO

Yimin Xu

Abstract

This research investigates the application of different deep reinforcement learning methods for optimizing traffic light control in multi-modal urban traffic environments using the SUMO traffic simulator. Urban traffic congestion, with its significant economic, environmental, and social impacts, necessitates more sophisticated control strategies that can adapt to varying traffic conditions. Traditional traffic control systems, like fixed-time and adaptive methods, are often insufficient in handling the complexity of multi-modal traffic, which includes various traffic modes such as passenger cars and buses. Deep reinforcement learning, with its ability to dynamically optimize traffic light control without requiring prior knowledge of traffic patterns, is a promising method to improve traffic efficiency and achieve transit priority in multi-modal traffic. The research aims to address the limitations of the existing relevant research by employing several deep reinforcement learning algorithms, particularly multi-agent deep reinforcement learning methods, to coordinate multiple traffic lights in SUMO simulation. Research experiments are conducted in three different cases, which are set in road networks of different sizes respectively, and fixed traffic light control and max-pressure traffic light control are implemented for comparison. The applied deep reinforcement methods are evaluated in terms of training process and model evaluation. And the research results demonstrate that deep reinforcement learning methods, especially multi-agent deep reinforcement learning methods, can significantly enhance traffic flow efficiency and achieve transit priority in complex urban settings in multi-modal simulation.

1 Introduction

Traffic congestion poses a significant challenge in many urban areas, detrimentally impacting the quality of life and environmental sustainability. Increased travel times, fuel consumption, and emissions associated with congestion contribute to economic losses, air pollution, and greenhouse gas emissions. Additionally, congestion-induced stress and reduced accessibility to essential services disproportionately affect vulnerable populations, highlighting the urgent need for effective solutions to optimize multi-modal traffic flow and mitigate the adverse consequences of traffic congestion.

Multi-modal traffic refers to the traffic flow of two or more traffic modes within a road network, which can include different major transport modes like passenger vehicles, buses, trams, trains, bicycles, pedestrians, and so on. Achieving transit priority in multi-modal traffic is of great importance as this can help to reduce congestion and improve the efficiency of public transportation [13]. Traffic control is one of the key strategies to reduce congestion, improve traffic flow, and achieve transit priority. However, it is also a complex and dynamic task that requires careful planning and coordination among different modes of transportation, such as passenger cars, and public transport. Traditional control methods like fixed traffic light control rely on handcrafted rules or fixed schedules that may not adapt well to changing traffic conditions or user preferences. Adaptive traffic light control systems are not effective in handling highly dynamic and complex traffic patterns [40]. Responsive traffic light control systems may struggle with scalability and adaptability to rapidly changing traffic conditions [8].

Currently, reinforcement learning (RL), especially deep reinforcement learning (DRL), has also attracted attention as a promising technique for helping optimize traffic control [39, 14, 45], as it can potentially overcome the limitations of traditional methods such as the need for prior knowledge, and adapt to changing traffic patterns in real-time, making them suitable for dynamic and unpredictable environments like urban traffic networks. It is a type of machine learning where agents learn from their actions and states within an environment to maximize rewards.

Based on the methods used in reinforcement learning, it can be divided into the following three categories [7]:

- **Model-based RL:** In model-based reinforcement learning, the main goal of the learner is to estimate the underlying model of the environment and then improve the policy based on this estimated model [7].
- **Value-based RL:** In value-based reinforcement learning, attention turns towards approximating the value function, and policy updates are guided by the estimated value function [7].
- **Policy-based RL:** Policy-based reinforcement learning enhances the performance of the agent through direct policy updates [7].

Deep reinforcement learning (DRL) extends traditional reinforcement learning by using deep neural networks to approximate decision-making functions like the value function, policy, or environmental model. This advancement has allowed agents to function in high-dimensional, complex environments where traditional methods fall short.

SUMO is an open-source microscopic traffic simulator that supports the modeling of complex, multi-modal traffic systems, including vehicles, public transport, and pedestrians. It accounts for various factors influencing traffic flow, such as road

layout, vehicle behavior, and user interactions. SUMO also supports custom model integration and provides multiple APIs for external simulation control. A notable feature is the Traffic Control Interface (TraCI), which enables real-time manipulation of an active traffic simulation [24]. Due to these features, SUMO is employed as the simulation tool in this research.

Even though there are already many studies on deep reinforcement learning for traffic control optimization in traffic simulation of SUMO, there are still some limitations in the existing relevant studies as stated in Section 2.3. This research aims to address the mentioned limitations by applying different deep reinforcement learning methods for traffic light control optimization to improve traffic efficiency and achieve transit priority in multi-modal traffic scenarios by using SUMO simulator.

The scope of this research includes the following aspects:

- The research area is set in the urban area, as urban areas typically have multi-modal traffic and more complex traffic dynamics, which makes traffic control a challenging problem and an interesting domain for applying deep reinforcement learning.
- The traffic control in this research focuses on traffic light control, as it is the main traffic control means in the urban area.
- The traffic light control in the research is formulated as a discrete-time process, as it allows for more manageable and computationally efficient modeling of the complex traffic dynamics.
- The multi-modal traffic simulation is set up by using SUMO, as SUMO is able to model different traffic modes and achieve good interactions between vehicles and traffic lights.
- DRL algorithms, especially MADRL which can handle the situation of multiple traffic lights and facilitate collaboration between traffic lights, are applied to optimize traffic light control.

The main objective of this research is to optimize and evaluate DRL-based traffic light control for multi-modal traffic scenarios using SUMO simulator, which is achieved by answering the following main research question:

How can deep reinforcement learning algorithms be effectively applied to optimize traffic light control in multi-modal simulation of SUMO?

And the main research question is jointly answered by the following sub-questions:

1. *How to achieve transit priority in multi-modal simulation?*
2. *How to effectively represent the state space of a traffic system in multi-modal simulation of SUMO for deep reinforcement learning?*
3. *What actions are applied to improve traffic flow and reduce congestion in multi-modal simulation of SUMO?*
4. *What reward functions can effectively incentivize traffic control algorithms to improve traffic flow and reduce congestion in multi-modal simulation of SUMO?*
5. *What deep reinforcement learning algorithms can be used for traffic light control optimization in multi-modal simulation of SUMO?*
6. *How to define performance metrics and evaluate the performance of deep reinforcement learning algorithms for traffic light control optimization in multi-modal simulation of SUMO?*

The structure of the following paper is arranged as follows. Section 2 explores the existing literature concerning deep reinforcement learning for traffic control optimization in traffic simulation of SUMO and identifies some research limitations in the relevant research so far. Section 3 describes the principle of the main research methods used for multi-modal simulation. Section 4 describes the case study of the research including the simulation setups, the implementation of different DRL algorithms, the comparison methods, the computation setups, the experimental evaluation for these methods, as well as the experimental results with analysis and discussions on different DRL algorithms in different cases. Section 5 summarizes the main findings, contributions, limitations, and implications of the thesis, and propose some recommendations for future research.

2 Literature review

This section provides a review of the application of deep reinforcement learning for traffic control optimization in traffic simulation of SUMO at three different levels: intersection level, vehicle level, and combined level. The intersection level focuses on optimizing traffic light control to improve traffic flow and reduce congestion. The vehicle level focuses on optimizing the behavior of individual vehicles to improve traffic efficiency and safety. The combined level focuses on coordinating controls of different levels to achieve optimal traffic performance across an entire transportation network.

The literature review presented in this section discusses the application of DRL for traffic control optimization at each of these levels. The various DRL algorithms that have been used, the performance of these algorithms, and the challenges and opportunities for future research are discussed.

2.1 Deep reinforcement learning for traffic control optimization in single-modal simulation of SUMO

Currently, the majority of the relevant research about deep reinforcement learning for traffic control optimization using SUMO is conducted in single-modal simulation, and the utilized traffic control varies a lot including at intersection, vehicle, and combined level for different optimization objectives.

2.1.1 Deep reinforcement learning for traffic control optimization at intersection level in single-modal simulation of SUMO

In terms of the studies about deep reinforcement learning for traffic control optimization at intersection level in single-modal simulation of SUMO, DRL of different categories are applied.

Many related studies use value-based DRL, Deep Q-Network (DQN) or its variants, as the main research methods to optimize traffic light control in single-modal simulation. Genders and Razavi employed DQN to construct an adaptive traffic light control agent in SUMO. This system effectively reduced average cumulative delay, average queue length, and average travel time [10]. Chen, Zhu, and Lu introduced a new adaptive signal control method utilizing DQN to coordinate traffic light controls on arterial roads. Through traffic data detected in real time, the hybrid global and local reward functions were optimized, showcasing the effectiveness and efficiency of the method over actual and fix-time traffic light control methods in SUMO simulations [4]. Guo et al. utilized a DQN DRL approach for traffic light control optimization at urban intersections. Their simulation results demonstrated that the approach converged well and generalized effectively, showing notable improvements in wait time and queue length compared to some benchmarking traffic light control approaches [11]. Bouktif, Cheniki, and Ouni adopted a Parameterized Deep Q-Networks (P-DQN) DRL, considering both continuous and discrete decisions for traffic light control optimization. Their SUMO simulation results illustrated that the presented framework using P-DQN markedly reduced average queue length and travel time compared to the alternative traffic light control systems based on deep reinforcement learning [3]. Wang et al. introduced EP-D3QN, a DRL method for optimizing traffic light timing that utilizes double dueling deep Q-network (3DQN), self-organizing traffic light control, and max pressure traffic light control. The simulation results using SUMO demonstrated that EP-D3QN is better than the other four methods in the scenarios of heavy and light traffic flow respectively, reducing the travel time and waiting time of vehicles [43]. Yu et al. introduced an innovative approach to enhance the service level of both the bus system and car traffic within a multi-modal road network. By integrating bus priority and holding strategies with traffic light control using decentralized DQN controllers, the proposed approach surpasses the performance of model-based adaptive traffic light control approaches and the centralized reinforcement learning approach in terms of bus stability and traffic efficiency [48]. Shabab, Ali, and Zaki applied DQN for traffic light control optimization in the road network of multiple intersections. Simulations conducted in SUMO demonstrate that the suggested deep reinforcement learning model, by optimizing traffic signal timing across multiple intersections, greatly decreases both waiting time and traffic conflict when compared with the benchmark and is beneficial for both safety and mobility [37]. Li et al. presented an enhanced Distributional DQN to develop a traffic light control optimization decision-making model , which effectively leverages intersection environment information for each phase action to predict the distribution of future total returns. Their experiment results show that the Distributional DQN achieves a quicker rate of convergence compared to DQN and has a much lower cumulative intersection delay and higher mean driving velocity [19]. Liang et al. used Dueling Double Deep Q-Network (3DQN) for traffic light control optimization by using the states of small grids discretized from the intersection and the reward calculated by the difference in cumulative waiting time between two consecutive time steps. The simulation results using SUMO demonstrate the effectiveness of the model in managing traffic lights [23]. Li et al. developed an adaptive traffic light control model in SUMO using the Deep Q-Network algorithm. Real-time traffic data, including the number of vehicles and mean speed at one or more intersections, are utilized as the states of the model. To minimize the mean waiting time, an optimal traffic signal phase and duration are determined by the agents for both single-intersection and multi-intersection scenarios. Testing the model with datasets from three different traffic situations shows that it outperforms three other methods including Q-learning, Webster fixed timing control, and longest queue first method in terms of the mean travel time and waiting time [18].

There are also some studies applying policy-based DRL for traffic light control optimization in single-modal simulation of SUMO. Li, Xu, and Zhang introduced a multi-agent deep deterministic policy gradient (MADDPG) method with centralized learning and decentralized execution, which builds upon the actor-critic policy gradient algorithms. The simulation results of the model demonstrate that this method can efficiently manage traffic lights [21]. Li et al. presented the Fairness Scheduling Proximal Policy Optimization (FSPPO), a DRL algorithm that integrates the Proximal Policy Optimization (PPO) algorithm with a fairness criterion. The algorithm aimed to reduce the longest waiting time for drivers during a traffic light cycle, and the results indicated its efficient optimization for the fairness criterion [17]. Ma et al. introduced a traffic light timing optimization strategy utilizing Proximal Policy Optimization (PPO), which enables traffic lights to select proper phases based on the traffic conditions for each direction of the corresponding intersection. and to dynamically adjust the duration of these phases. Experiments conducted on actual traffic data using SUMO demonstrate that this approach significantly reduces the queue length and vehicle waiting times under different traffic scenarios compared to conventional traffic light control methods [27]. Kumar et al. suggested a traffic-light scheduling scheme using SDDRL, incorporating the dynamics of vehicles from real-time traffic environments. Their SUMO simulation results demonstrate that the suggested method has improved multiple performance indicators including throughput, mean speed, mean waiting time, and mean queue length compared to several state-of-the-art methods including DQN, NFM, FLTC, FTA, and MPA [16]. Li et al. introduced Knowledge Sharing Deep Deterministic Policy Gradient (KS-DDPG), a multi-agent DRL approach for optimizing traffic light control through

enhanced agent cooperation. By enabling knowledge sharing, every agent is able to access the combined traffic environment data gathered by all agents. Experiments with synthetic and real-world data show that KS-DDPG outperforms traditional traffic light control methods and state-of-the-art RL-based methods in efficiently managing large-scale road networks and handling traffic flow fluctuations [22].

Morever, some studies utilize both value-based and policy-based DRL for traffic light control optimization in single-modal simulation. Louw, Labuschagne, and Woodley applied two DRL algorithms PPO and DQN to improve urban traffic management on a simulated intersection in SUMO. The experiment results show that both methods showcase significant enhancements over conventional traffic light control methods [25]. Mei et al. applied two DRL algorithms PPO and DQNs to establish a model in SUMO simulation in an area with two signalized intersections and two railroad crossings. The results of SUMO simulation underscore the superior performance of the presented DRL-based traffic light control method over the fixed traffic light control [29].

In conclusion, this review underscores the diverse and effective applications of deep reinforcement learning (DRL) for optimizing traffic control at the intersection level within single-modal SUMO simulations. The studies discussed demonstrate that value-based DRL methods, particularly Deep Q-Networks (DQN) and its variations, have consistently improved traffic efficiency by reducing delays, queue lengths, and travel times across various traffic scenarios. Additionally, policy-based DRL approaches, such as those utilizing Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), have proven effective in more complex, multi-agent environments, further enhancing traffic signal control. Moreover, hybrid approaches combining both value-based and policy-based methods have shown significant promise, underscoring the adaptability and robustness of DRL in managing urban traffic at intersections. These findings collectively suggest that DRL offers a powerful and flexible solution for optimizing traffic signal control, paving the way for more intelligent and responsive urban traffic management systems.

2.1.2 Deep reinforcement learning for traffic control optimization at vehicle level in single-modal simulation of SUMO

In terms of the studies about deep reinforcement learning for traffic control optimization at vehicle level in single-modal simulation of SUMO, value-based or policy-based DRL is also applied in the existing relevant research.

Many related studies use value-based DRL, Deep Q-Network (DQN) or its variants, as the main research methods to optimize traffic control at vehicle level. Koh et al. introduced an improved DQN approach for constructing a smart vehicle routing and navigation system that operates in real time. The experimental results show that the introduced approach showcased superior performance over the benchmark algorithms for routing optimization [15]. Makantasis, Kontorinaki, and Nikolos used Double Deep Q-Network (DDQN) DRL for developing a driving policy under the traffic situations both involving manual and autonomous driving vehicles on the road. The simulation results of SUMO show that the developed DRL-based driving policy has better performance over SUMO policies, under the scenarios with and without the introduction of uncertainties [28]. In a different study, Zhao, Qu, and Xu utilized the DDQN DRL to model the decision-making process and interactions among vehicles during highway driving. The agent vehicle, as per the simulation results, demonstrated the capability to accomplish the highway driving task easily, approximating the maximum safe driving velocity that avoids collisions [49].

There are also some research applying policy-based DRL for traffic control optimization at vehicle level in single-modal simulation of SUMO. Bagwe et al. introduced an innovative method for ensuring the robust on-ramp merge of Connected and Autonomous Vehicles through Augmented and Multi-modal DRL considering comfortable driving behavior, driving safety, and traffic efficiency. To enhance the reliability of the merging operations, surveillance images and basic safety messages are both used at the same time for multi-modal observation to train a policy model using PPO with augmented data. The simulation results of SUMO demonstrate the effectiveness and efficiency of their robust on-ramp merging design in two typical merging scenarios [2]. In another study, Li et al. used Deep Deterministic Policy Gradient (DDPG) to develop a driving strategy for individual vehicles with the purpose of reducing oscillations and enhancing traffic safety in stop-and-go waves. Their SUMO simulation results demonstrate that the developed strategy outperforms the adaptive cruise control and jam-absorbing driving strategies, showcasing its effectiveness in lowering the risk of accidents [20]. Szőke et al. employed the Vanilla Policy Gradient method to develop an agent for navigation and driving behavior optimization. The experimental results reveal that the developed agent can safely navigate through varying highway traffic conditions and successfully traverse the specified section while maintaining the reference velocity in a preset highway situation [41].

In summary, the application of deep reinforcement learning (DRL) for traffic control optimization at the vehicle level within single-modal SUMO simulations has yielded promising results. Both value-based and policy-based DRL approaches have been effectively employed to enhance vehicle routing, driving policies, and decision-making processes. Studies utilizing Deep Q-Networks (DQN) and its variants have demonstrated significant improvements in vehicle navigation, route optimization, and highway driving scenarios, showcasing the ability of these methods to adapt to dynamic traffic conditions and avoid congestion. Additionally, policy-based DRL methods, such as Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), have proven effective in addressing challenges like safe merging of autonomous vehicles and mitigating traffic oscillations. Collectively, these findings highlight the potential of DRL to optimize vehicle-level traffic control, enhancing both safety and efficiency in various driving environments.

2.1.3 Deep reinforcement learning for traffic control optimization at combined level in single-modal simulation of SUMO

In terms of the studies about deep reinforcement learning for traffic control optimization at combined level, much fewer studies are conducted and the existing study of Wang et al. uses both policy-based DRL and actor-critic DRL which uses both a policy network (actor) and a value network (critic) to learn optimal policies.

Wang et al. presented a centralized traffic control system featuring a unique double-layer structure designed to synchronize a plurality of variable speed limit and ramp metering traffic controllers. The policy-based DRL approaches are incorporated in the system to learn coordinated actions within a high-dimensional traffic environment on the freeway. The simulation results indicate the superiority of policy-based methods over alternative approaches including TD3, DDPG, fixed-time ramp metering, and integrated feedback control in terms of the total travel time savings in the scenarios of a single-ramp interweaving area and a large freeway corridor with a plurality of off-ramps and on-ramps [44].

In conclusion, research on deep reinforcement learning (DRL) for traffic control optimization at the combined level in single-modal SUMO simulations is relatively sparse, with only a few studies exploring this complex area. The work by C. Wang et al. stands out by employing policy-based DRL to develop a centralized traffic control system that synchronizes multiple traffic management strategies, such as ramp metering and variable speed limits. This approach has demonstrated the potential to significantly enhance freeway mobility and reduce congestion, particularly in challenging traffic environments. The study's results underscore the effectiveness of policy-based DRL in managing high-dimensional traffic scenarios, offering a promising direction for future research in integrated traffic control systems.

2.2 Deep reinforcement learning for traffic control optimization in multi-modal simulation of SUMO

Compared to the research about deep reinforcement learning for traffic control optimization in multi-modal simulation, much fewer studies conduct the research in multi-modal simulation. And it is often the case that only one kind of traffic control at intersection or vehicle level is used in each study.

2.2.1 Deep reinforcement learning for traffic control optimization at intersection level in multi-modal simulation of SUMO

As for the research regarding deep reinforcement learning for traffic control optimization at intersection level, most studies focus on using DQN as their training algorithms to optimize traffic light control.

Shen et al. utilized DQN for managing bus signal priority at intersections within a bus network, which considers the needs of both general traffic and pedestrians while ensuring priority access for each bus flow. The simulation experiments using SUMO demonstrate that the proposed method significantly decreases bus waiting times and the mean waiting time of all vehicles in comparison to the actuated traffic light control and the fixed-time traffic light control methods [38].

A system that uses connected vehicle technology to give priority to transit signals is proposed by Zhong, Liu, Li, et al., employing DQN for traffic light control optimization. The system utilizes Vehicle-to-Infrastructure (V2I) communication technology to gather real-time data on vehicle movement, traffic light phase information, and states of the road traffic. Experiments conducted using SUMO demonstrate that the proposed system has significant reductions in both vehicle delay and cumulative delay per passenger in comparison to the conventional traffic light control under the scenarios with low and medium traffic densities [50].

Han et al. proposed a traffic light control method considering pedestrian behavior based on the Dueling Double Deep Q-Network (3DQN) algorithm, which integrates the strengths of Double DQN and Dueling DQN algorithms. The method simultaneously considers traffic safety, traffic efficiency, pedestrian waiting times, and vehicle waiting times. Simulation experiments conducted using SUMO on real intersection scenarios demonstrate that the proposed approach significantly reduces both pedestrian waiting times and the number of pedestrians waiting at crossroads compared with the Dueling DQN method [12].

In summary, the application of deep reinforcement learning (DRL) for traffic control optimization at the intersection level in multi-modal SUMO simulations has primarily focused on enhancing traffic light management using DQN-based approaches. The studies reviewed have shown significant advancements in reducing vehicle and pedestrian waiting times while improving overall traffic flow. By incorporating considerations for various traffic participants, such as buses and pedestrians, these DRL methods have demonstrated superior performance compared to traditional control techniques. The success of these approaches highlights the potential of DRL to create more efficient and responsive traffic control systems in multi-modal urban environments.

2.2.2 Deep reinforcement learning for traffic control optimization at vehicle level in multi-modal simulation of SUMO

As for the research regarding deep reinforcement learning for traffic control optimization at vehicle level, only the existing study of Codeca and Cahill has been known so far.

Codeca and Cahill assessed the potential of employing multi-agent PPO (MAPPO) to coordinate journey plans in large-scale events like concerts and sports events. They utilize multi-agent MAPPO to develop synchronized plans that consider the availability, needs, and constraints of various traffic modes with the objective of increasing just-in-time arrival. The findings

indicate that the devised plan effectively enhances the mean travel time and punctuality rates compared to a simplistic decision-making algorithm relying on estimated travel times [5].

In conclusion, the study by Codeca and Cahill offers valuable insights into the use of multi-agent PPO for optimizing vehicle-level traffic control in multi-modal SUMO simulations, particularly in the context of large-scale events. Their research demonstrates the effectiveness of coordinated transportation planning, taking into account various transportation modes and infrastructure constraints. The results highlight the potential of advanced DRL methods like MAPPO to significantly enhance travel efficiency and punctuality, providing a promising approach to managing complex traffic scenarios during major events.

2.3 Discussion

The reviewed literature has shown that the potential of DRL for traffic control optimization is significant, and has provided valuable insights into the application of DRL for traffic control optimization. However, there are still several limitations existing in the relevant literature.

One of the limitations is the limited consideration of using total travel time as the optimization indicator for the application of DRL in multi-modal simulation of SUMO. Most existing studies focus on scenarios with only passenger vehicles, while real-world traffic systems also include other traffic modes. Even though there are few relevant studies considering multiple traffic modes, they did not use total travel time as the optimization indicator. Total travel time includes not only the time spent waiting at traffic lights but also the time spent moving through the network. And total travel time can also capture the impact of various traffic conditions, including congestion, traffic light delays, and road capacity, which makes it a more holistic indicator than other factors like total waiting time, total delay, and so on.

The second limitation is that the potential of using single agent to control multiple traffic lights remains unexplored. So far, for the case of multiple intersections, related studies have used multi-agent deep reinforcement learning methods but have not considered using a single agent to control multiple intersections.

Another limitation is that the performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes is not explored. Exploring the performance of deep reinforcement learning for traffic light control optimization in road networks of different sizes is crucial because it allows for a comprehensive understanding of the scalability and adaptability of DRL algorithms. Different road network sizes present unique challenges and complexities, such as varying traffic patterns, congestion levels, and infrastructure constraints. Applying DRL only to a single road network cannot well show the strengths, limitations, and its applicability for different situations.

In addition, the evaluation of DRL-based traffic light control policies is very limited in the reviewed literature. Most of the related studies only mention the training process and do not conduct a reasonable and diversified evaluation of the DRL training model. Therefore, the effect and applicability of DRL on traffic control can not be fully reflected.

Based on the findings of the literature review, several points are presented to address the current limitations:

- **Expand the scope of DRL research to use total travel time as the optimization indicator in multi-modal simulation.**
  Total travel time accounts for the entire journey of a vehicle from start to finish, providing a more comprehensive measure of traffic efficiency.

- **Explore the use and effectiveness of using different single-agent DRL to control multiple traffic lights in multi-modal simulation.**
  A single-agent DRL system has the potential to optimize traffic flow holistically by considering the states and the impact of decisions on the entire network rather than making independent decisions for each intersection, which might lead to better training performance compared to the multi-agent DRL system.

- **Explore the performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes.**
  By exploring the performance of different DRL for traffic light control optimization in multi-modal simulation in road networks of different sizes, the effectiveness of different DRL in different situations can be fully demonstrated.

- **Evaluate different DRL in terms of different aspects.**
  By evaluating different deep reinforcement learning methods considering more factors, the advantages and disadvantages of each deep reinforcement learning can be analyzed from a more comprehensive perspective, showing the effect and applicability of each method.

By addressing these research limitations, DRL has the potential to revolutionize traffic control and significantly improve traffic efficiency, safety, and sustainability.

2.4 Conclusion

This literature review provides a comprehensive overview of the application of deep reinforcement learning for traffic control optimization at three different levels: intersection level, vehicle level, and combined level. Plenty of studies on DRL-based traffic control have demonstrated promising results in improving traffic efficiency and reducing congestion. However, there are still some limitations in the existing relevant literature to be addressed. Several points are presented to solve the limitations existing in the relevant research.

3 Research methodology

This section illustrates the research methodology used for this research in detail. In this research, the process of traffic light control is modeled as a Markov Decision Process. And different single-agent and multi-agent DRL including PPO, A2C, MAPPO, MAA2C, and MADQN are applied to optimize traffic light control in multi-modal simulation as all of them support multiple action choices for one agent at each step. In order to test the effectiveness and performance of these DRL algorithms, fixed traffic light control and max-pressure traffic light control are also implemented for comparison.

3.1 Markov Decision Process

A Markov Decision Process (MDP) is introduced by Puterman, which provides a framework for representing and analyzing the interaction between an agent and its environment. It is characterized by a set of states, a set of actions available to the agent in each state, a transition model describing the probability of moving from one state to another given a specific action, and a reward function that assigns a numerical reward to each state-action pair.

Traffic light control in multi-modal simulation using SUMO can be effectively formulated as an MDP for several reasons:

1. **State Dependence:** MDPs rely on the principle of Markov property, where the future state depends on the current state and chosen action. This aligns well with traffic light control, where the current traffic conditions fully determine the potential outcomes of the next action.

2. **Discrete States and Actions:** Even though the real traffic environment is continuous, SUMO discretizes both states and actions for simulation purposes. This discrete nature makes it compatible with the discrete framework of MDPs.

3. **Sequential Decision Making:** Traffic light control involves making a series of decisions over time, where each decision affects the subsequent state and possible future decisions. This sequential decision-making process mirrors the core concept of MDPs.

4. **Reward Signal Definition:** MDPs require a reward signal to evaluate the effectiveness of chosen actions. In traffic light control simulations, various relevant indicators can be defined as rewards, such as travel time, queue length, and fuel consumption.

5. **Reinforcement Learning Compatibility:** MDPs offer a solid foundation for applying reinforcement learning algorithms, which are particularly suited to multi-agent environments. By learning from reward feedback, the control system can adapt its actions dynamically to optimize chosen performance metrics.

3.1.1 State design

In this research, the traffic information of each edge in the road network is assumed to be known. The weighted traffic volume (i.e. number of vehicles) and weighted queue length at each intersection are used as the states for observation, as they can well reflect the traffic conditions on the road and help realize the transit priority.

The weighted traffic volume of the incoming lanes of each green phase for each traffic light at each time step is calculated by the sum of the results of multiplying the total number of each vehicle type in the incoming lanes of each green phase for each traffic light at each time step by the weight of the corresponding vehicle type as shown in Equation 1.

$$Vol_t^{mp} = \sum_{v \in V, l \in L^{mp}} \alpha_v * nv_t^{mlv} \quad \forall t \in T, m \in M, p \in P^m \tag{1}$$

where Vol_t^{mp} is the weighted traffic volume of the incoming lanes of the green phase p for the traffic light m at the time step t, nv_t^{mlv} is the number of vehicles of vehicle type v of the incoming lane l of the traffic light m at the time step t, V is the vehicle type set, α_v is the weight for the vehicle type v, M is the traffic light set, P^m is the green phase set of the traffic light m, and L^{mp} is the lane set of the phase p of the traffic light m, T is the time step set, the length of a time step δ is set to be 5 seconds, because the length of a time step has to be larger than yellow time (3 seconds) to ensure that the new green phase chosen at the last time step can be activated before the new green phase.

The weighted queue length of the incoming lanes of each green phase for each traffic light at each time step is calculated by the sum of the results of multiplying the total queue length of each vehicle type in the incoming lanes of each green phase for each traffic light at each time step by the weight of the corresponding vehicle type as shown in Equation 2.

$$Que_t^{mp} = \sum_{v \in V, l \in L^{mp}} \alpha_v * ns_t^{mlv} \quad \forall t \in T, m \in M, p \in P^m \tag{2}$$

where Que_t^{mp} is the weighted queue length of the incoming lanes of the green phase p for the traffic light m at the time step t, ns_t^{mlv} is the number of stopping vehicles of vehicle type v of the incoming lane l of the traffic light m at the time step t.

According to the research conducted by Network for Transport Measures (NTM) which is a non-profit organization striving to develop a unified framework for assessing the environmental impact of diverse transportation modes encompassing both freight and passenger travel, the usual passenger occupancy of the bus is 15-20 [33]. Thus, in this research, the weight for bus α_{bus} is set to be 20, and the weight for passenger car $\alpha_{passenger}$ is set to be 1.

As for multi-agent DRL, each agent controls one traffic light and the state of the agent is the state of the controlled traffic light, including the normalized weighted volume and the normalized weighted queue length, as shown in Equation 3.

$$s_t^m = [\frac{Vol_t^{mp}}{200}, \frac{Que_t^{mp}}{200}] \quad \forall t \in T, m \in M, p \in P^m, s_t^m \in S^m \tag{3}$$

where s_t^m is the state for the traffic light m at the time step t, S^m is the state set of the traffic light m, P^m is the green phase set of the traffic light m.

As for single-agent DRL, one single agent controls all the traffic lights and the state of the agent is the combination of the states of all the traffic lights as shown in Equation 4.

$$s_t = [s_t^1, s_t^2, ..., s_t^m] \quad \forall t \in T, m \in M, p \in P^m, s_t \in S \tag{4}$$

3.1.2 Action design

In this research, the action for each traffic light is assumed to be which green phase to activate at each time step so that the number of actions for each episode is fixed, which is beneficial to the learning process. Every time a green phase change occurs, the next phase is preceded by a corresponding yellow phase with a duration of 3 seconds.

For multi-agent DRL, each agent controls one traffic light, so the action space of each agent has to be Discrete, which allows each agent to determine the action of the controlled traffic light as shown in Equation 5.

$$action_t^m = p \quad \forall t \in T, m \in M, action_t^m \in A^m, p \in P^m \tag{5}$$

where A^m is the action set of the traffic light m. At each time step t, each agent chooses the action $action_t^m$ for the controlled traffic light m from its green phase set P^m. The number of actions of the traffic light m is equal to the number of green phases of it. That is, $|A^m| = |P^m|$.

For single-agent DRL, one single agent has to control the actions of all the traffic lights. Therefore, the action space of the agent in this research has to be MultiDiscrete, which allows one single agent to determine the action for each traffic light at each time step as shown in Equation 6.

$$a_t = [action_t^1, action_t^2, ..., action_t^m] \quad \forall t \in T, m \in M, a_t \in A, action_t^m \in A^m \tag{6}$$

where A is the action set of the agent. At each time step t, the agent chooses the action $action_t^m$ for the traffic light m from its green phase set P^m.

3.1.3 Reward design

In this research, the weighted travel time of vehicles in all the incoming lanes of the intersection controlled by each traffic light at each time step is calculated by multiplying the weighted traffic volume of the incoming lanes of each green phase for each traffic light at each time step by the length of a time step as shown in Equation 7.

$$TTS_t^m = \sum_{p \in P^m} Vol_t^{mp} * \delta \quad \forall t \in T, m \in M \tag{7}$$

where TTS_t^m is the weighted travel time of vehicles in all the incoming lanes of the intersection controlled by the traffic light m at the time step t.

And the total weighted travel time of vehicles in all the incoming lanes of all the intersections at each time step is the sum of the weighted travel time of vehicles in all the incoming lanes of the intersection controlled by each traffic light at each time step as shown in Equation 8.

$$TTS_t = \sum_{m \in M} TTS_t^m \quad \forall t \in T, m \in M \tag{8}$$

where TTS_t is the total weighted travel time of vehicles in all the incoming lanes of all the intersections at the time step t.

For multi-agent DRL, the reward at each time step for each agent r_t^m in this research is defined as the normalized negative total weighted travel time of vehicles in all the incoming lanes of the traffic light controlled by each agent at the time step t as shown in Equation 9.

$$r_t^m = -\frac{TTS_t^m}{1000} \quad \forall t \in T, m \in M, r_t^m \in R^m \tag{9}$$

where r_t^m is the reward of the traffic light m at the time step t, R^m is the reward set of the traffic light m.

For single-agent DRL, the reward at each time step r_t of the agent in this research is defined as the negative total weighted travel time of vehicles in all the incoming lanes of all the traffic lights at the time step t as shown in Equation 10.

$$r_t = -TTS_t \quad \forall t \in T, r_t \in R \tag{10}$$

3.2 Deep Q-Network

As can be seen from Section 2, many relevant studies apply Deep Q-Network for traffic light optimization, which shows that DQN has already been a commonly used DRL method for traffic control optimization in traffic simulation.

The Deep Q-Network (DQN) algorithm was first introduced by Mnih et al. to address the limitations of Q-Learning [32]. It is a pioneering algorithm in the field of deep reinforcement learning that utilizes a convolutional neural network (CNN) to approximate the Q-values, enabling the agent to generalize across similar states and actions based on their representations in the neural network. This architecture allows DQN to handle complex and high-dimensional state spaces more effectively than traditional Q-Learning [31].

The DQN agent interacts with the environment and stores its experiences (state, action, reward, next state) in an experience replay buffer. From this buffer, random mini-batches of experiences are sampled and passed through the prediction network, which estimates Q-values for various actions. To stabilize learning, a target network, updated less frequently, is used to generate target Q-values. These target values are compared with the predicted Q-values using the DQN loss function, and the resulting error is used to update the prediction network, gradually improving the performance of the agent.

3.3 Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a deep reinforcement learning algorithm introduced by Schulman et al. and is part of the broader category of policy gradient methods, where the policy directly maps states to actions.

The PPO agent interacts with the environment, storing its experiences in a sample memory. Mini-batches of these experiences are drawn from memory and passed to both the actor and critic networks. The actor network calculates the probability ratios, which are then clipped to ensure stable policy updates. Meanwhile, the critic network computes the value function, providing an estimate of the expected returns. The advantage function is used to guide the policy improvement by comparing actual returns with the estimated value. The PPO loss is a combination of the clipped probability ratio (for stable policy optimization) and the value loss (for accurate state-value predictions), and this loss is minimized to improve both the actor and critic networks.

3.4 Advantage Actor Critic

Advantage Actor-Critic (A2C) is a synchronous version of the Asynchronous Advantage Actor-Critic (A3C) algorithm, which is a deep reinforcement learning method introduced by Mnih et al., Dhariwal et al. that enhances the stability and efficiency of training by running multiple agents in parallel, each interacting with separate environments.

The A2C agent interacts with the environment, gathering information about the current state. The actor network uses this state to determine the optimal action policy, deciding which actions the agent should take. Simultaneously, the critic network evaluates the state by estimating the value function, which assesses the expected rewards for the given state. The actor and critic networks work together: the actor optimizes the policy, and the critic helps guide the actor by providing feedback on how good or bad the chosen actions are based on the expected future rewards.

While A3C uses multiple independent neural networks to generate trajectories and update parameters asynchronously, A2C achieves this synchronously. This synchronous update mechanism is more cost-effective and performs better than asynchronous implementations and can leverage GPUs efficiently [34]. So far, A2C has been widely used in reinforcement learning tasks, including Atari games and continuous control [30, 6].

3.5 Multi-agent deep reinforcement learning

Multi-agent deep reinforcement learning (MADRL) builds on traditional DRL by applying it to settings with multiple agents that may have individual goals or work towards a common objective. In these scenarios, agents make decisions through interactions with their environment and each other, refining their strategies based on the rewards or feedback they receive. Based on the way of learning, multi-agent deep reinforcement learning can be classified into two categories: centralized multi-agent DRL and decentralized multi-agent DRL.

In centralized multi-agent deep reinforcement learning, agents either follow a shared policy or are directed by a central controller with access to the global state of the environment. This centralized method enables better coordination among agents, as the central controller can evaluate the actions and states of all agents at once. This can result in more optimized collective strategies, especially in situations where agent interactions are intricate and closely linked [26].

However, centralized multi-agent deep reinforcement learning can be demanding in terms of computation and may struggle to scale with an increasing number of agents or a larger environment. It can also face communication bottlenecks and delays, as the central controller must process and respond to data from all agents in real time. This can be a significant disadvantage in dynamic settings where quick decision-making is crucial.

In contrast, decentralized multi-agent deep reinforcement learning enables each agent to function and learn on its own, relying solely on its local observations and experiences. Each agent formulates its own policy, striving to enhance its performance while taking into account the actions of nearby agents as part of the environment. Decentralized methods are generally more scalable and robust, as they reduce the reliance on a single point of failure and are better equipped to manage the complexity of environments with numerous agents.

In summary, decentralized multi-agent deep reinforcement learning is used in this research due to its scalability and robustness in complex urban environments.

3.6 Fixed traffic light control

Fixed traffic light control is the most common and straightforward method for managing traffic flow at intersections. It operates on a predetermined schedule, cycling through green, yellow, and red lights according to a fixed timing plan which is based on average traffic conditions and remains unchanged regardless of real-time traffic variations. The simplicity and predictability of fixed traffic light control make it an easily implemented and widely used approach.

However, fixed traffic light control also has some disadvantages. The most significant is its lack of adaptability. Because the timing of the lights does not change in response to real-time traffic conditions, fixed traffic light control can lead to unnecessary delays during off-peak hours or when traffic is unusually light. Additionally, it can struggle to effectively manage traffic during unexpected surges or incidents.

Despite these limitations, fixed traffic light control remains a viable option for managing traffic in many situations, particularly in areas with predictable traffic patterns or limited resources for more advanced control systems. The research conducted by Allsop illustrates the effectiveness of fixed light control in situations where traffic demand is relatively predictable [1].

3.7 Max-pressure traffic light control

Max-pressure traffic light control, designed by Varaiya, is an advanced adaptive traffic light control method for managing traffic at signalized intersections in real time [42]. Unlike fixed traffic light control, max-pressure traffic light control dynamically adjusts the phase based on the current traffic state, aiming to minimize congestion and delays.

The core idea behind max-pressure control is to calculate the pressure of each phase, which is a measure of the imbalance between the incoming and outgoing traffic on links of each phase connected to the intersection. By prioritizing movements that relieve the most pressure, this approach helps to prevent the buildup of queues and smooths traffic flow across the network [46].

So far, max-pressure traffic light control has showed its effectiveness in traffic light control optimization. Comparative experiments conducted by Varaiya demonstrate that the max-pressure control system offers a more reliable and efficient solution for managing traffic at intersections compared to other local controllers such as priority service and fully actuated control [42].

4 Case study

This section describes the case study of the research in detail including the simulation setups, the implementation of different DRL algorithms (PPO, A2C, MAPPO, MAA2C, MADQN), the comparison methods (fixed, max-pressure), the computation setups as well as the experimental evaluation for these methods.

4.1 Simulation setups

4.1.1 Network setups

Network setups are conducted by using *netedit*, which is a graphical network editor in SUMO to create the road network for simulation [24].

This research conducted training under three different cases including a road network of single intersection, a road network of 2*2 intersections, and a road network of 3*2 intersections as shown in Figure 1, Figure 2, and Figure 3 respectively.

For all the cases, each edge has three lanes and the length of each edge is 1 km. At the end of the three lanes, only left turn, straight ahead, and right turn is allowed respectively. And there are traffic lights controlling the traffic flow at each intersection of both road networks. For Case 1, there is a bus line with two opposite routes. For Case 2, there are two bus lines with two opposite routes. For Case 3, there are also two bus lines with two opposite routes. To simplify, one bus stop is set in the middle of each edge that bus routes pass through. And at each bus stop in these routes, the bus is set to stop for 15 seconds to imitate the behavior of a bus picking up and dropping off passengers.

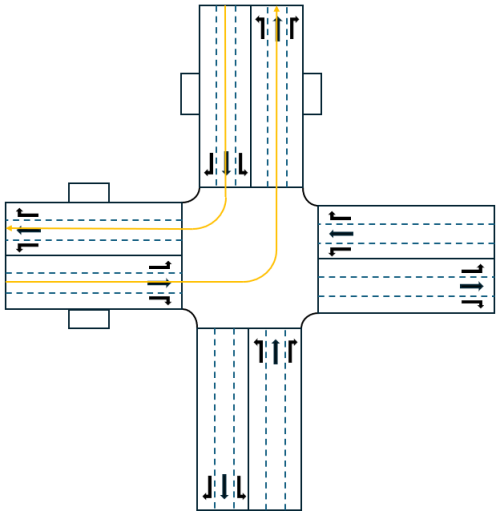


Figure 1: Case 1: road network of single intersection

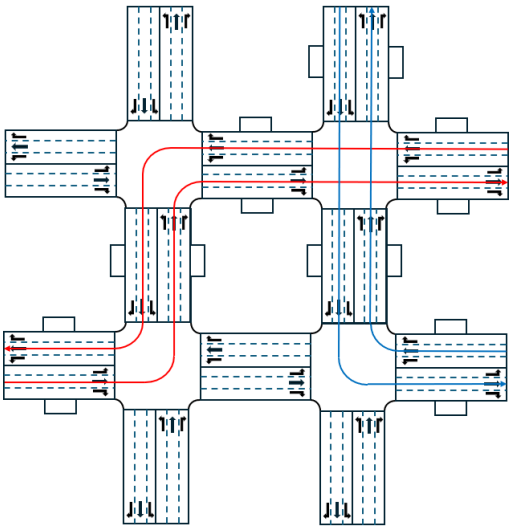


Figure 2: Case 2: road network of 2*2 intersections

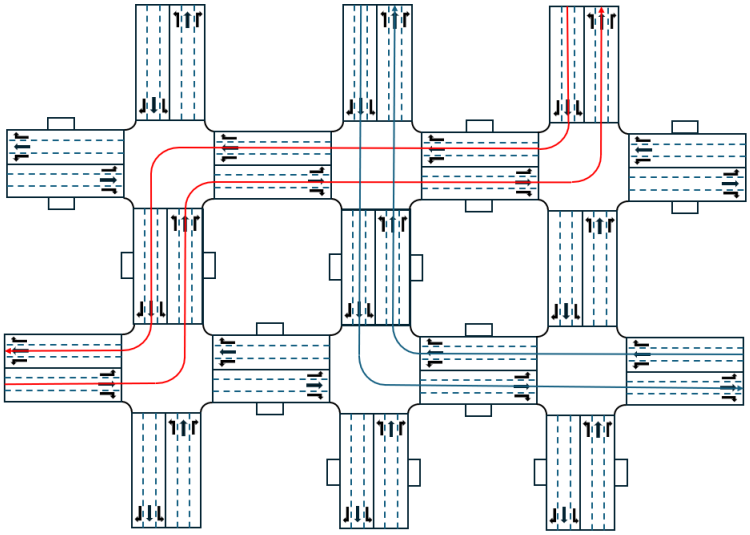


Figure 3: Case 3: road network of 3*2 intersections

4.1.2 Phase setups

As for the phase setups, the typical phase settings are applied in the case study. As shown in Figure 4, the straight and right turns of each pair of opposite roads are designed in one separate phase, and the left turns of each pair of opposite roads are

also designed in one separate phase to ensure smooth passage of left-turning traffic.

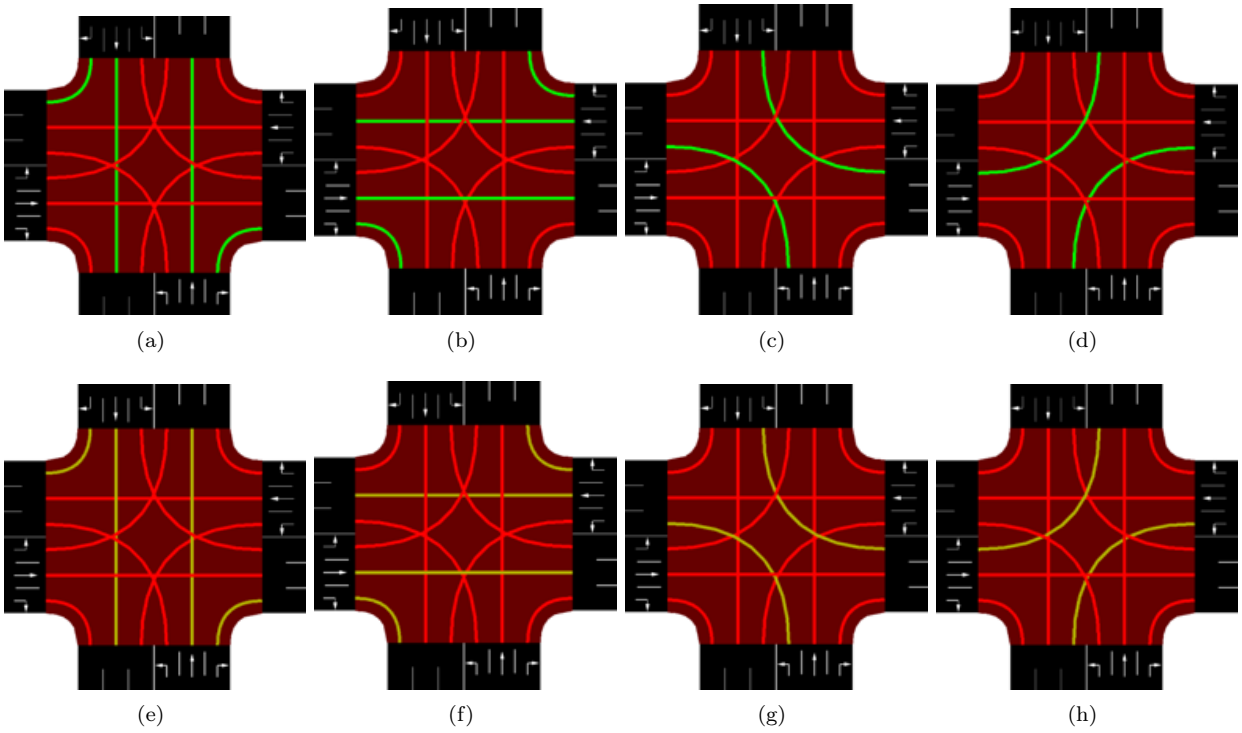


Figure 4: Phase settings

4.1.3 Demand setups

As there are two kinds of traffic modes, passenger cars and buses, considered in this research, the demands of both are set up.

As for the demands of vehicle passengers, *randomTrips* of SUMO is used to generate a set of random trips starting from and ending towards the marginal edges of the road network for passenger cars. For Case 1, the demand for passenger cars is set to be 1500 veh/h. For Case 2, the demand for passenger cars is set to be 2500 veh/h. For Case 3, the demand for passenger cars is set to be 3000 veh/h. The values of these demands are set to cause bottlenecks in each case and avoid the complete blockage of the road network.

As for the demands for buses, 6 buses per hour are set for each bus route for all 3 cases.

4.2 DRL algorithm implementation

The implementation process of the DRL algorithm is shown in Figure 5.

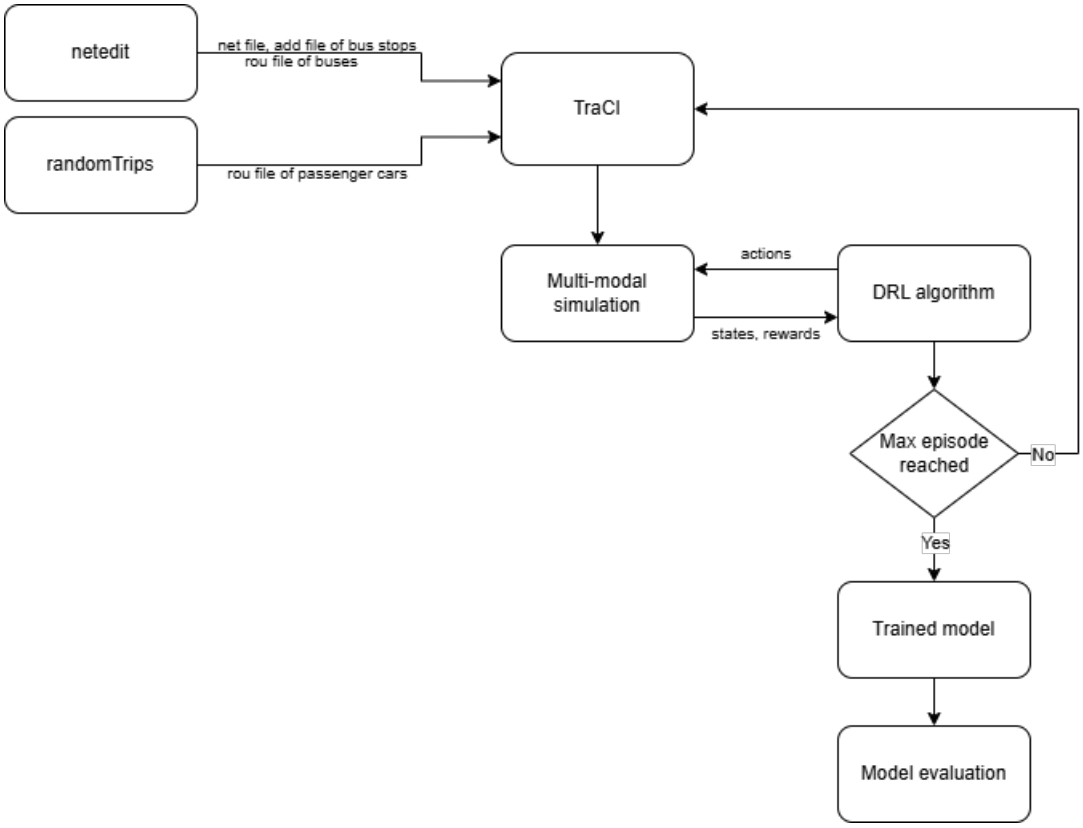


Figure 5: Flow chart of the DRL algorithm implementation process

netedit and *randomTrips* are first used to create the network file, additional file of bus stops, route file of passenger cars, and route files of buses respectively. Then, these files are used by *TraCI* for initializing and running the simulation. During the simulation process, the agents in the DRL algorithm assign the actions for the corresponding traffic lights, and then the simulation continues by conducting the actions and feeding the states and rewards back to the agent in the DRL algorithm. This process is repeated until the maximum number of training episodes is reached, and the trained model is obtained. After the training process of the DRL algorithm is finished, the obtained trained model is evaluated by using two new multi-modal simulations set up by adding 10% random demand and deleting 10% random demand for passenger cars to the original multi-modal simulation.

In this research, the majority of hyperparameters use the default value in the python package *stable_ baselines3*, while the learning rate and number of steps per update of MAA2C vary a little bit. The learning rate of single-agent DRL is 0.0001, the learning rate of multi-agent DRL in Case 2 is 0.001, and the learning rate of multi-agent DRL in Case 2 is 0.0005. The number of steps per update of MAA2C is 128.

4.3 Fixed traffic light control implementation

In the case study, the implementation of fixed traffic light control is set to shift to the next green phase every 15 seconds, and the next green phase is determined by a fixed green phase sequence schedule, which is the sequence as shown in Figure 4.

4.4 Max-pressure traffic light control implementation

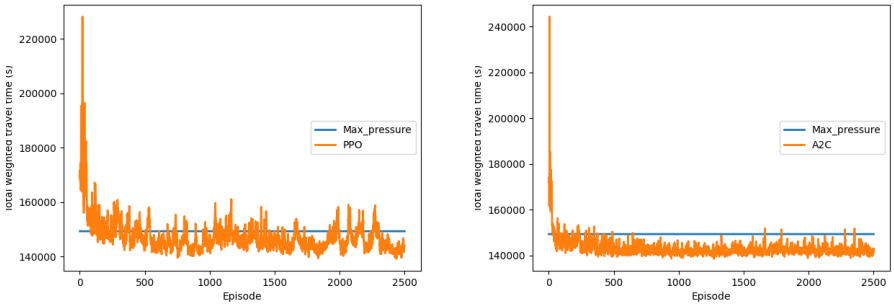
As the max-pressure traffic light control is very complicated and difficult to program and implement directly, many studies use a variant of max-pressure for research instead. A variant of max-pressure used by some research [47, 9] is implemented in this case study. In the case study, the implementation of max-pressure traffic light control is set to choose the next green phase every 15 seconds to guarantee the release of the traffic pressure in time and avoid wasting traffic efficiency due to too frequent green phase changes, and the next green phase is determined by max pressure green phase, which is the green phase having the most difference between the total weighted incoming traffic volume and the total weighted outgoing traffic volume.

4.5 Results

This section provides a description and discussion of the training results and model evaluation results of different DRL algorithms in different cases.

4.5.1 Training results

The training results use line plots to show the change, trend, and training performance of different DRL algorithms in different cases.

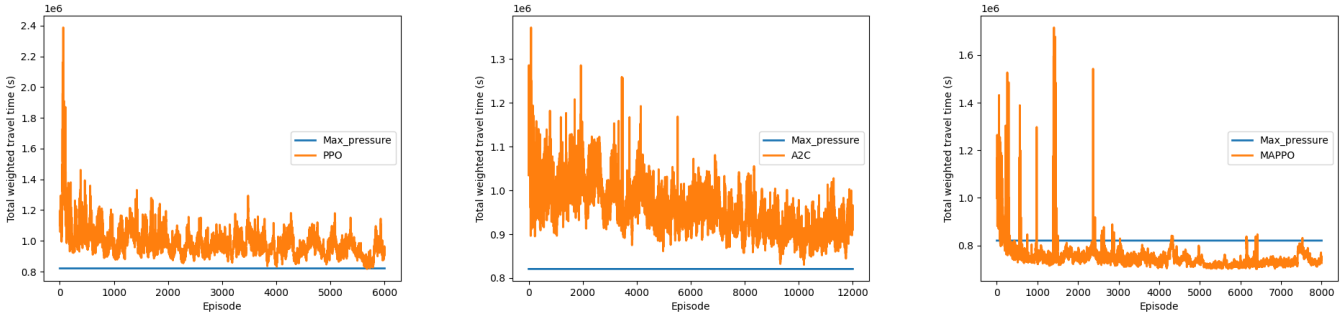


(a) Training process of PPO (b) Training process of A2C

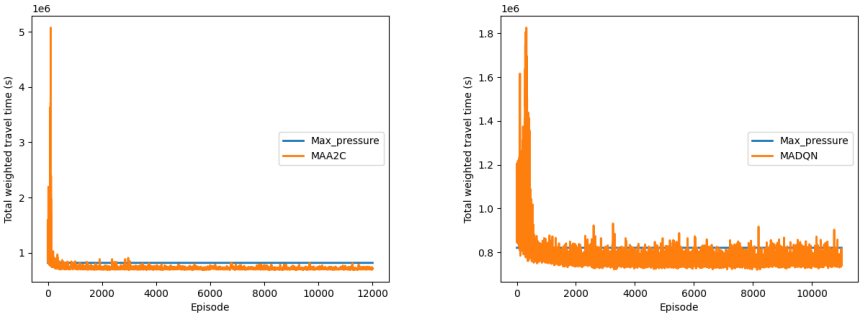
Figure 6: Training Process of different DRL algorithms of Case 1

In Case 1, the total weighted travel time, the total travel time of passenger cars, and the total travel time of buses of fixed traffic light control are 712355 s, 445855 s, and 13325 s respectively.

As can be seen from Figure 6, both PPO and A2C perform slightly better than max-pressure after the convergence of the training process, and A2C shows smaller fluctuations than PPO.



(a) Training process of PPO (b) Training process of A2C (c) Training process of MAPPO

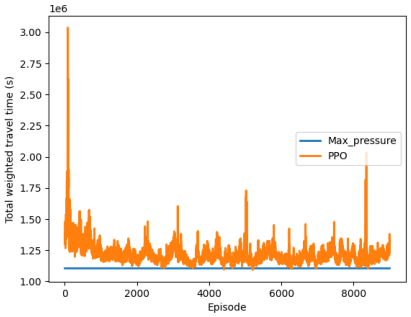


(d) Training process of MAA2C (e) Training process of MADQN

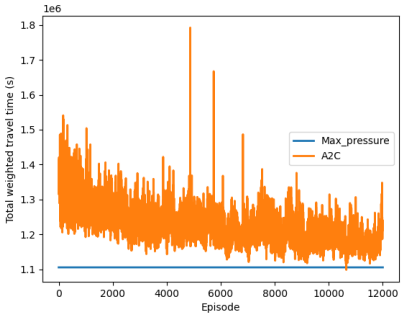
Figure 7: Training Process of different DRL algorithms of Case 2

In Case 2, the total weighted travel time, the total travel time of passenger cars, and the total travel time of buses of fixed traffic light control are 2514635 s, 1951935 s, and 28135 s respectively.

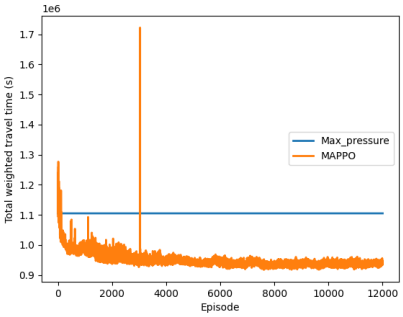
As can be seen from Figure 7, the single-agent deep reinforcement learning methods PPO and A2C become more volatile and perform worse than max-pressure in Case 2, while multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MAPPO perform better than max-pressure and demonstrate smaller fluctuations than single-agent deep reinforcement learning methods after convergence. Among the three multi-agent deep reinforcement learning methods, MAA2C performs best after convergence, MAPPO needs more episodes to converge, and MADQN shows the largest fluctuations after convergence.



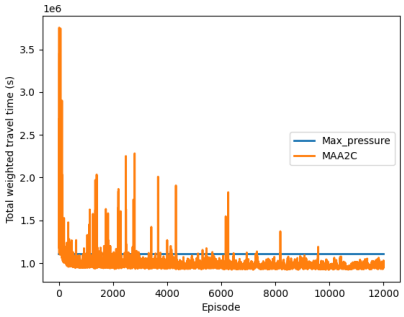
(a) Training process of PPO



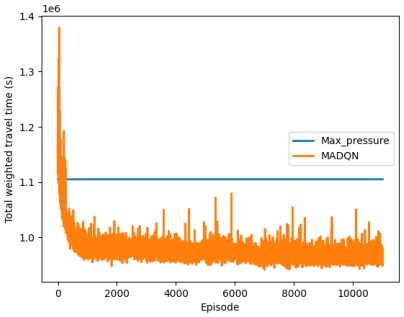
(b) Training process of A2C



(c) Training process of MAPPO



(d) Training process of MAA2C



(e) Training process of MADQN

Figure 8: Training Process of different DRL algorithms of Case 3

In Case 3, the total weighted travel time, the total travel time of passenger cars, and the total travel time of buses of fixed traffic light control are 3126960 s, 2533160 s, and 29690 s respectively.

As can be seen from Figure 8, the single-agent deep reinforcement learning methods PPO and A2C become more volatile and perform worse than max-pressure in 3, while multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MAPPO perform better than max-pressure and demonstrate smaller fluctuations than single-agent deep reinforcement learning methods after convergence. Among the three multi-agent deep reinforcement learning methods, MAPPO performs best after convergence, MAA2C needs more episodes to converge, and MADQN shows the largest fluctuations after convergence.

4.5.2 Model evaluation results

In this research, the model evaluation experiments are conducted under two scenarios with 10% less demand of passenger cars and 10% more demand of passenger cars. In each scenario, the evaluation experiments are repeated for 10 times. The model evaluation results use tables to demonstrate the mean and standard error of the three indicators of DRL performance for different DRL algorithms in different cases. The statistics of model evaluation results of Case 1 are shown in Table 5 and 6.

Table 1: Statistics of model evaluation results with 10% less demand of Case 1

Method	$\overline{T}_{total}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	665095.0	-	398395.0	-	13335.0	-
Max-pressure	135125.0	-	108425.0	-	1335.0	-
PPO	131690.5	633.9	105290.5	477.1	1320.0	13.6
A2C	129384.0	442.5	103164.0	417.8	1311.0	6.6

Table 2: Statistics of model evaluation results with 10% more demand of Case 1

Method	$\overline{T}_{total}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	753430.0	-	486830.0	-	13330.0	-
Max-pressure	158260.0	-	131260.0	-	1350.0	-
PPO	157036.0	843.2	129716.0	950.7	1366.0	26.9
A2C	154631.0	694.3	127561.0	713.9	1353.5	25.3

As can be seen from Table 1 and 2, both PPO and A2C perform better than max-pressure with a small standard error in terms of total weighted travel time and total travel time of passenger cars under the scenarios with 10% less demand and

10% more demand. As for total travel time of buses, PPO and A2C perform similarly to max-pressure. As for standard errors, A2C has lower standard errors for evaluation results than PPO, which means A2C has lower variance than PPO. Overall, A2C performs best with lower total travel time of passenger cars, similar total travel time of buses to max-pressure, and lower variance among all the deep reinforcement learning methods in Case 1.

Table 3: Statistics of model evaluation results with 10% less demand of Case 2

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	2296885.0	-	1734685.0	-	28110.0	-
Max-pressure	736890.0	-	606690.0	-	6510.0	-
PPO	819064.0	20272.4	655684.0	15746.6	8169.0	316.2
A2C	816489.0	6043.8	679729.0	5683.3	6838.0	101.0
MAPPO	663550.5	4359.4	533970.5	3236.2	6479.0	67.9
MAA2C	641319.0	2794.6	514089.0	3166.9	6361.5	35.4
MADQN	675208.5	6119.8	540848.5	5961.2	6718.0	88.7

Table 4: Statistics of model evaluation results with 10% more demand of Case 2

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	2721580.0	-	2158680.0	-	28145.0	-
Max-pressure	886675.0	-	757775.0	-	6445.0	-
PPO	1078274.5	32353.1	883344.5	22847.8	9746.5	546.1
A2C	1047563.0	16627.0	905913.0	16078.0	7082.5	107.8
MAPPO	805418.0	7477.1	672558.0	7118.7	6643.0	50.3
MAA2C	780059.0	4990.4	650749.0	4933.8	6465.5	57.6
MADQN	851752.5	27663.1	704052.5	18478.0	7385.0	496.0

As can be seen from Table 3 and 4, both the single-agent deep reinforcement learning methods PPO and A2C perform worse than max-pressure, but still much better than fixed traffic control. The multi-agent deep reinforcement learning methods MAPPO, MAA2C, and MADQN perform better than max-pressure in terms of total weighted travel time and total travel time of passenger cars and they all have a similar total travel time of buses to max-pressure. MAPPO and MADQN perform worse than MAA2C in terms of total weighted travel time in both scenarios and have quite large standard errors for evaluation results which means that the performance of MAPPO and MADQN have large variance. Overall, MAA2C performs best with the lowest travel time of each mode and variance among all the deep reinforcement learning methods in Case 2.

The statistics of model evaluation results of Case 3 are shown in Table 5 and 6.

Table 5: Statistics of model evaluation results with 10% less demand of Case 3

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	2836885	-	2245185	-	29585	-
Max-pressure	1007545	-	831145	-	8820	-
PPO	1125164.0	14061.4	917664.0	13030.7	10375.0	175.5
A2C	1116447.5	20277.9	912877.5	18981.8	10178.5	203.3
MAPPO	857570.5	3508.6	680670.5	2737.3	8845.0	77.4
MAA2C	869156.5	9406.5	693766.5	9277.3	8769.5	45.7
MADQN	879149.0	7454.0	695799.0	5926.4	9167.5	147.4

Table 6: Statistics of model evaluation results with 10% more demand of Case 3

Method	$\overline{T}$ (s)	σ_{total}	$\overline{T}_{passenger}$ (s)	$\sigma_{passenger}$	$\overline{T}_{bus}$ (s)	σ_{bus}
Fixed	3396745.0	-	2804445.0	-	29615.0	-
Max-pressure	1214005.0	-	1035305.0	-	8935.0	-
PPO	1374221.5	25523.1	1156591.5	22259.8	10881.5	251.3
A2C	1379712.0	40784.1	1168222.0	39914.3	10574.5	390.1
MAPPO	1026981.5	4987.7	847811.5	4829.3	8958.5	63.1
MAA2C	1034160.0	12214.5	855570.0	12007.7	8929.5	59.3
MADQN	1064534.5	10186.8	876634.5	8238.0	9395.0	136.5

As can be seen from Table 5 and 6, both the single-agent deep reinforcement learning methods PPO and A2C perform worse than max-pressure, but still much better than fixed traffic control. The multi-agent deep reinforcement learning

methods MAPPO, MAA2C, and MADQN perform much better than max-pressure in terms of total weighted travel time and total travel time of passenger cars and they all have a similar total travel time of buses to max-pressure. MAA2C and MADQN perform worse than MAPPO in terms of total weighted travel time in both scenarios and have quite large standard errors for evaluation results which means that the performance of MAA2C and MADQN have large variance. Overall, MAPPO performs best with the lowest total weighted travel time and variance among all the deep reinforcement learning methods in Case 3.

4.5.3 Discussion

By reviewing all the results, it can be seen that A2C, MAA2C, MAPPO are the deep reinforcement learning methods with the best training performance, evaluation performance, and lowest variance in the situations of single traffic light, 2*2 traffic lights, and 3*2 traffic lights respectively. A2C achieves the best traffic light controls on total travel time of each traffic mode in Case 1. Even though MAA2C and MAPPO perform slightly worse than max-pressure in terms of total travel time of max-pressure in the evaluation in Case 2 and 3 respectively, they have a much better performance in total weighted travel time and total travel time of passenger cars, which proves their best general performance under the premise of considering both traffic efficiency and transit priority.

Both single-agent deep reinforcement methods PPO and A2C perform well in the training and evaluation in the situation of single traffic light while A2C performs slightly better than PPO. The reason can be that the clipping mechanism of PPO to prevent large updates might inadvertently cause it to overfit to suboptimal policies in simple environments while less constrained updates of A2C allow it to explore and generalize better in straightforward tasks. Therefore, both PPO and A2C are recommended for traffic light control optimization in the road network of single traffic light in multi-modal simulation.

Even though single-agent deep reinforcement learning methods PPO and A2C perform well in the situation of single traffic light, they struggle to achieve a good performance in the situations of multiple traffic lights. This is due to the fact that the single-agent DRL integrates the states of each traffic light to form a large state set and the formed large state set can not directly show what states belong to which traffic light, which makes the training process more difficult with the increase in the number of traffic lights. Therefore, it is recommended to use multi-agent deep reinforcement learning methods instead of single-agent deep reinforcement learning methods in the situation of multiple traffic lights in multi-modal simulation to avoid too large size of state set and help the better learning process for DRL algorithms.

As for value-based deep reinforcement learning method MADQN, it performs worse than policy-based deep reinforcement learning methods MAPPO and MAA2C in both Case 2 and 3. The reason for this is probably that policy-based deep reinforcement learning methods MAPPO and MAA2C directly learn a policy, which is a probability distribution over actions, and generate actions from it, allowing them to explore different strategies and action combinations more effectively for better discovery of optimal policies in environments where multiple agents interact while the value-based deep reinforcement learning method MADQN focuses on estimating the Q-value function for each possible action and optimizing it, which can lead to overfitting to specific action values, potentially limiting exploration. Therefore, it is advisable to use policy-based deep reinforcement learning methods for traffic light control optimization in multi-modal simulation.

As for policy-based deep reinforcement learning methods, MAA2C performs better than MAPPO in the road network of small sizes but performs worse in the road network of large sizes. This is probably due to the reason that the advantage-based updates in MAA2C can lead to more aggressive policy changes, which might be beneficial in simpler, less crowded environments. However, in more complex environments with more agents, this aggressive updating can lead to poor performance due to over-exploitation, which is adverse to the coordination between agents. The conservative updates of MAPPO based on the clipping mechanism encourage more consistent exploration across episodes. This becomes an advantage in scenarios with more agents, where the environment is more complex, and the need for consistent policy updates is crucial to maintain performance. Therefore, it is recommended to use MAA2C in road networks of multiple traffic lights with small sizes and use MAPPO in road networks with large sizes for more optimized traffic light control.

5 Conclusions

5.1 Conclusions

In conclusion, the research applied both single-agent deep reinforcement learning methods PPO, A2C and multi-agent deep reinforcement learning methods MAPPO, MAA2C, MADQN for traffic light control optimization in multi-modal simulation. Experiments are conducted in 3 different cases, which are set in road networks of different sizes respectively. And all the deep reinforcement methods used are evaluated in terms of training process, and model evaluation.

The results of the research show that A2C, MAA2C, and MAPPO perform best in road networks of single traffic light, 2*2 traffic lights, and 3*2 traffic lights respectively. Both PPO and A2C are recommended for traffic light optimization in the road network of single traffic light in multi-modal simulation. But for road networks of multiple traffic lights, it is recommended to use multi-agent deep reinforcement learning methods instead of single-agent deep reinforcement learning methods in multi-modal simulation to avoid too large size of state set and help the better learning process for DRL algorithms. And it is advisable to use policy-based deep reinforcement learning methods for traffic light control optimization in multi-modal simulation. As for road networks of multiple traffic lights, it is recommended to use MAA2C in road networks of multiple traffic lights of small sizes and use MAPPO in road networks of multiple traffic lights of large sizes for more optimized traffic light control.

The innovations of the research are mainly reflected in the following three points.

1. The research innovatively considers both traffic efficiency and transit priority for traffic light control optimization by using the way of assigning a weight for each traffic mode in the computations of queue length, traffic volume, and travel time for states and rewards in the application of deep reinforcement learning methods for traffic light control optimization in multi-modal simulation, which can guide people to choose more environmentally friendly traffic modes and ultimately achieve the goal of forming a more sustainable transportation system in the urban area.

2. The research explores the use and effectiveness of using different single-agent deep reinforcement learning methods to control multiple traffic lights in multi-modal simulation.

3. The research evaluates different deep reinforcement learning methods in terms of different aspects including training curves, performance of each traffic mode, performance of model evaluation, and performance in road networks of different sizes under the topic of the application of different deep reinforcement learning methods for traffic light control optimization in multi-modal simulation.

In summary, this research discovers the use and effects of different deep reinforcement learning methods in multi-modal simulation, evaluates their performance from multiple aspects, and paves the way for developing efficient DRL agents that consider the needs of different road users and prioritize public transport for the application of DRL in multi-modal simulation, which is of great significance for achieving more optimized traffic light control that considers multiple factors and realizing a more efficient and sustainable urban transportation system.

However, there are still some limitations existing in this research as shown below:

1. Only two traffic modes, passenger cars and buses, are considered.
   In modern urban traffic, the participation and interaction of multiple traffic modes have become the norm. Therefore, covering more traffic modes in multimodal simulation is of great significance for the optimal control of traffic lights in complex scenarios.

2. The action is only limited to selecting the green phase without involving the corresponding green phase duration.
   In the current framework, the duration for a green phase is just a multiple of 5 seconds, which is less flexible than also considering the corresponding green phase duration in the action selection directly.

3. Only the travel time is considered as the reward and performance indicator.
   Actually, the control of urban traffic is a large and complicated task considering plenty of factors like efficiency, safety, energy consumption, level of public transport service, sustainability, and so on. A more comprehensive reward design and evaluation can be of great significance to optimizing the DRL-based traffic light control in multi-modal simulation for achieving a balance of multiple goals.

4. The road network used is simpler than the real road network.
   The road network used in this research is simple, which may not adequately reflect the complexity of real road network scenarios.

5.2 Recommendations for future work

Based on the limitations mentioned in Section 5.1, further research regarding the application of DRL algorithms in multi-modal simulation of SUMO can focus on the following several points:

1. **Incorporate more traffic modes.**
   This can enhance the realism and complexity of the simulated environment, better approximating real-world scenarios with diverse traffic compositions, and enable the exploration of complex interactions and dependencies between different traffic modes.

2. **Use both the green phase and the corresponding green phase duration as the action choice.**
   The application of both kinds of actions can achieve more accurate traffic light control. But this has to be considered carefully as this could increase the difficulty for the learning process of DRL algorithms.

3. **Consider multiple factors in reward design.**
   A more comprehensive reward function allows for promoting the system performance from a more holistic perspective.

4. **Consider multiple factors in model evaluation.**
   Evaluating trained models based on a variety of performance metrics provides a more robust assessment of their capabilities, identifying potential trade-offs and enabling the selection of agents that excel in specific operational conditions or prioritize particular performance criteria.

5. **Use a part of a real road network as the case study.**
   This provides a more realistic and complex environment compared to synthetic networks, allowing for the validation of algorithm performance under realistic traffic conditions. This also enables the direct comparison of DRL-based policies with existing traffic management systems, providing insights into the potential benefits and challenges of implementing such algorithms in real-world applications.

References

[1] Richard E Allsop. "Some possibilities for using traffic control to influence trip distribution and route choice". In: *Transportation and traffic theory, proceedings*. Vol. 6. 1974.

[2] Gaurav Bagwe et al. "RAMRL: Towards Robust On-Ramp Merging via Augmented Multimodal Reinforcement Learning". In: *2023 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*. 2023, pp. 23–33. DOI: 10.1109/MOST57249.2023.00011.

[3] Salah Bouktif, Abderraouf Cheniki, and Ali Ouni. "Traffic Signal Control Using Hybrid Action Space Deep Reinforcement Learning". In: *Sensors* 21.7 (2021). ISSN: 1424-8220. DOI: 10.3390/s21072302. URL: https://www.mdpi.com/1424-8220/21/7/2302.

[4] Peng Chen, Zemao Zhu, and Guangquan Lu. "An Adaptive Control Method for Arterial Signal Coordination Based on Deep Reinforcement Learning". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 3553–3558. DOI: 10.1109/ITSC.2019.8917051.

[5] Lara Codeca and Vinny Cahill. "Using Deep Reinforcement Learning to Coordinate Multi-Modal Journey Planning with Limited Transportation Capacity". In: *SUMO Conference Proceedings* 2 (June 2022), pp. 13–32. DOI: 10.52825/scp.v2i.89. URL: https://www.tib-op.org/ojs/index.php/scp/article/view/89.

[6] Prafulla Dhariwal et al. *OpenAI Baselines*. https://github.com/openai/baselines. 2017.

[7] Guhao Feng and Han Zhong. *Rethinking Model-based, Policy-based, and Value-based Reinforcement Learning via the Lens of Representation Complexity*. 2023. arXiv: 2312.17248 [cs.LG].

[8] M Ebrahim Fouladvand, Zeinab Sadjadi, and M Reza Shaebani. "Optimized traffic flow at a single intersection: traffic responsive signalization". In: *Journal of Physics A: Mathematical and General* 37.3 (Jan. 2004), p. 561. DOI: 10.1088/0305-4470/37/3/002. URL: https://dx.doi.org/10.1088/0305-4470/37/3/002.

[9] Wade Genders and Saiedeh Razavi. *An Open-Source Framework for Adaptive Traffic Signal Control*. 2019. DOI: 10.48550/arXiv.1909.00395. arXiv: 1909.00395 [eess.SY]. URL: https://arxiv.org/abs/1909.00395.

[10] Wade Genders and Saiedeh N. Razavi. "Using a Deep Reinforcement Learning Agent for Traffic Signal Control". In: *CoRR* abs/1611.01142 (2016). DOI: 10.48550/arXiv.1611.01142. arXiv: 1611.01142. URL: http://arxiv.org/abs/1611.01142.

[11] Mengyu Guo et al. "A Reinforcement Learning Approach for Intelligent Traffic Signal Control at Urban Intersections". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 4242–4247. DOI: 10.1109/ITSC.2019.8917268.

[12] Guangjie Han et al. "Deep Reinforcement Learning for Intersection Signal Control Considering Pedestrian Behavior". In: *Electronics* 11.21 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11213519. URL: https://www.mdpi.com/2079-9292/11/21/3519.

[13] Qing He, K. Larry Head, and Jun Ding. "Multi-modal traffic signal control with priority, signal actuation and coordination". English (US). In: *Transportation Research Part C: Emerging Technologies* 46 (Sept. 2014), pp. 65–82. ISSN: 0968-090X. DOI: 10.1016/j.trc.2014.05.001.

[14] Xingshuai Huang, Di Wu, and Benoit Boulet. *Traffic Signal Control Using Lightweight Transformers: An Offline-to-Online RL Approach*. 2023. DOI: 10.48550/arXiv.2312.07795. arXiv: 2312.07795 [cs.LG]. URL: https://arxiv.org/abs/2312.07795.

[15] Songsang Koh et al. "Real-time deep reinforcement learning based vehicle navigation". In: *Applied Soft Computing* 96 (2020), p. 106694. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2020.106694. URL: https://www.sciencedirect.com/science/article/pii/S1568494620306323.

[16] Neetesh Kumar et al. "Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System". In: *IEEE Transactions on Intelligent Transportation Systems* 23.3 (2022), pp. 2411–2421. DOI: 10.1109/TITS.2021.3095161.

[17] Chenghao Li et al. "Fairness Control of Traffic Light via Deep Reinforcement Learning". In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. 2020, pp. 652–658. DOI: 10.1109/CASE48305.2020.9216899.

[18] Duowei Li et al. "Adaptive Traffic Signal Control Model on Intersections Based on Deep Reinforcement Learning". In: *Journal of Advanced Transportation* 2020.1 (2020), p. 6505893. DOI: https://doi.org/10.1155/2020/6505893. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2020/6505893. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/6505893.

[19] Jianming Li et al. "Research on signal control method of deep reinforcement learning based on value distribution". In: *Journal of Physics: Conference Series* 2330.1 (Aug. 2022), p. 012019. DOI: 10.1088/1742-6596/2330/1/012019. URL: https://dx.doi.org/10.1088/1742-6596/2330/1/012019.

[20] Meng Li et al. "Deep Reinforcement Learning-Based Vehicle Driving Strategy to Reduce Crash Risks in Traffic Oscillations". In: *Transportation Research Record* 2674.10 (2020), pp. 42–54. DOI: 10.1177/0361198120937976. eprint: https://doi.org/10.1177/0361198120937976. URL: https://doi.org/10.1177/0361198120937976.

[21] Zhenning Li, Chengzhong Xu, and Guohui Zhang. *A Deep Reinforcement Learning Approach for Traffic Signal Control Optimization*. 2021. DOI: 10.48550/arXiv.2107.06115. arXiv: 2107.06115 [eess.SP]. URL: https://arxiv.org/abs/2107.06115.

[22] Zhenning Li et al. "Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning". In: *Transportation Research Part C: Emerging Technologies* 125 (2021), p. 103059. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2021.103059. URL: https://www.sciencedirect.com/science/article/pii/S0968090X21000851.

[23] Xiaoyuan Liang et al. "Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks". In: *CoRR* abs/1803.11115 (2018). DOI: 10.48550/arXiv.1803.11115. arXiv: 1803.11115. URL: http://arxiv.org/abs/1803.11115.

[24] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL: https://elib.dlr.de/124092/.

[25] Cobus Louw, Louwrens Labuschagne, and Tiffany Woodley. "A Comparison of Reinforcement Learning Agents Applied to Traffic Signal Optimisation". In: *SUMO Conference Proceedings* 3 (Sept. 2022), pp. 15–43. DOI: 10.52825/scp.v3i.116. URL: https://www.tib-op.org/ojs/index.php/scp/article/view/116.

[26] Ryan Lowe et al. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf.

[27] Zibo Ma et al. "Adaptive optimization of traffic signal timing via deep reinforcement learning". In: *Journal of Advanced Transportation* 2021 (2021), pp. 1–14. DOI: 10.1155/2021/6616702.

[28] Konstantinos Makantasis, Maria Kontorinaki, and Ioannis Nikolos. "Deep reinforcement-learning-based driving policy for autonomous road vehicles". In: *IET Intelligent Transport Systems* 14.1 (2020), pp. 13–24. DOI: 10.1049/iet-its.2019.0249. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2019.0249. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2019.0249.

[29] Xutao Mei et al. "Reinforcement Learning-Based Intelligent Traffic Signal Control Considering Sensing Information of Railway". In: *IEEE Sensors Journal* 23.24 (2023), pp. 31125–31136. DOI: 10.1109/JSEN.2023.3327696.

[30] Volodymyr Mnih et al. "Asynchronous Methods for Deep Reinforcement Learning". In: *CoRR* abs/1602.01783 (2016). arXiv: 1602.01783. URL: http://arxiv.org/abs/1602.01783.

[31] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533. DOI: 10.1038/nature14236.

[32] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[33] NTM. *Passenger Occupancy*. 2024. URL: https://www.transportmeasures.org/en/wiki/manuals/11-road-passenger-bus-transport/11-8-passenger-occupancy/.

[34] OpenAI. *OpenAI Baselines: ACKTR & A2C*. 2017. URL: https://openai.com/index/openai-baselines-acktr-a2c/.

[35] Martin L. Puterman. "Chapter 8 Markov decision processes". In: *Stochastic Models*. Vol. 2. Handbooks in Operations Research and Management Science. Elsevier, 1990, pp. 331–434. DOI: 10.1016/S0927-0507(05)80172-0. URL: https://www.sciencedirect.com/science/article/pii/S0927050705801720.

[36] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[37] Kazi Redwan Shabab, Syed Mostaquim Ali, and Mohamed H Zaki. "Deep reinforcement learning-based short-term traffic signal optimizing using disaggregated vehicle data". In: *Data science for transportation* 5.2 (2023), p. 13. DOI: 10.1007/s42421-023-00074-x.

[38] Wenchao Shen et al. "A Bus Signal Priority Control Method Based on Deep Reinforcement Learning". In: *Applied Sciences* 13.11 (2023). ISSN: 2076-3417. DOI: 10.3390/app13116772. URL: https://www.mdpi.com/2076-3417/13/11/6772.

[39] Tianyu Shi et al. *Improving the generalizability and robustness of large-scale traffic signal control*. 2023. DOI: 10.48550/arXiv.2306.01925. arXiv: 2306.01925 [cs.LG]. URL: https://arxiv.org/abs/2306.01925.

[40] Swapnil Manohar Shinde. "Adaptive traffic light control system". In: *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*. 2017, pp. 300–306. DOI: 10.1109/ICISIM.2017.8122189.

[41] László Szőke et al. "Driving on Highway by Using Reinforcement Learning with CNN and LSTM Networks". In: *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*. 2020, pp. 121–126. DOI: 10.1109/INES49302.2020.9147185.

[42] Pravin Varaiya. "Max pressure control of a network of signalized intersections". In: *Transportation Research Part C: Emerging Technologies* 36 (2013), pp. 177–195. ISSN: 0968-090X. DOI: 10.1016/j.trc.2013.08.014. URL: https://www.sciencedirect.com/science/article/pii/S0968090X13001782.

[43] Bin Wang et al. "Deep Reinforcement Learning for Traffic Light Timing Optimization". In: *Processes* 10.11 (2022). ISSN: 2227-9717. DOI: 10.3390/pr10112458. URL: https://www.mdpi.com/2227-9717/10/11/2458.

[44] Chong Wang et al. "Integrated Traffic Control for Freeway Recurrent Bottleneck Based on Deep Reinforcement Learning". In: *IEEE Transactions on Intelligent Transportation Systems* 23.9 (2022), pp. 15522–15535. DOI: 10.1109/TITS.2022.3141730.

[45] Shijie Wang and Shangbo Wang. *A Novel Multi-Agent Deep RL Approach for Traffic Signal Control*. 2023. DOI: 10.48550/arXiv.2306.02684. arXiv: 2306.02684 [cs.AI]. URL: https://arxiv.org/abs/2306.02684.

[46] Xingmin Wang et al. "Learning the max pressure control for urban traffic networks considering the phase switching loss". In: *Transportation Research Part C: Emerging Technologies* 140 (2022), p. 103670. ISSN: 0968-090X. DOI: 10.1016/j.trc.2022.103670. URL: https://www.sciencedirect.com/science/article/pii/S0968090X22001139.

[47] Hua Wei et al. "PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1290–1298. ISBN: 9781450362016. DOI: 10.1145/3292500.3330949. URL: https://doi.org/10.1145/3292500.3330949.

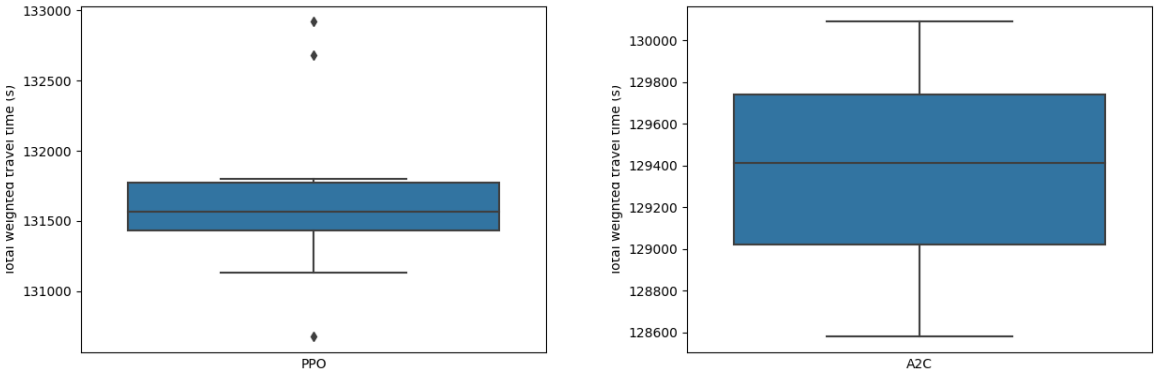
[48] Jiajie Yu et al. "Decentralized signal control for multi-modal traffic network: A deep reinforcement learning approach". In: *Transportation Research Part C: Emerging Technologies* 154 (2023), p. 104281. ISSN: 0968-090X. DOI: 10.1016/j.trc.2023.104281. URL: https://www.sciencedirect.com/science/article/pii/S0968090X2300270X.

[49] Junwu Zhao, Ting Qu, and Fang Xu. "A Deep Reinforcement Learning Approach for Autonomous Highway Driving". In: *IFAC-PapersOnLine* 53.5 (2020). 3rd IFAC Workshop on Cyber-Physical and Human Systems CPHS 2020, pp. 542–546. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2021.04.142. URL: https://www.sciencedirect.com/science/article/pii/S240589632100272X.

[50] Nan Zhong, Kaifeng Liu, Yurong Li, et al. "Deep Q-Learning Network Model for Optimizing Transit Bus Priority at Multiphase Traffic Signal Controlled Intersection". In: *Mathematical Problems in Engineering* 2023 (2023). DOI: 10.1155/2023/9137889.

B

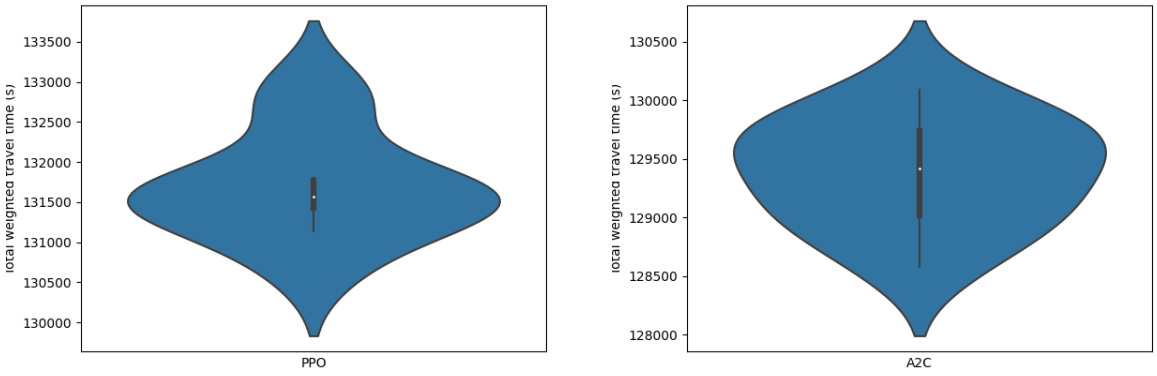
Boxplots and Violinplots of evaluation results



(a) Box plot of total weighted travel time for PPO

(b) Box plot of total weighted travel time for A2C

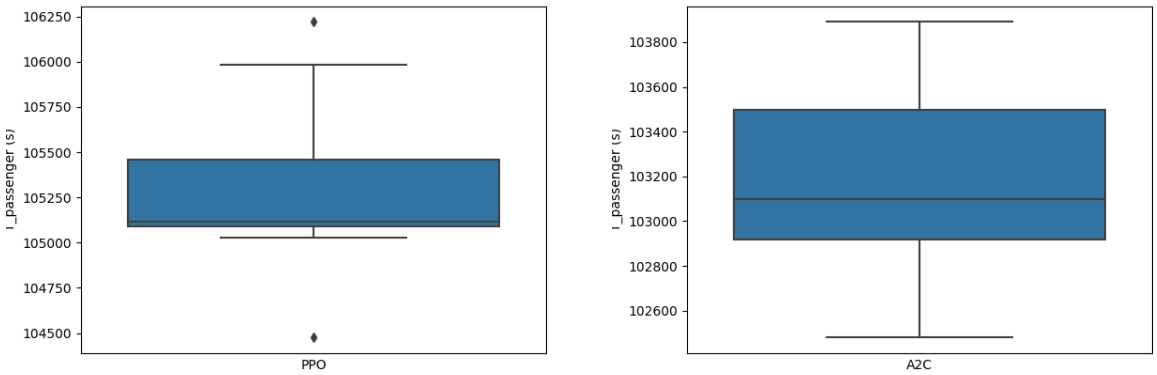
Figure B.1: Box plots of total weighted travel time with 10% less demand of Case 1



(a) Violin plot of total weighted travel time for PPO

(b) Violin plot of total weighted travel time for A2C

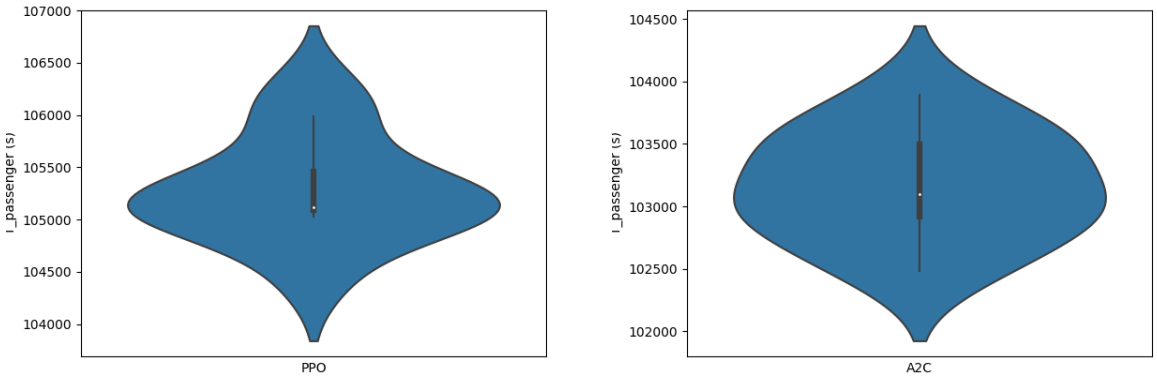
Figure B.2: Violin plots of total weighted travel time with 10% less demand of Case 1



(a) Box plot of $T_{passenger}$ for PPO

(b) Box plot of $T_{passenger}$ for A2C

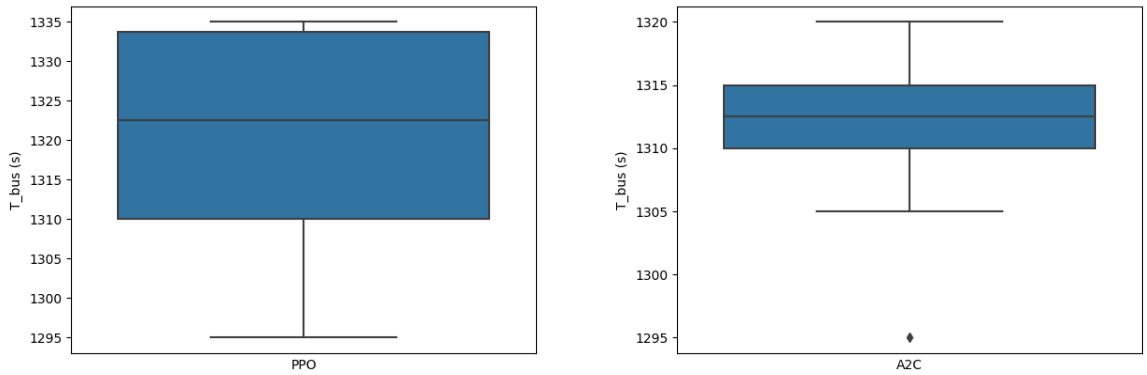
Figure B.3: Box plots of $T_{passenger}$ with 10% less demand of Case 1



(a) Violin plot of $T_{passenger}$ for PPO

(b) Violin plot of $T_{passenger}$ for A2C

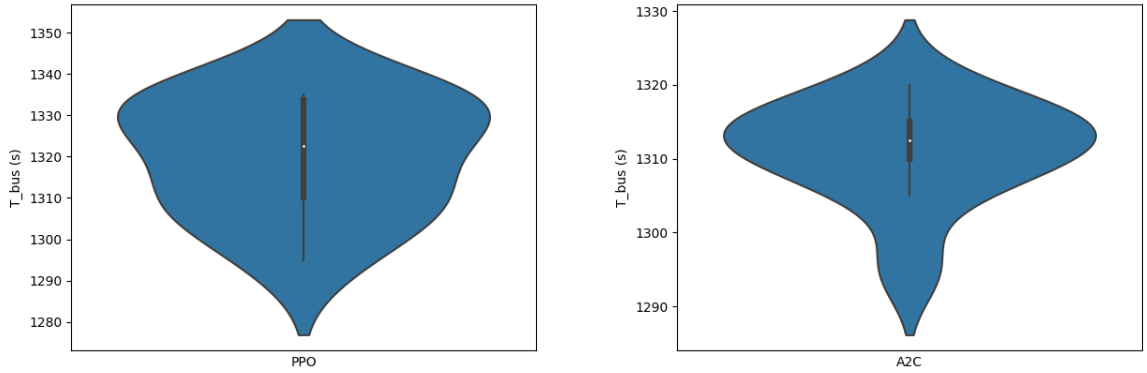
Figure B.4: Violin plots of $T_{passenger}$ with 10% less demand of Case 1



(a) Box plot of T_{bus} for PPO

(b) Box plot of T_{bus} for A2C

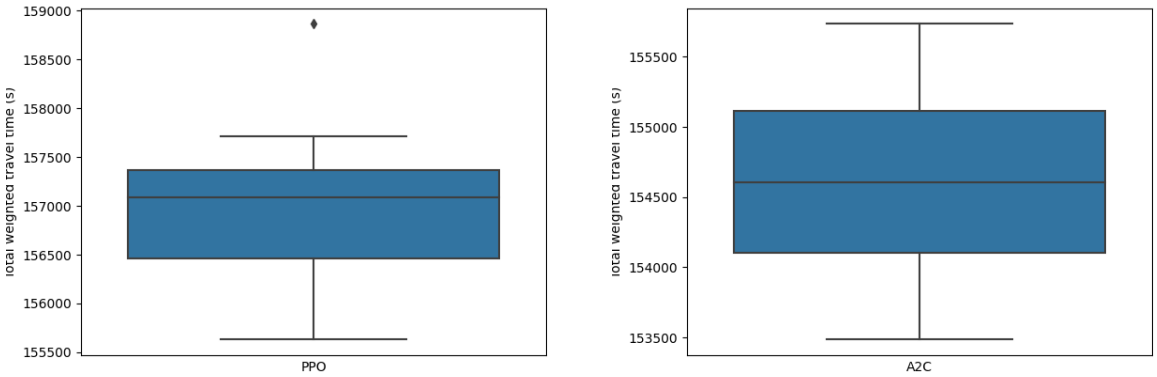
Figure B.5: Box plots of T_{bus} with 10% less demand of Case 1



(a) Violin plot of T_{bus} for PPO

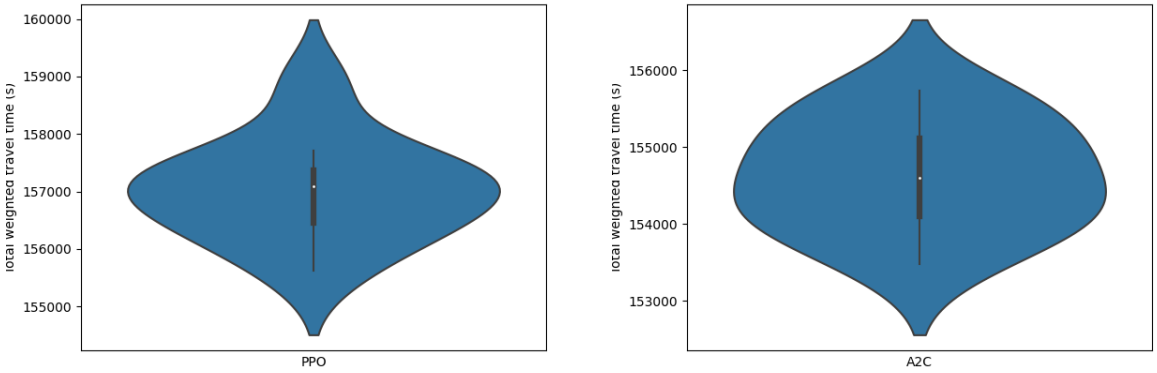
(b) Violin plot of T_{bus} for A2C

Figure B.6: Violin plots of T_{bus} with 10% less demand of Case 1



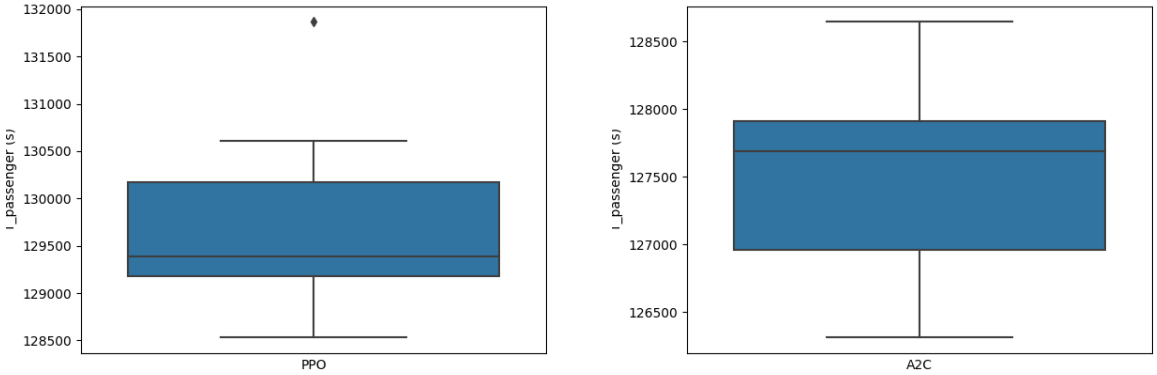
(a) Box plot of total weighted travel time for PPO (b) Box plot of total weighted travel time for A2C

Figure B.7: Box plots of total weighted travel time with 10% more demand of Case 1



(a) Violin plot of total weighted travel time for PPO (b) Violin plot of total weighted travel time for A2C

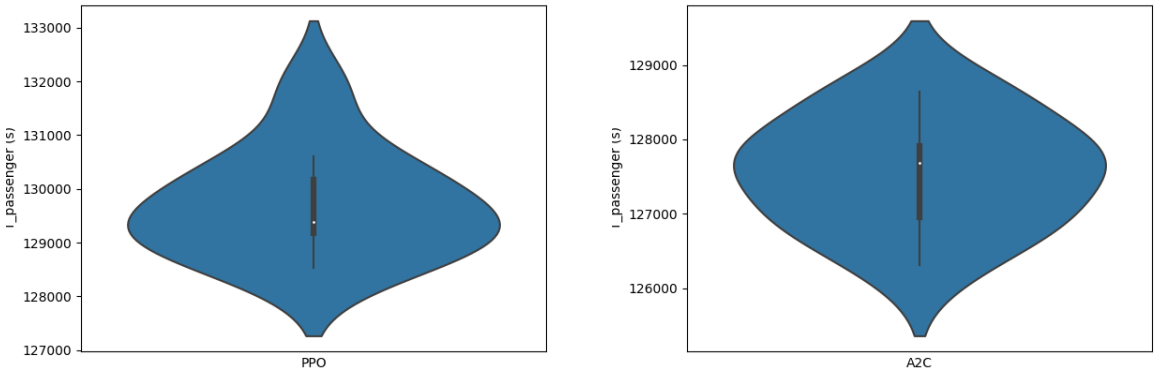
Figure B.8: Violin plots of total weighted travel time with 10% more demand of Case 1



(a) Box plot of $T_{passenger}$ for PPO

(b) Box plot of $T_{passenger}$ for A2C

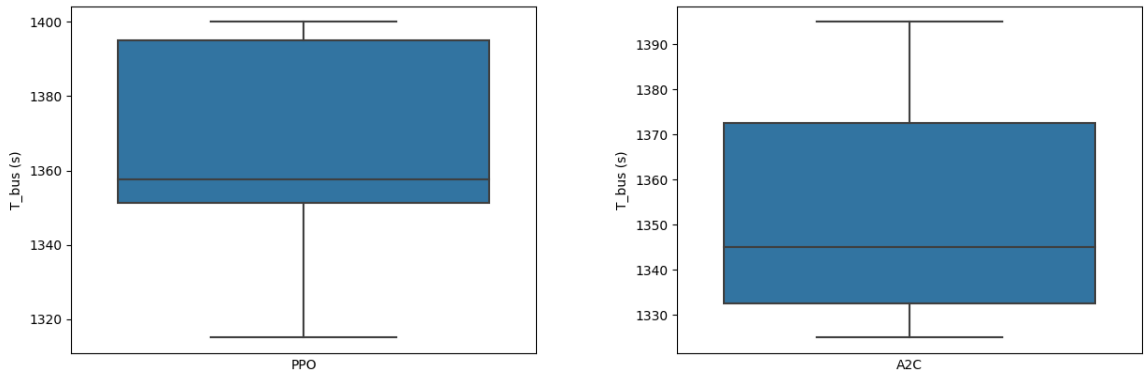
Figure B.9: Box plots of $T_{passenger}$ with 10% more demand of Case 1



(a) Violin plot of $T_{passenger}$ for PPO

(b) Violin plot of $T_{passenger}$ for A2C

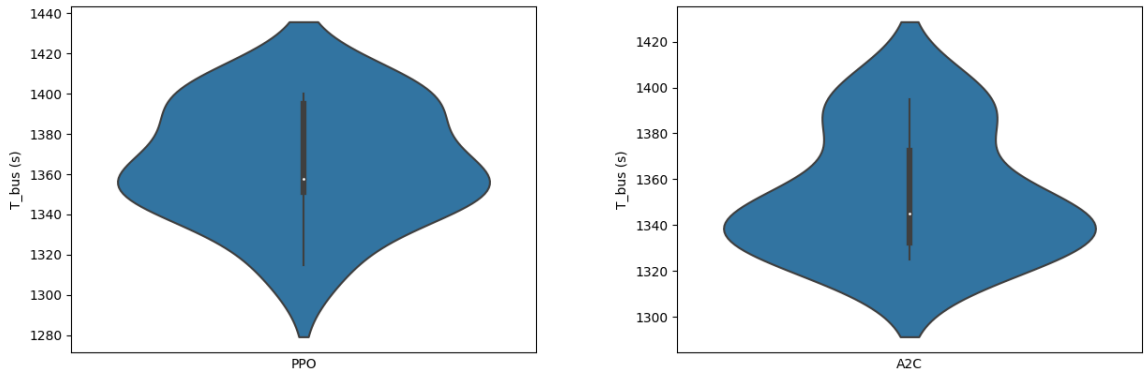
Figure B.10: Violin plots of $T_{passenger}$ with 10% more demand of Case 1



(a) Box plot of T_{bus} for PPO

(b) Box plot of T_{bus} for A2C

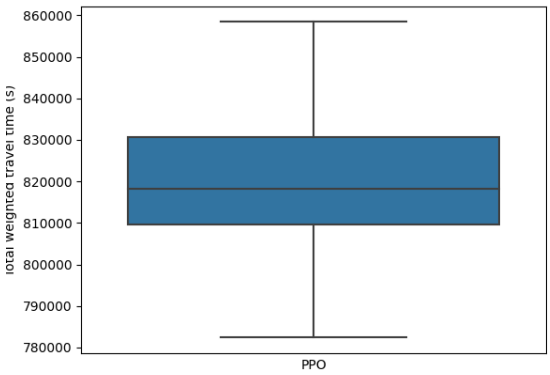
Figure B.11: Box plots of T_{bus} with 10% more demand of Case 1



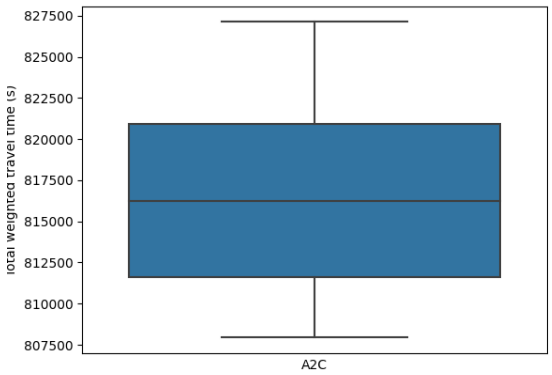
(a) Violin plot of T_{bus} for PPO

(b) Violin plot of T_{bus} for A2C

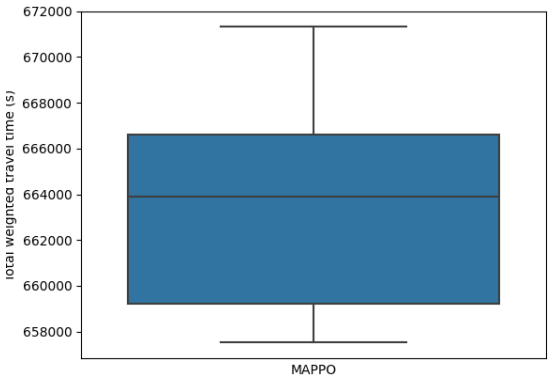
Figure B.12: Violin plots of T_{bus} with 10% more demand of Case 1



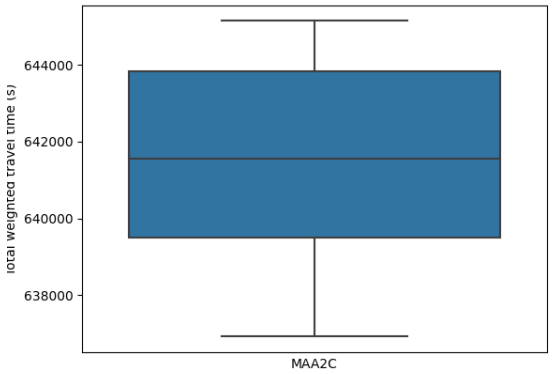
(a) Box plot of total weighted travel time for PPO



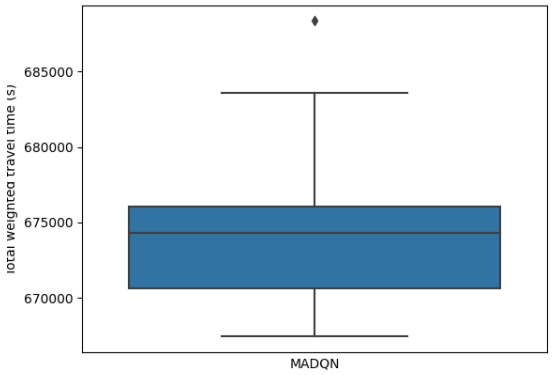
(b) Box plot of total weighted travel time for A2C



(c) Box plot of total weighted travel time for MAPPO

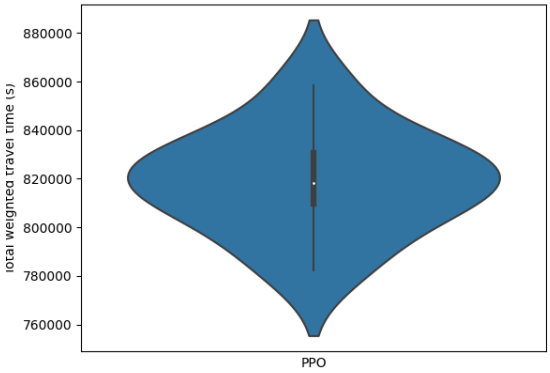


(d) Box plot of total weighted travel time for MAA2C

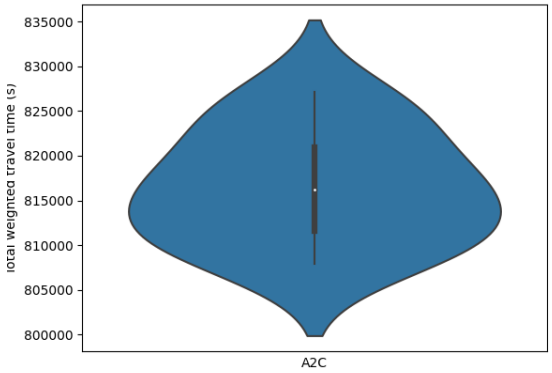


(e) Box plot of total weighted travel time for MADQN

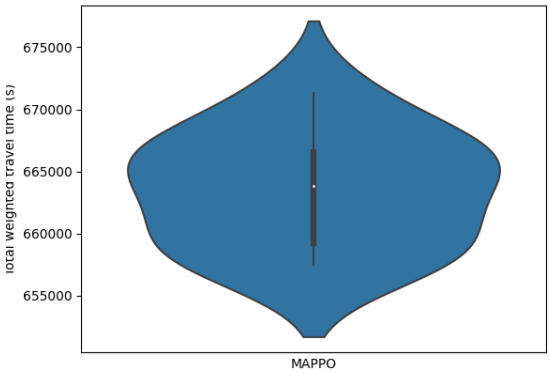
Figure B.13: Box plots of total weighted travel time with 10% less demand of Case 2



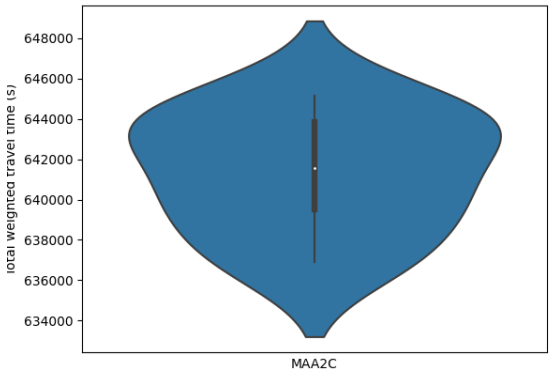
(a) Violin plot of total weighted travel time for PPO



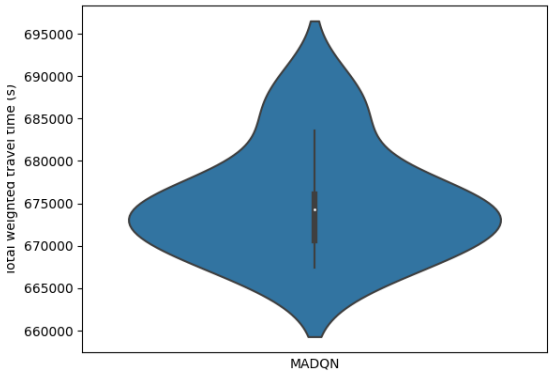
(b) Violin plot of total weighted travel time for A2C



(c) Violin plot of total weighted travel time for MAPPO

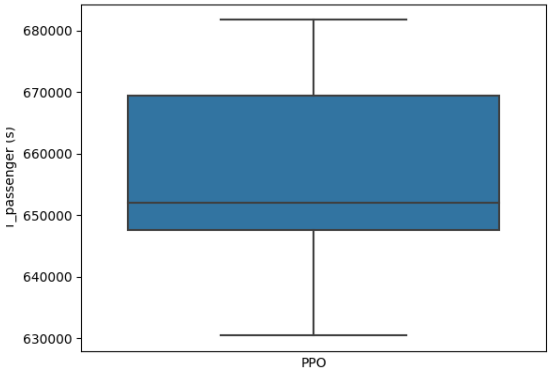


(d) Violin plot of total weighted travel time for MAA2C

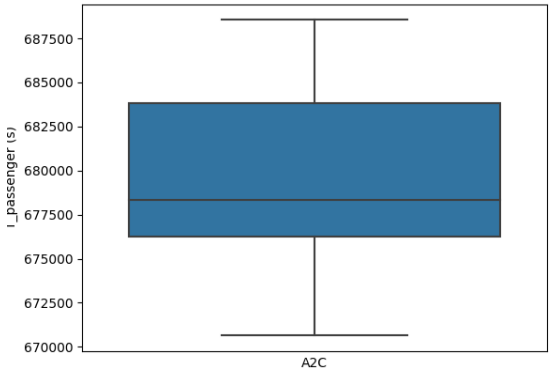


(e) Violin plot of total weighted travel time for MADQN

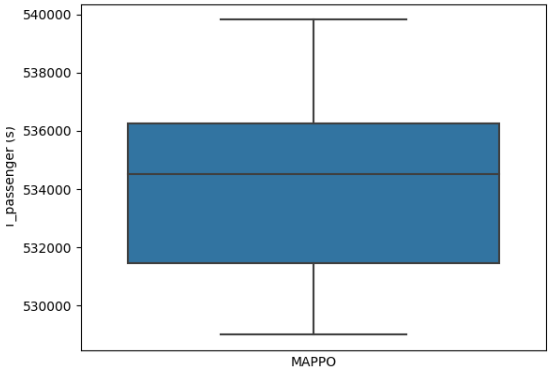
Figure B.14: Violin plots of total weighted travel time with 10% less demand of Case 2



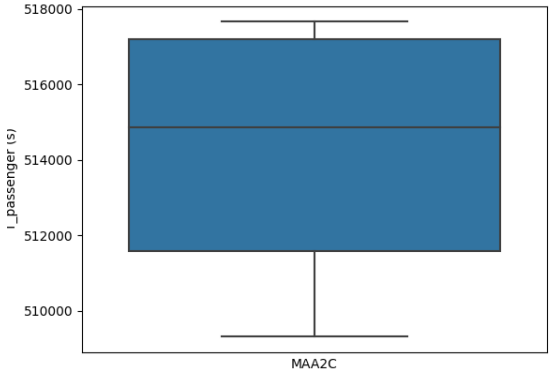
(a) Box plot of $T_{passenger}$ for PPO



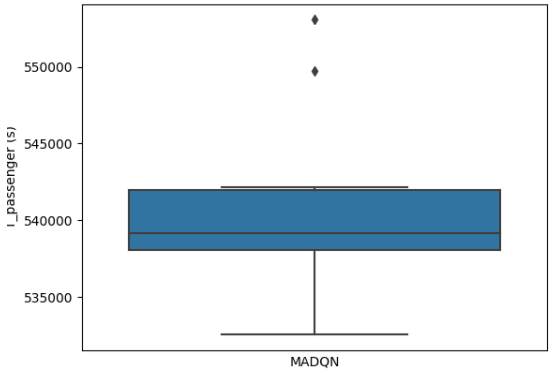
(b) Box plot of $T_{passenger}$ for A2C



(c) Box plot of $T_{passenger}$ for MAPPO

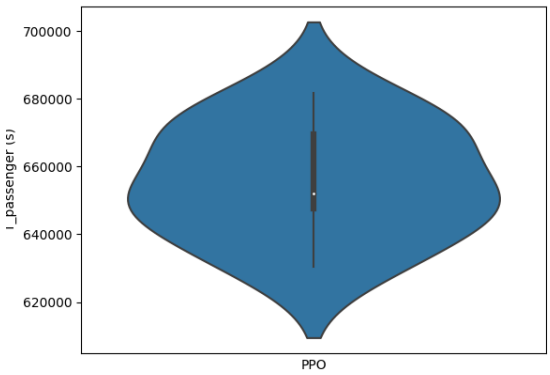


(d) Box plot of $T_{passenger}$ for MAA2C

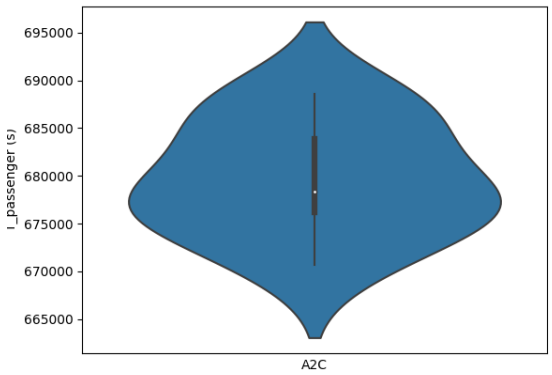


(e) Box plot of $T_{passenger}$ for MADQN

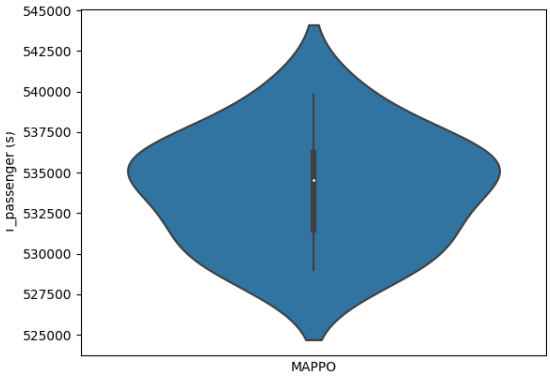
Figure B.15: Box plots of $T_{passenger}$ with 10% less demand of Case 2



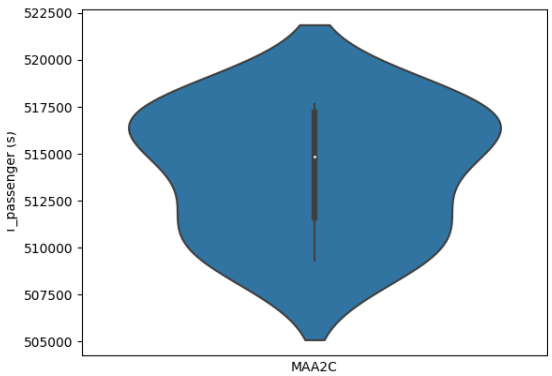
(a) Violin plot of $T_{passenger}$ for PPO



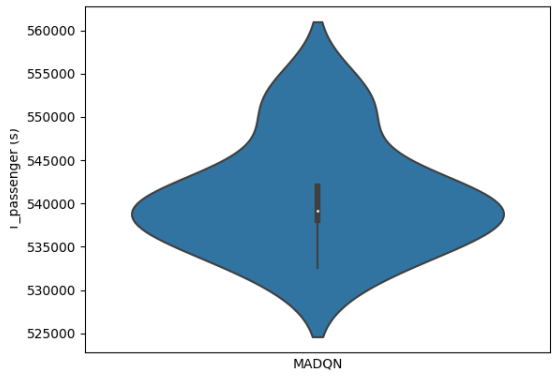
(b) Violin plot of $T_{passenger}$ for A2C



(c) Violin plot of $T_{passenger}$ for MAPPO

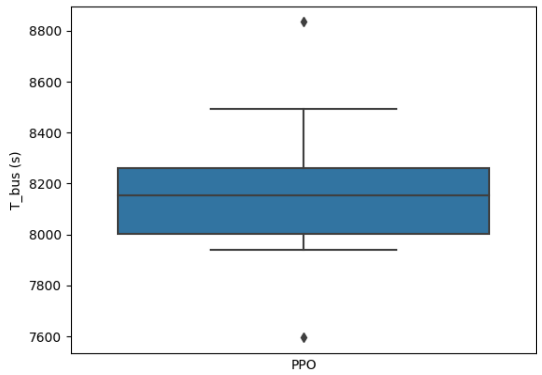


(d) Violin plot of $T_{passenger}$ for MAA2C

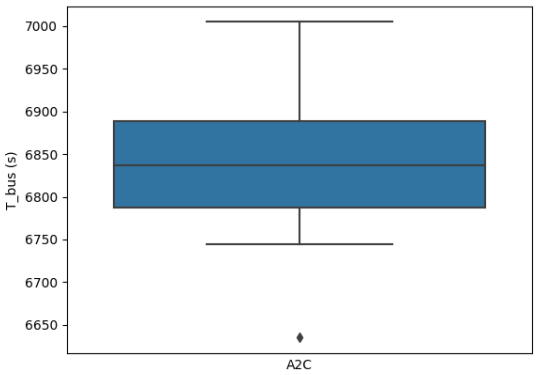


(e) Violin plot of $T_{passenger}$ for MADQN

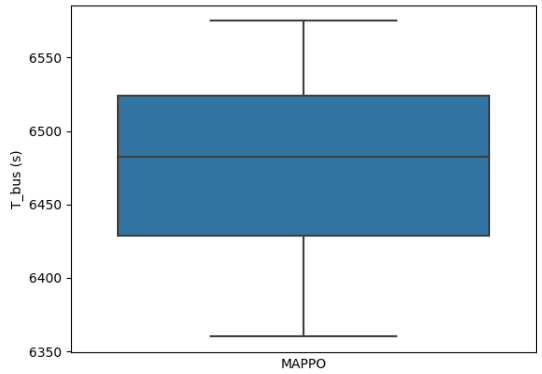
Figure B.16: Violin plots of $T_{passenger}$ with 10% less demand of Case 2



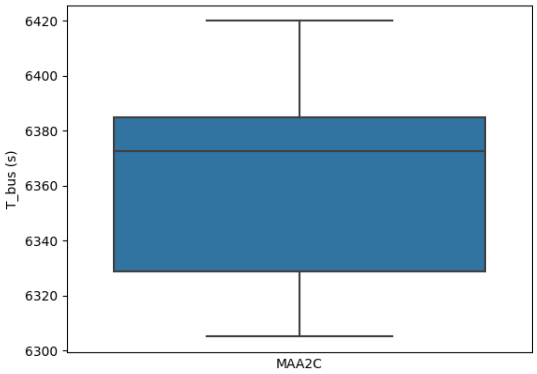
(a) Box plot of T_{bus} for PPO



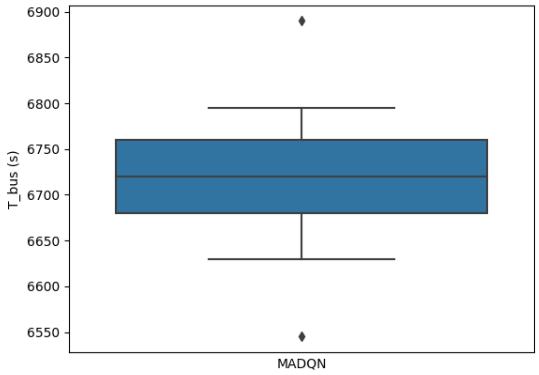
(b) Box plot of T_{bus} for A2C



(c) Box plot of T_{bus} for MAPPO

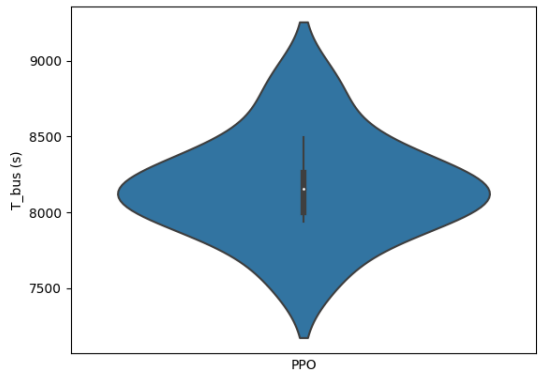


(d) Box plot of T_{bus} for MAA2C

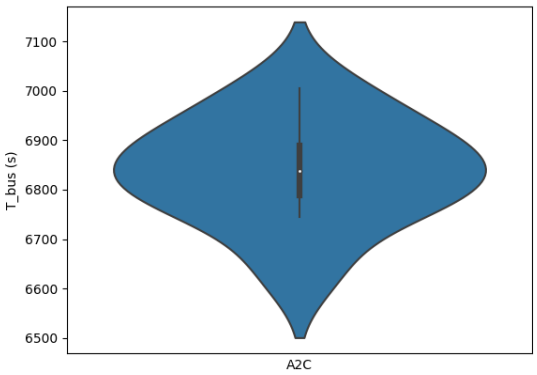


(e) Box plot of T_{bus} for MADQN

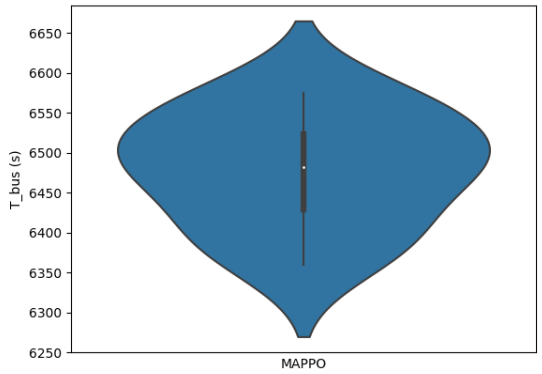
Figure B.17: Box plots of T_{bus} with 10% less demand of Case 2



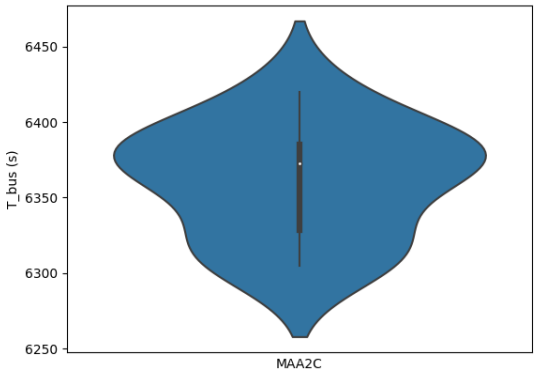
(a) Violin plot of T_{bus} for PPO



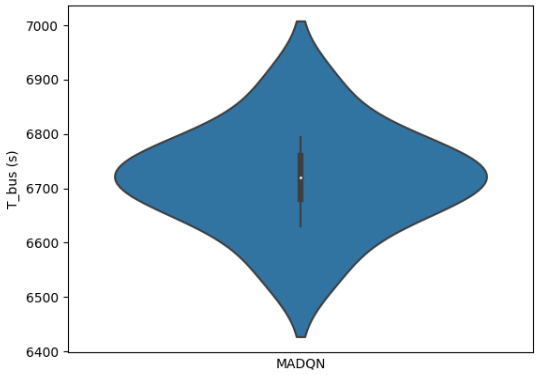
(b) Violin plot of T_{bus} for A2C



(c) Violin plot of T_{bus} for MAPPO

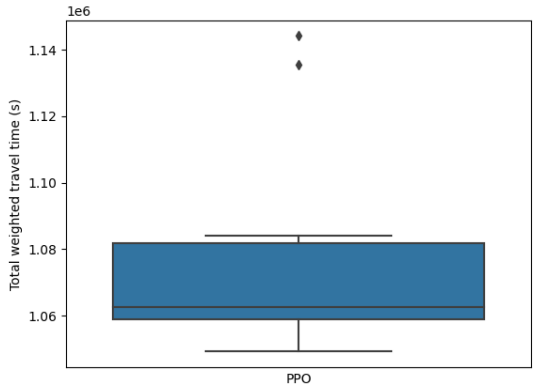


(d) Violin plot of T_{bus} for MAA2C

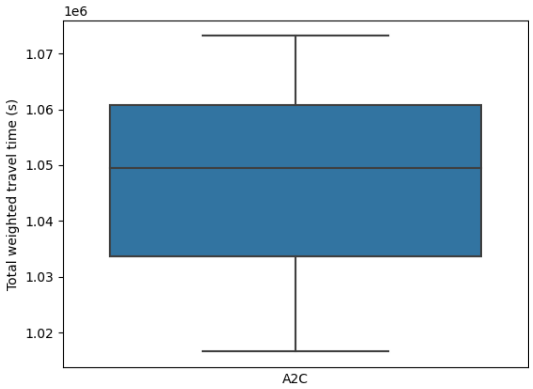


(e) Violin plot of T_{bus} for MADQN

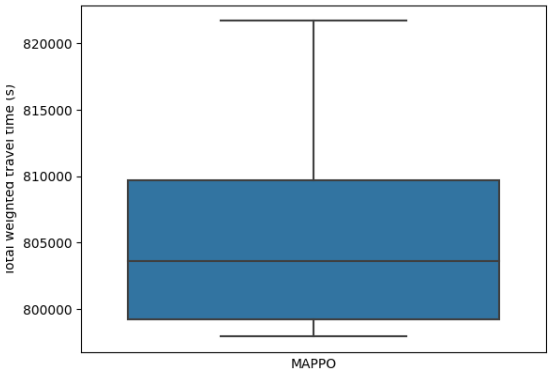
Figure B.18: Violin plots of T_{bus} with 10% less demand of Case 2



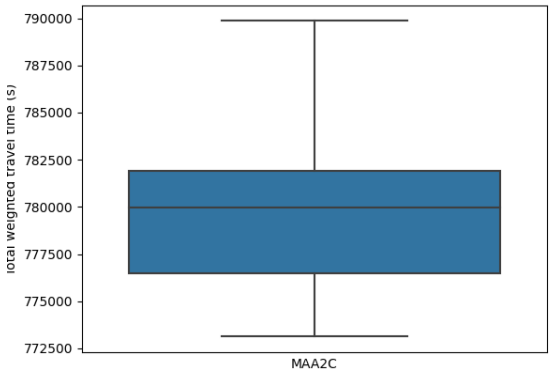
(a) Box plot of total weighted travel time for PPO



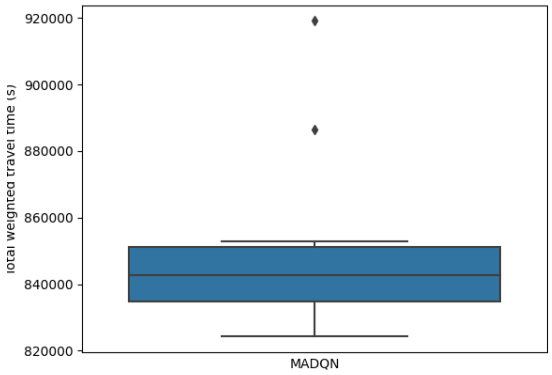
(b) Box plot of total weighted travel time for A2C



(c) Box plot of total weighted travel time for MAPPO

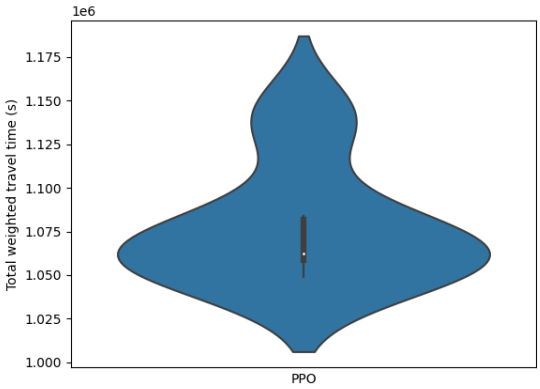


(d) Box plot of total weighted travel time for MAA2C

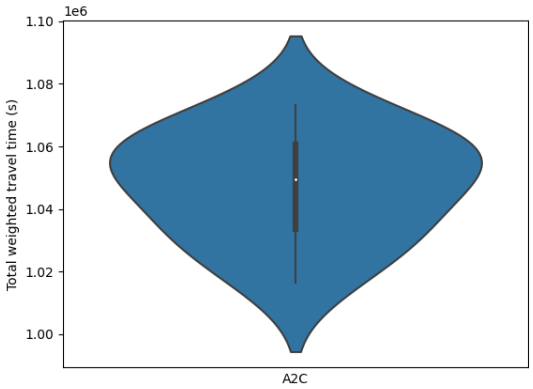


(e) Box plot of total weighted travel time for MADQN

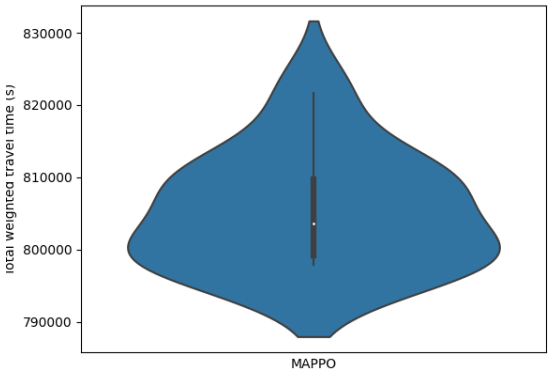
Figure B.19: Box plots of total weighted travel time with 10% more demand of Case 2



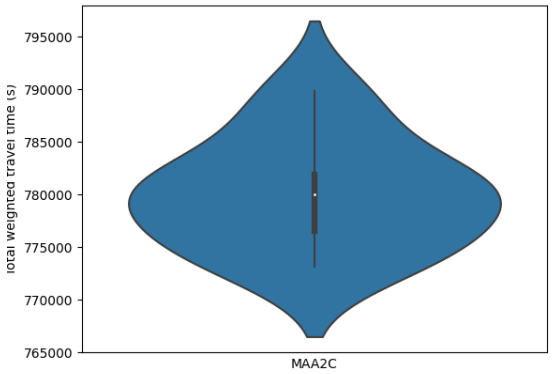
(a) Violin plot of total weighted travel time for PPO



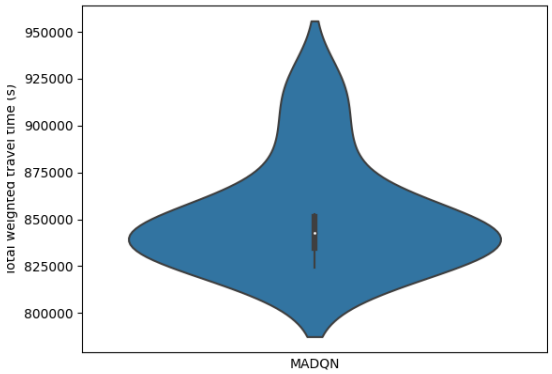
(b) Violin plot of total weighted travel time for A2C



(c) Violin plot of total weighted travel time for MAPPO

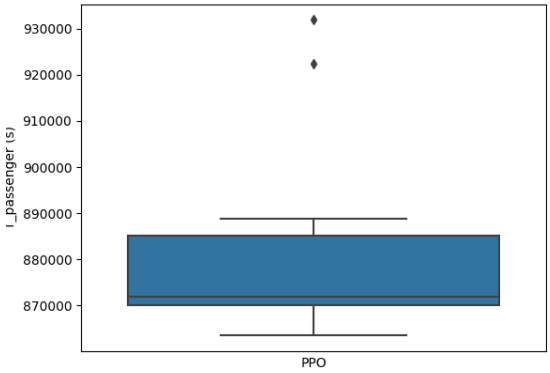


(d) Violin plot of total weighted travel time for MAA2C

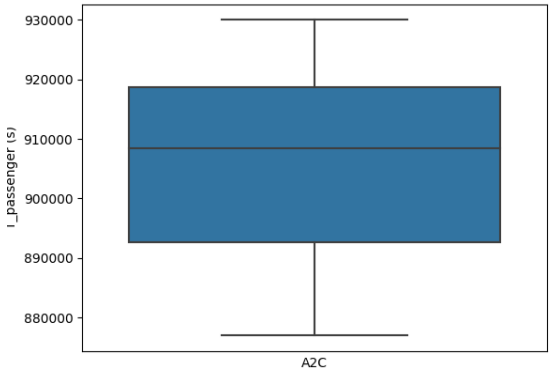


(e) Violin plot of total weighted travel time for MADQN

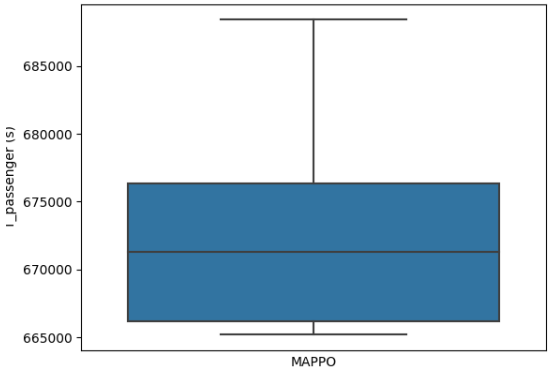
Figure B.20: Violin plots of total weighted travel time with 10% more demand of Case 2



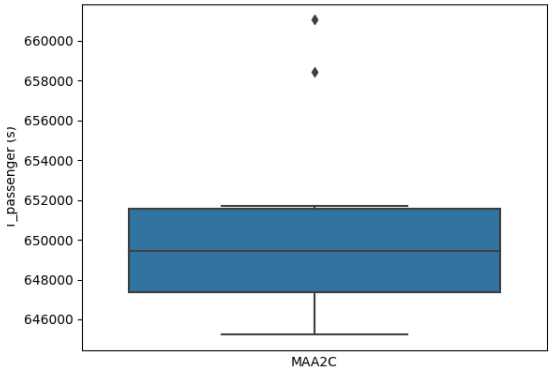
(a) Box plot of $T_{passenger}$ for PPO



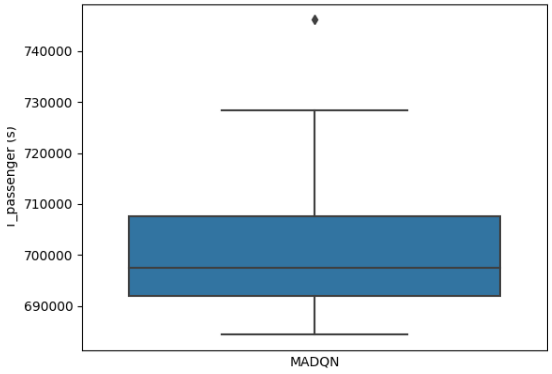
(b) Box plot of $T_{passenger}$ for A2C



(c) Box plot of $T_{passenger}$ for MAPPO

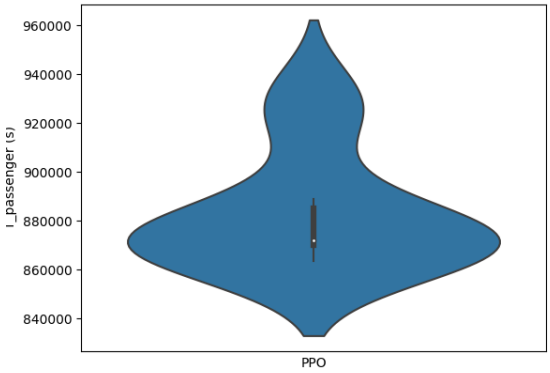


(d) Box plot of $T_{passenger}$ for MAA2C

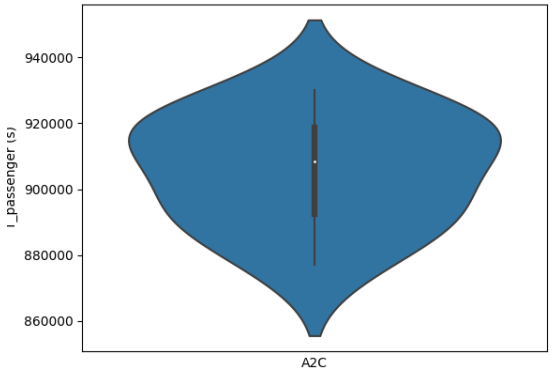


(e) Box plot of $T_{passenger}$ for MADQN

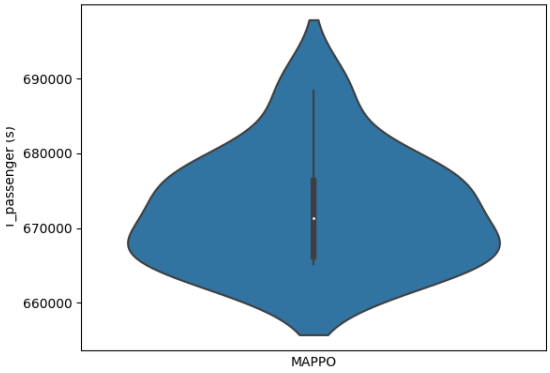
Figure B.21: Box plots of $T_{passenger}$ with 10% more demand of Case 2



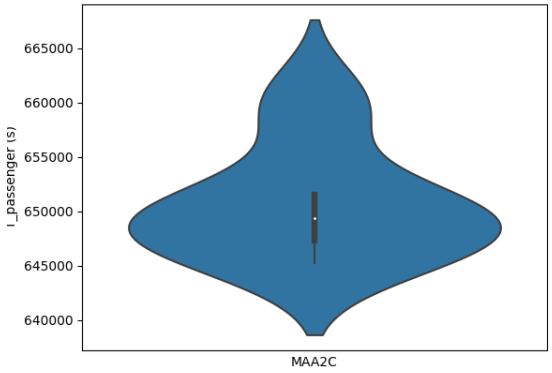
(a) Violin plot of $T_{passenger}$ for PPO



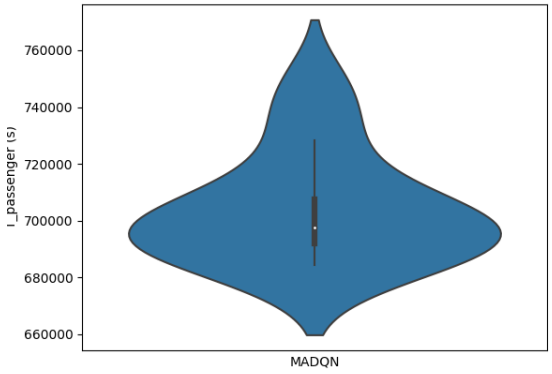
(b) Violin plot of $T_{passenger}$ for A2C



(c) Violin plot of $T_{passenger}$ for MAPPO

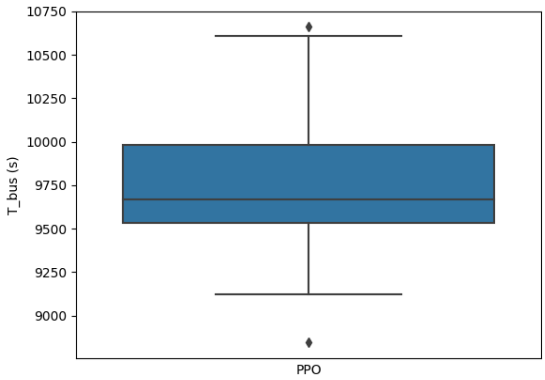


(d) Violin plot of $T_{passenger}$ for MAA2C

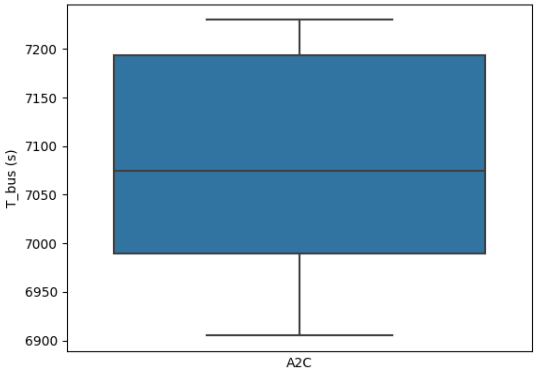


(e) Violin plot of $T_{passenger}$ for MADQN

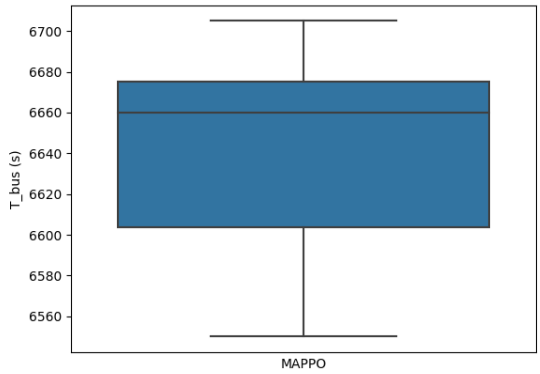
Figure B.22: Violin plots of $T_{passenger}$ with 10% more demand of Case 2



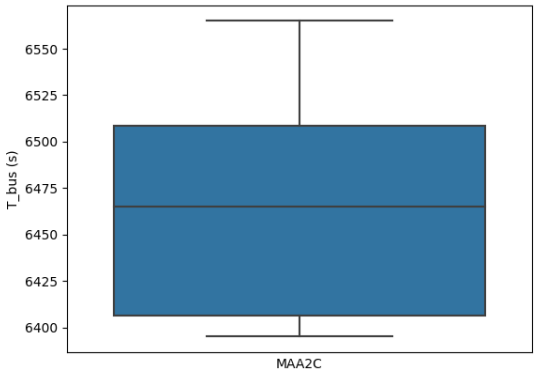
(a) Box plot of T_{bus} for PPO



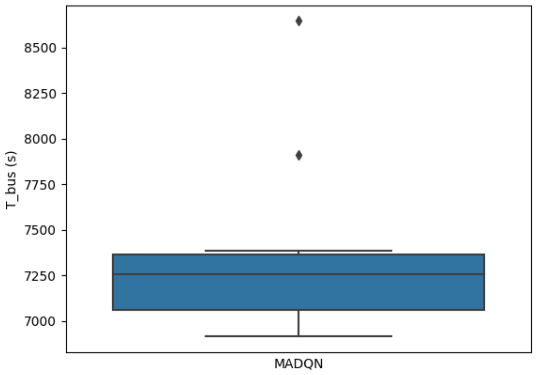
(b) Box plot of T_{bus} for A2C



(c) Box plot of T_{bus} for MAPPO

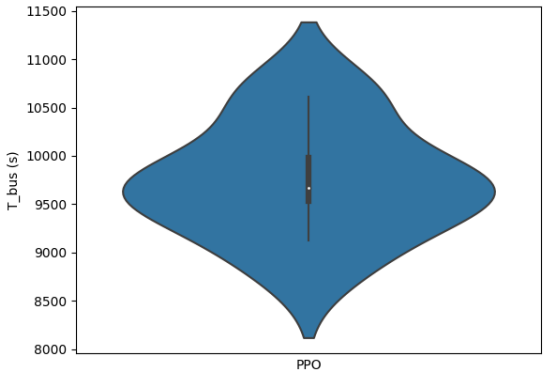


(d) Box plot of T_{bus} for MAA2C

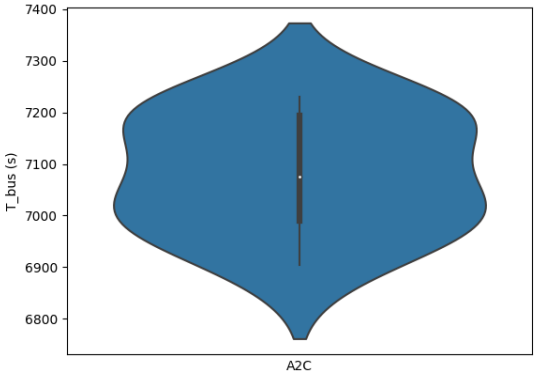


(e) Box plot of T_{bus} for MADQN

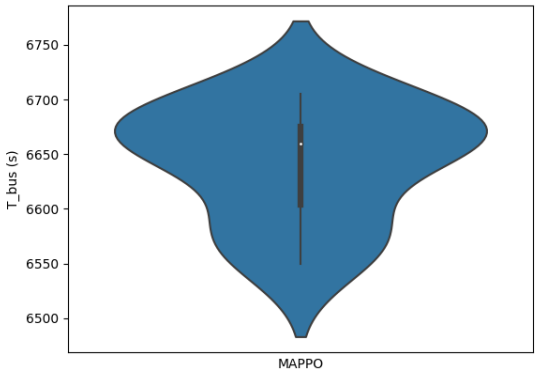
Figure B.23: Box plots of T_{bus} with 10% more demand of Case 2



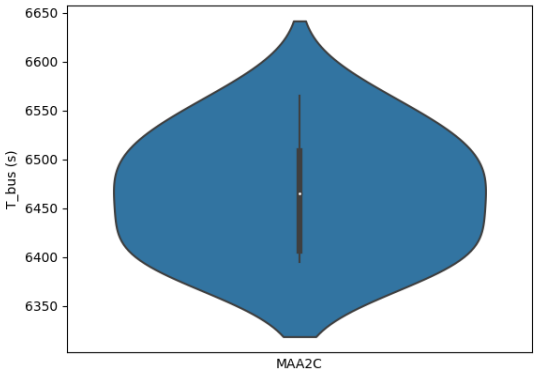
(a) Violin plot of T_{bus} for PPO



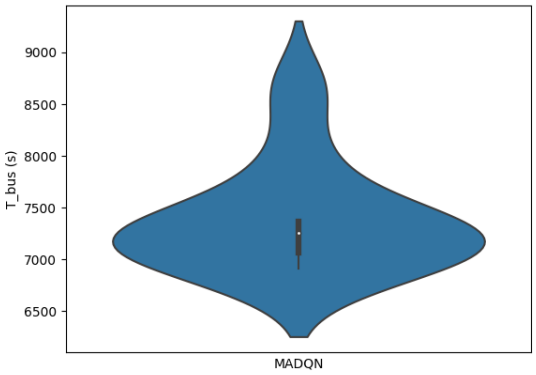
(b) Violin plot of T_{bus} for A2C



(c) Violin plot of T_{bus} for MAPPO

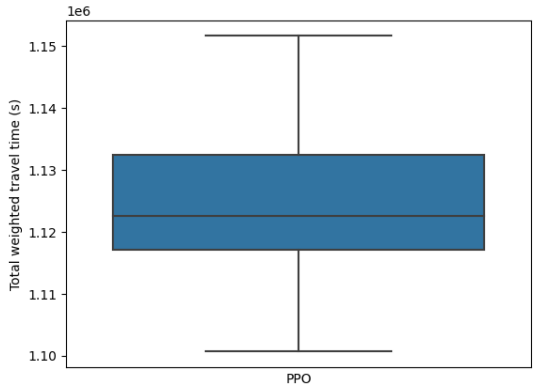


(d) Violin plot of T_{bus} for MAA2C

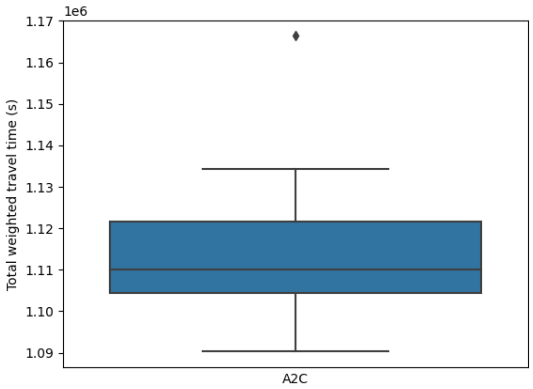


(e) Violin plot of T_{bus} for MADQN

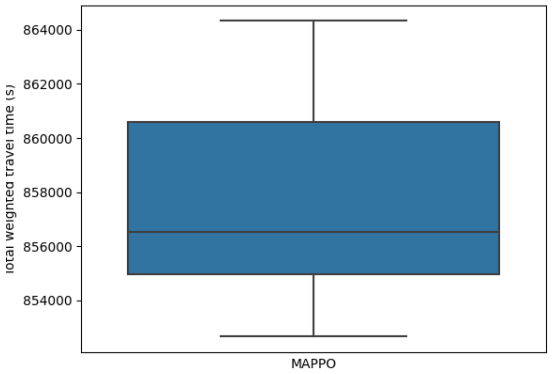
Figure B.24: Violin plots of T_{bus} with 10% more demand of Case 2



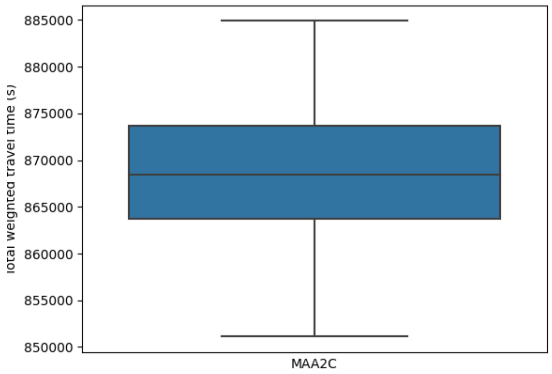
(a) Box plot of total weighted travel time for PPO



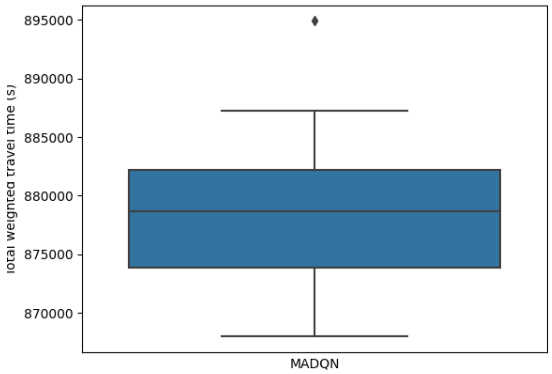
(b) Box plot of total weighted travel time for A2C



(c) Box plot of total weighted travel time for MAPPO

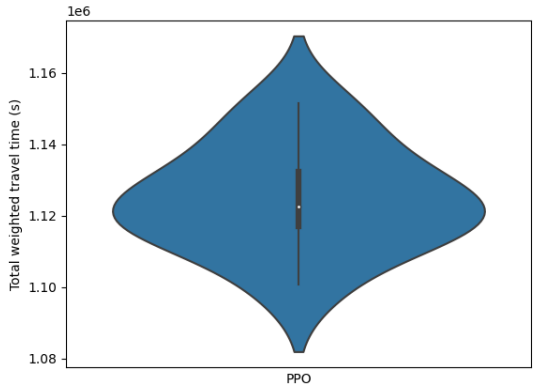


(d) Box plot of total weighted travel time for MAA2C

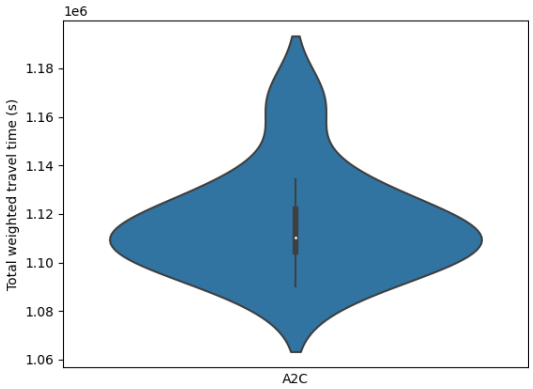


(e) Box plot of total weighted travel time for MADQN

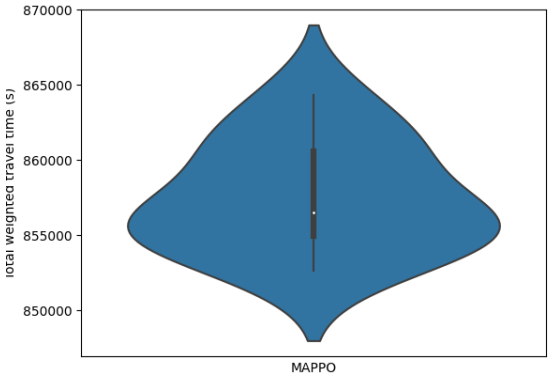
Figure B.25: Box plots of total weighted travel time with 10% less demand of Case 3



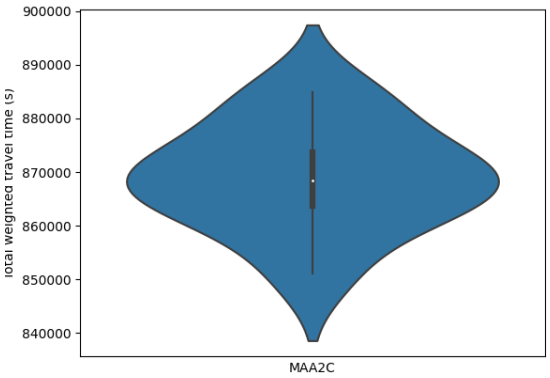
(a) Violin plot of total weighted travel time for PPO



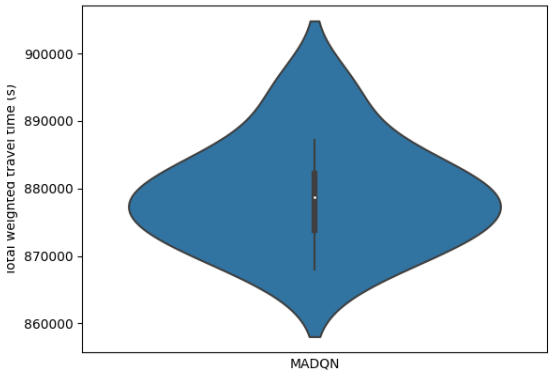
(b) Violin plot of total weighted travel time for A2C



(c) Violin plot of total weighted travel time for MAPPO

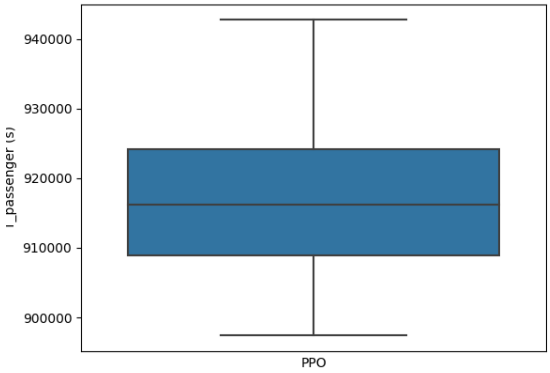


(d) Violin plot of total weighted travel time for MAA2C

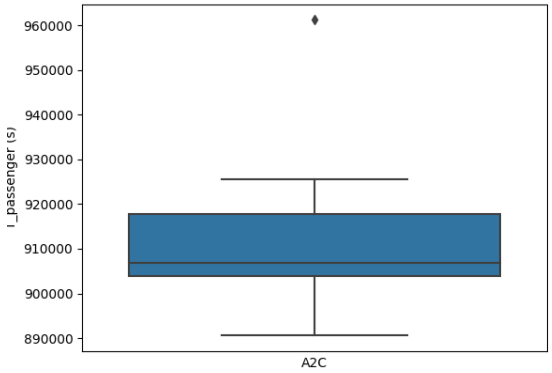


(e) Violin plot of total weighted travel time for MADQN

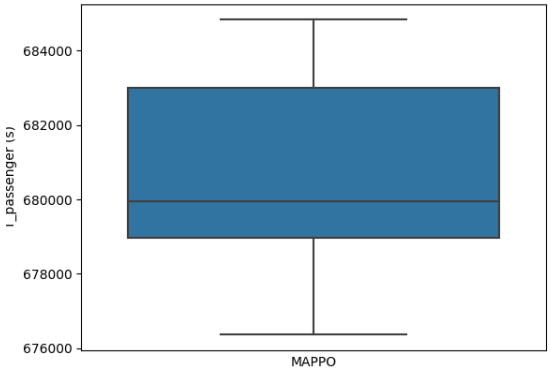
Figure B.26: Violin plots of total weighted travel time with 10% less demand of Case 3



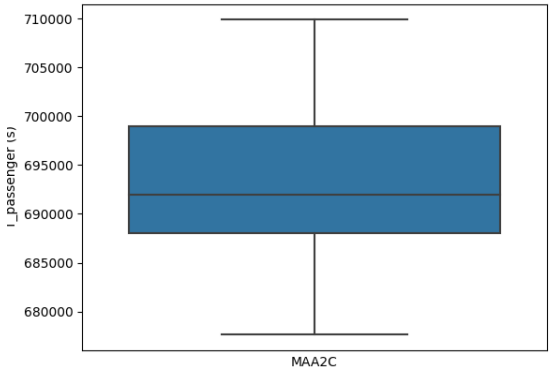
(a) Box plot of $T_{passenger}$ for PPO



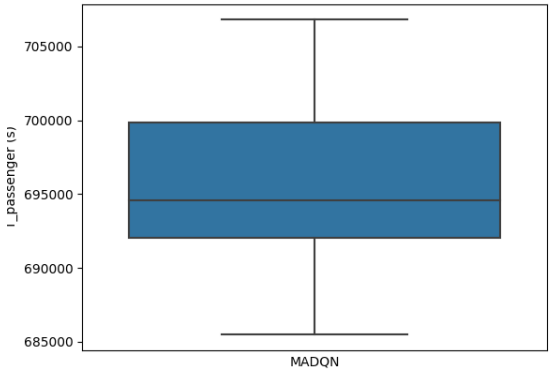
(b) Box plot of $T_{passenger}$ for A2C



(c) Box plot of $T_{passenger}$ for MAPPO

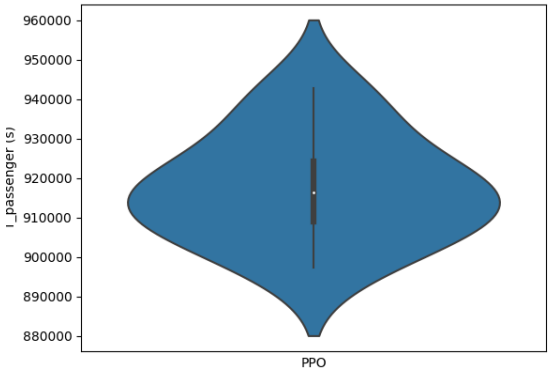


(d) Box plot of $T_{passenger}$ for MAA2C

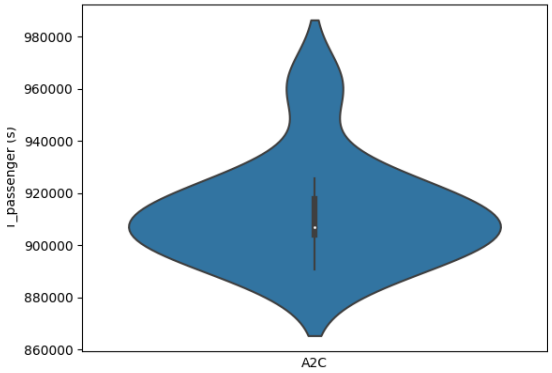


(e) Box plot of $T_{passenger}$ for MADQN

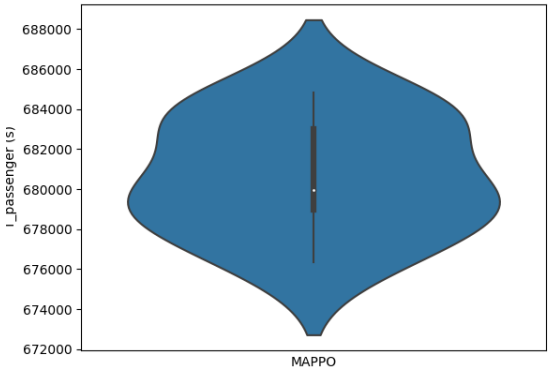
Figure B.27: Box plots of $T_{passenger}$ with 10% less demand of Case 3



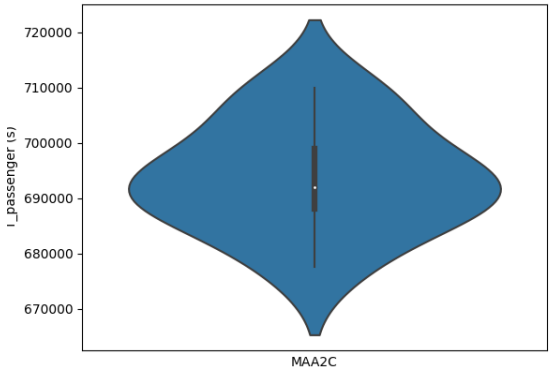
(a) Violin plot of $T_{passenger}$ for PPO



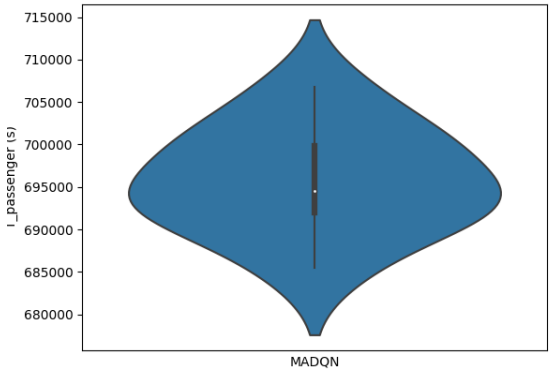
(b) Violin plot of $T_{passenger}$ for A2C



(c) Violin plot of $T_{passenger}$ for MAPPO

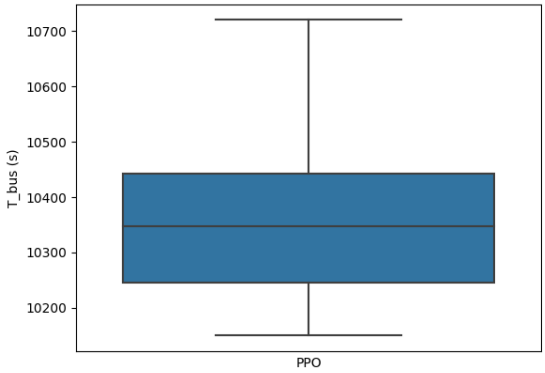


(d) Violin plot of $T_{passenger}$ for MAA2C

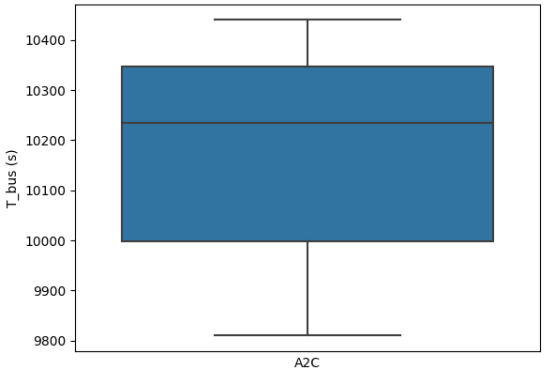


(e) Violin plot of $T_{passenger}$ for MADQN

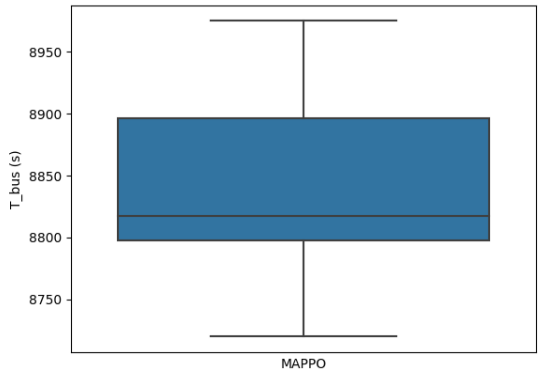
Figure B.28: Violin plots of $T_{passenger}$ with 10% less demand of Case 3



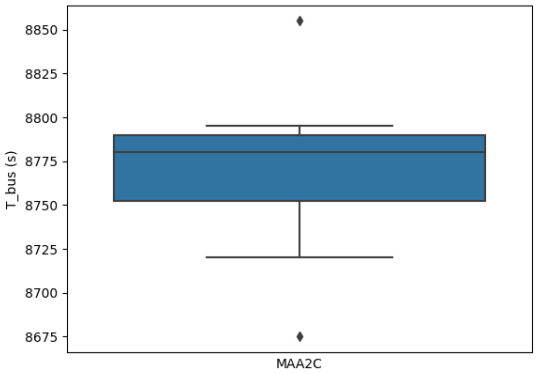
(a) Box plot of T_{bus} for PPO



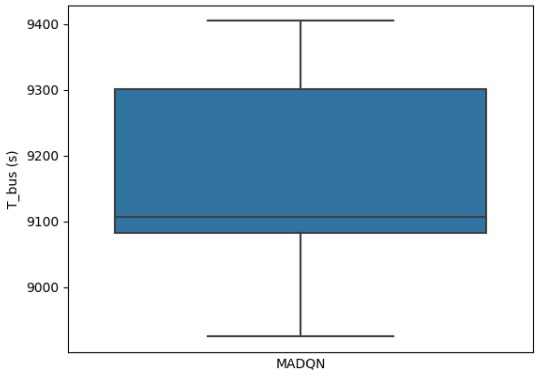
(b) Box plot of T_{bus} for A2C



(c) Box plot of T_{bus} for MAPPO

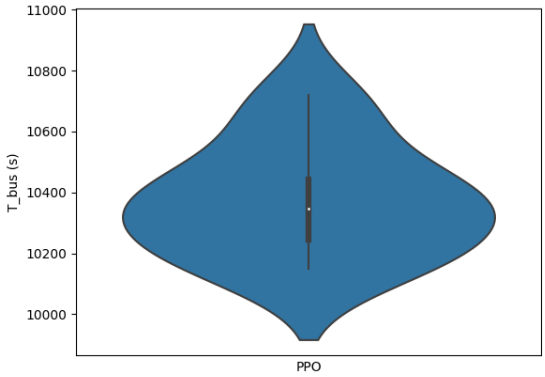


(d) Box plot of T_{bus} for MAA2C

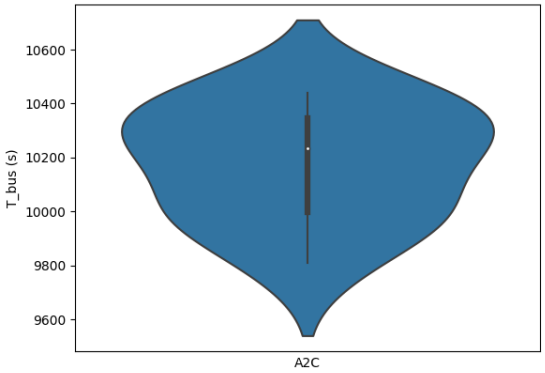


(e) Box plot of T_{bus} for MADQN

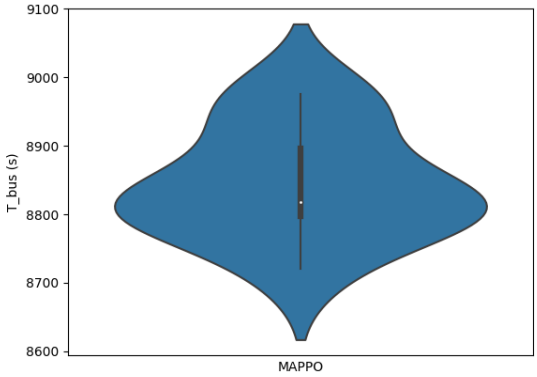
Figure B.29: Box plots of T_{bus} with 10% less demand of Case 3



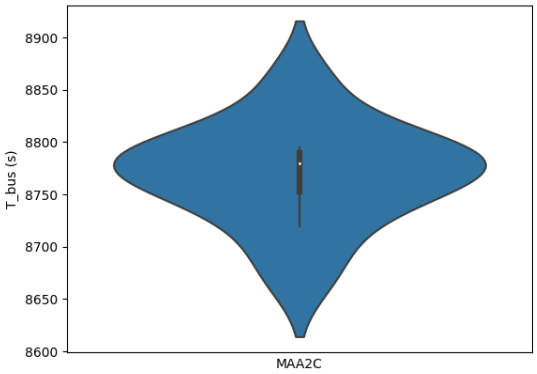
(a) Violin plot of T_{bus} for PPO



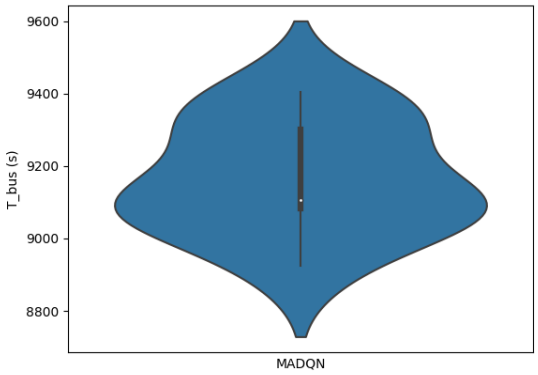
(b) Violin plot of T_{bus} for A2C



(c) Violin plot of T_{bus} for MAPPO

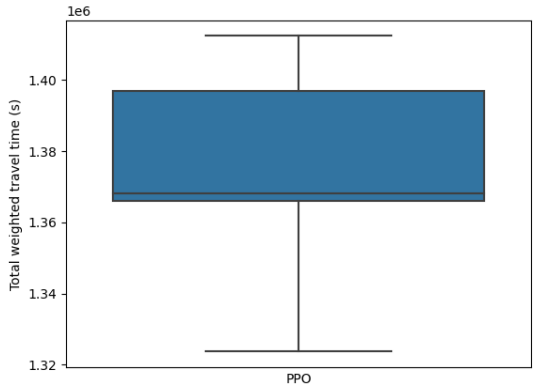


(d) Violin plot of T_{bus} for MAA2C

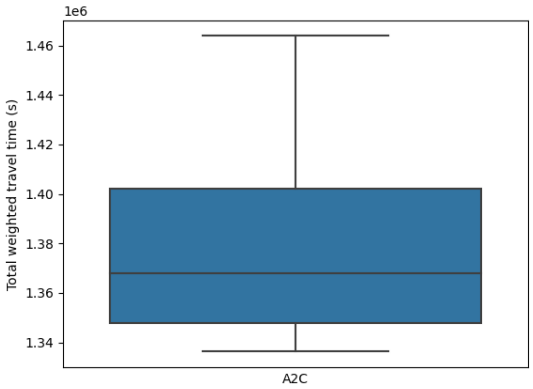


(e) Violin plot of T_{bus} for MADQN

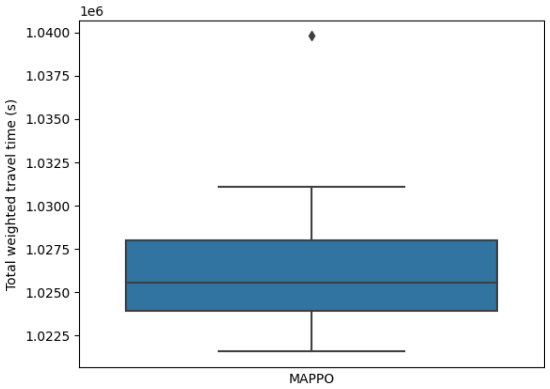
Figure B.30: Violin plots of T_{bus} with 10% less demand of Case 3



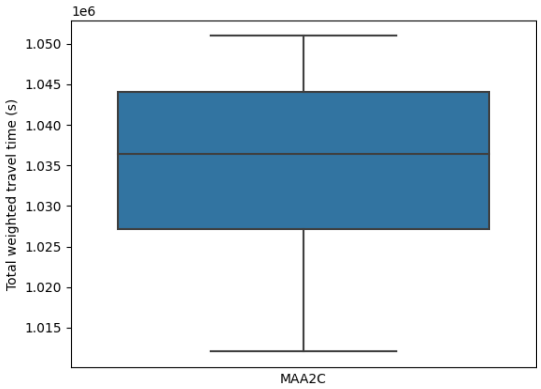
(a) Box plot of total weighted travel time for PPO



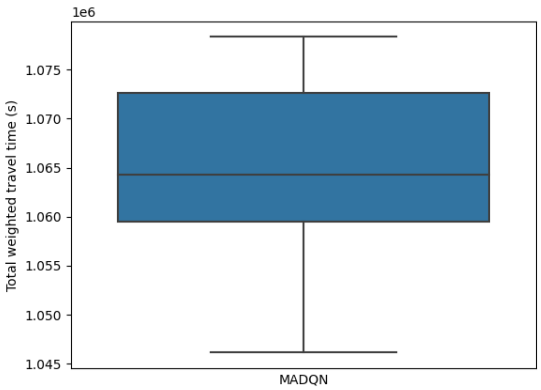
(b) Box plot of total weighted travel time for A2C



(c) Box plot of total weighted travel time for MAPPO

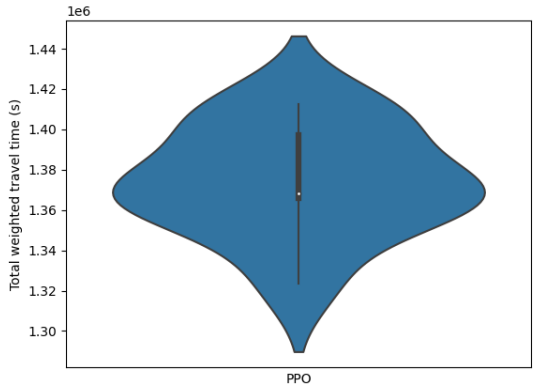


(d) Box plot of total weighted travel time for MAA2C

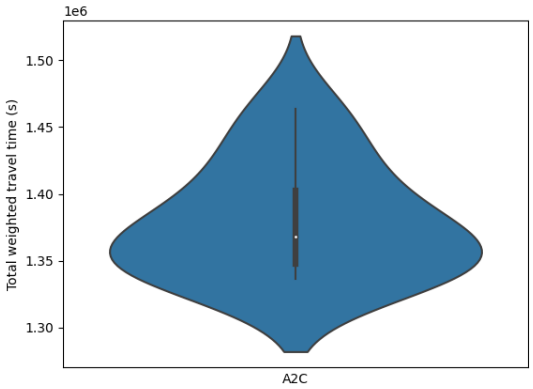


(e) Box plot of total weighted travel time for MADQN

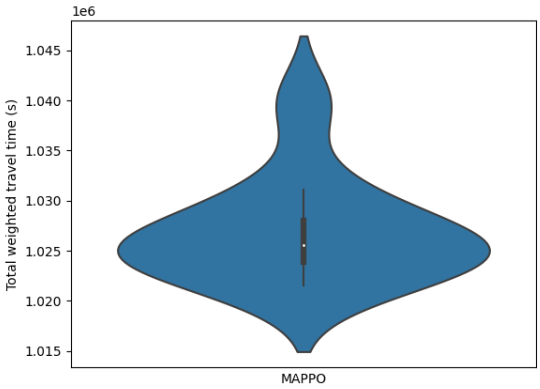
Figure B.31: Box plots of total weighted travel time with 10% more demand of Case 3



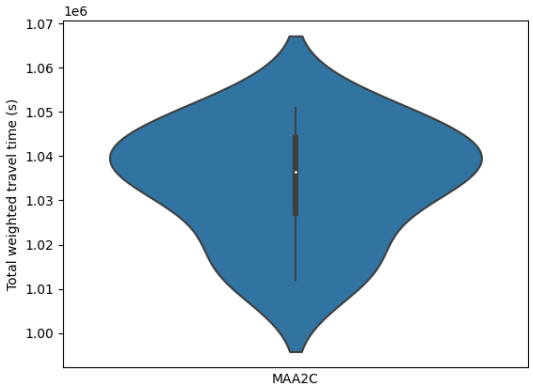
(a) Violin plot of total weighted travel time for PPO



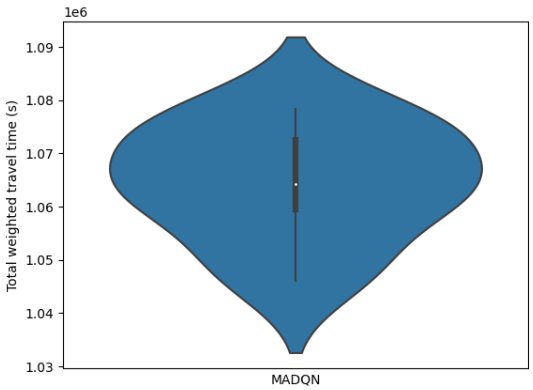
(b) Violin plot of total weighted travel time for A2C



(c) Violin plot of total weighted travel time for MAPPO

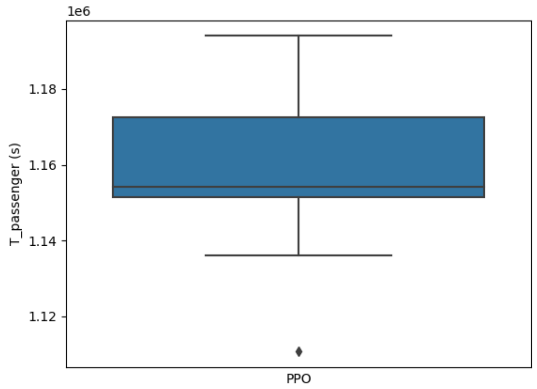


(d) Violin plot of total weighted travel time for MAA2C

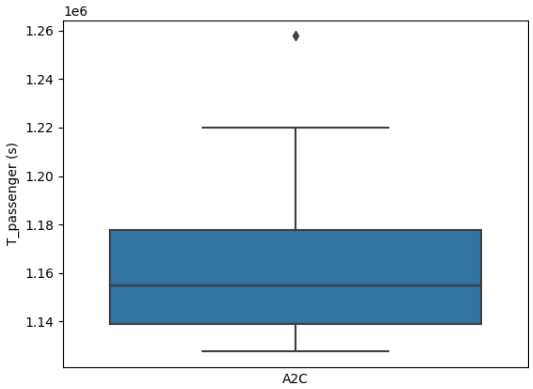


(e) Violin plot of total weighted travel time for MADQN

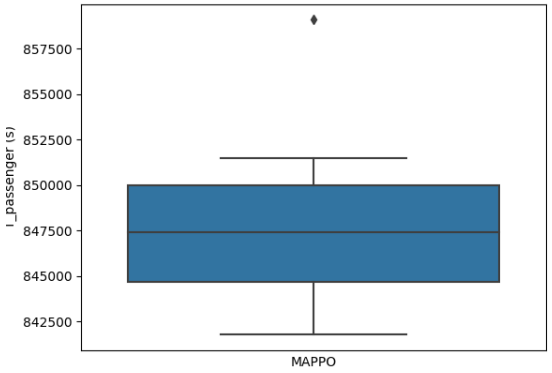
Figure B.32: Violin plots of total weighted travel time with 10% more demand of Case 3



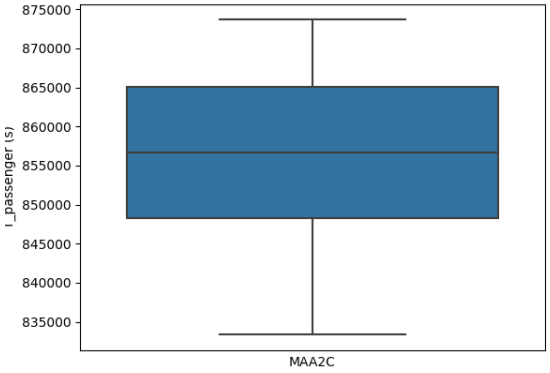
(a) Box plot of $T_{passenger}$ for PPO



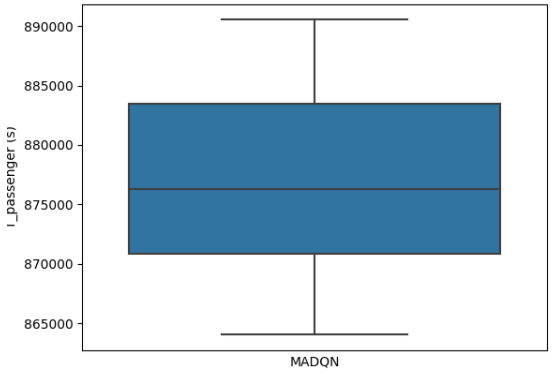
(b) Box plot of $T_{passenger}$ for A2C



(c) Box plot of $T_{passenger}$ for MAPPO

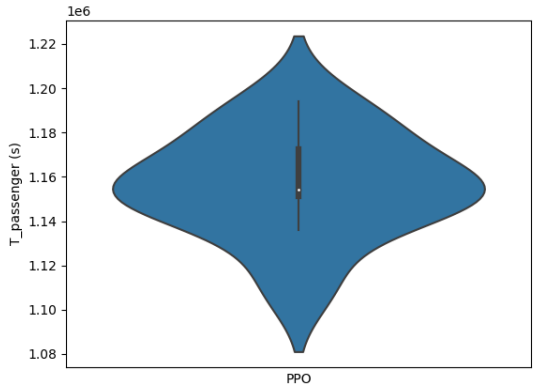


(d) Box plot of $T_{passenger}$ for MAA2C

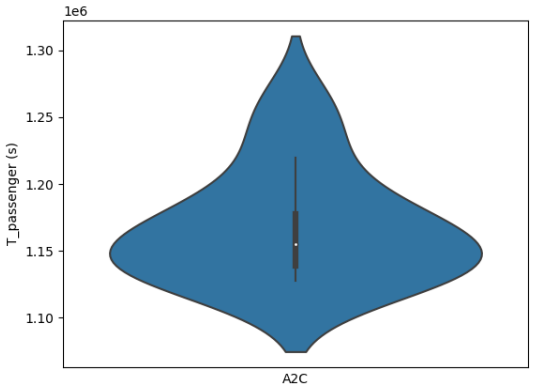


(e) Box plot of $T_{passenger}$ for MADQN

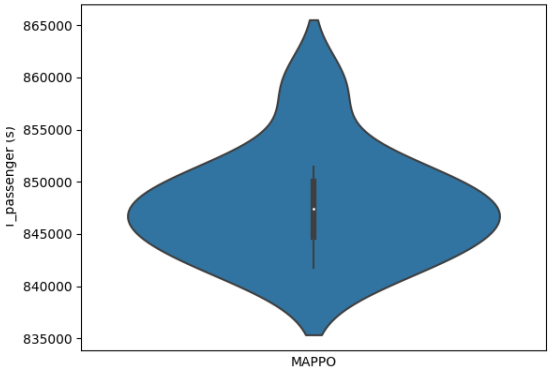
Figure B.33: Box plots of $T_{passenger}$ with 10% more demand of Case 3



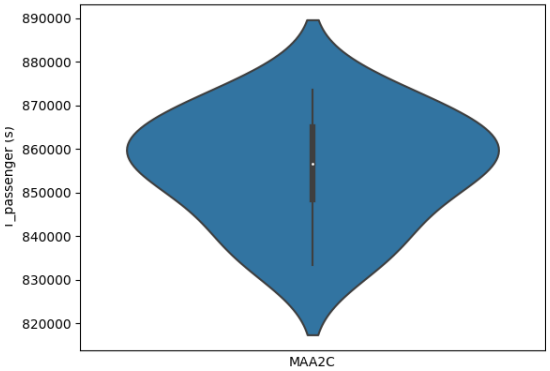
(a) Violin plot of $T_{passenger}$ for PPO



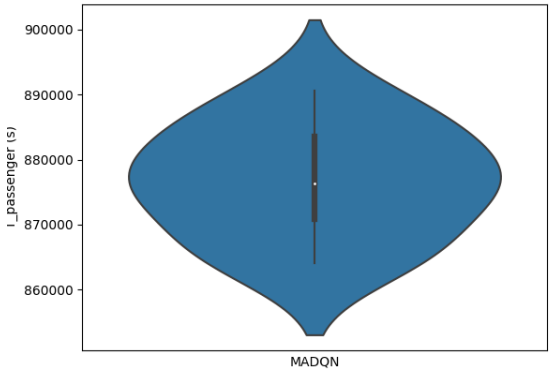
(b) Violin plot of $T_{passenger}$ for A2C



(c) Violin plot of $T_{passenger}$ for MAPPO

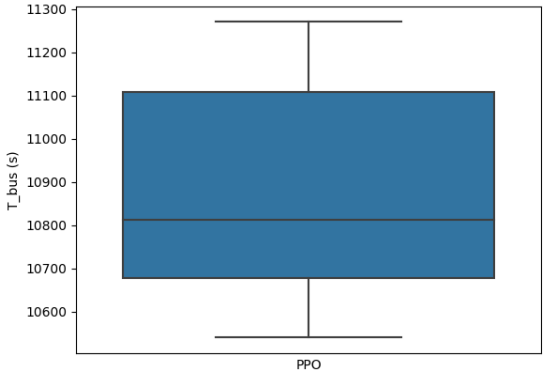


(d) Violin plot of $T_{passenger}$ for MAA2C

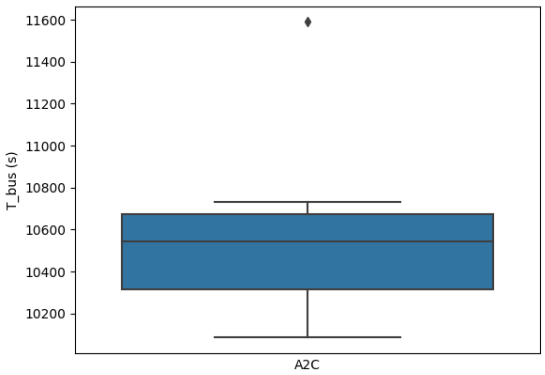


(e) Violin plot of $T_{passenger}$ for MADQN

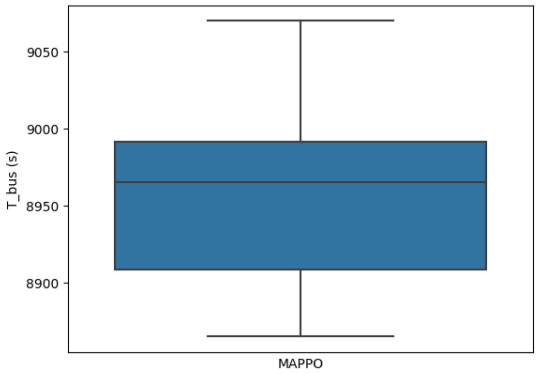
Figure B.34: Violin plots of $T_{passenger}$ with 10% more demand of Case 3



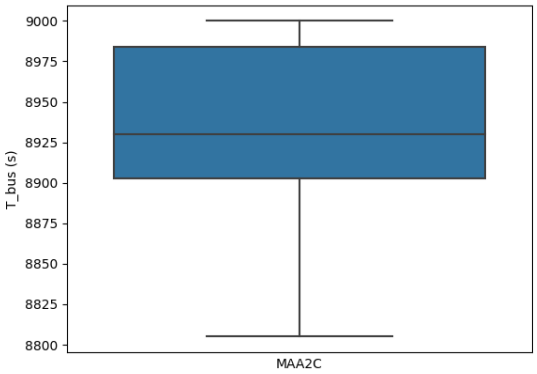
(a) Box plot of T_{bus} for PPO



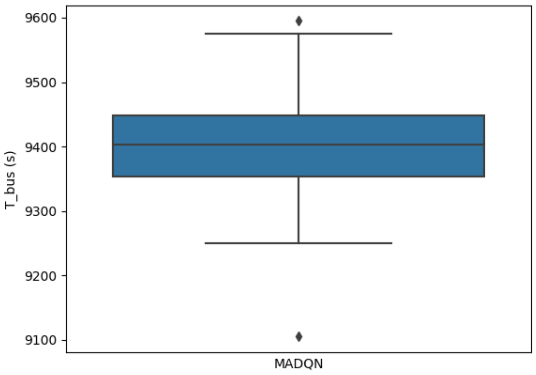
(b) Box plot of T_{bus} for A2C



(c) Box plot of T_{bus} for MAPPO

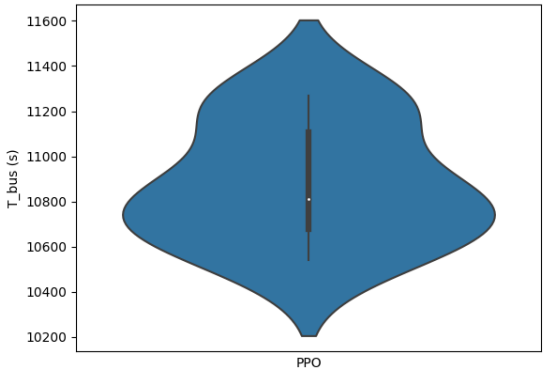


(d) Box plot of T_{bus} for MAA2C

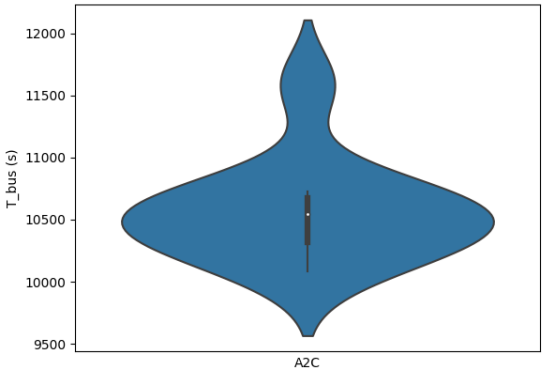


(e) Box plot of T_{bus} for MADQN

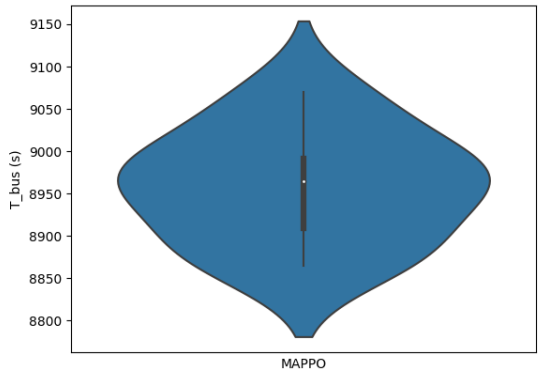
Figure B.35: Box plots of T_{bus} with 10% more demand of Case 3



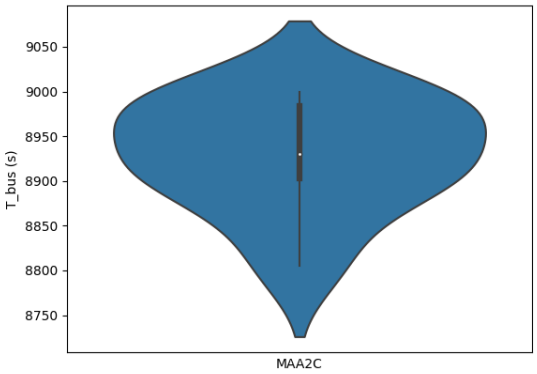
(a) Violin plot of T_{bus} for PPO



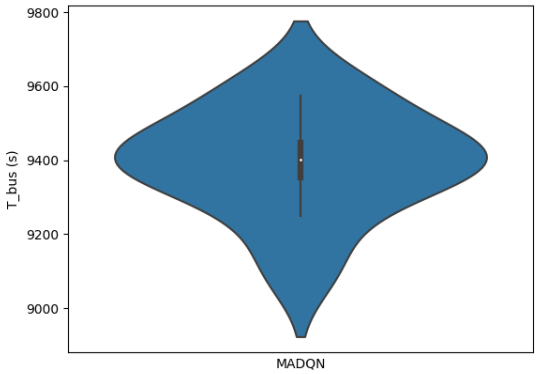
(b) Violin plot of T_{bus} for A2C



(c) Violin plot of T_{bus} for MAPPO



(d) Violin plot of T_{bus} for MAA2C



(e) Violin plot of T_{bus} for MADQN

Figure B.36: Violin plots of T_{bus} with 10% more demand of Case 3