UNIVERSITY MOHAMED BOUDHIAF- M'SILA
FACULTY OF MATHEMATICS AND INFORMATICS
Department of Computer Science

Thesis
Submitted in partial fulfilment of the requirements for the degree of *DOCTORATE 3rd Cycle* in Computer Science

By: Chergui Oussama

SUBJECT

Management and Optimization of road traffic in a smart city

Presented publicly: 05/10/2024, to the jury:

Mr. Akrouf Samir	Professor	University of M'sila	President
Mr. Lamri Sayad	Professor	University of M'sila	Supervisor
Mr. Fouzi Semcheddine	Professor	University of Setif	Examiner
Mr. Debbi Hicham	Professor	University of M'sila	Examiner
Mr. Barkat Abdelbassat.	MCA	University of M'sila	Examiner

Academic year 2024 /2025

Acknowledgements

I would like to sincerely thank my supervisor, Lamri Sayad, for his continuous support, wisdom, and mentorship throughout this journey. The completion of this thesis would not have been possible without his guidance and constructive feedback.

A heartfelt thank you to my parents for patience and belief in me. Your constant support has given me the strength to overcome every challenge.

Lastly, I am forever grateful to my aunt, Fadhila, whose kindness and encouragement have been a source of strength and inspiration. Your presence in my life has made this journey all the more meaningful.

Table of content

List of figures

List of tables

Chapter 1 General Introduction

The term "city" refers to a densely populated area with a significant human settlement and is usually managed by a governing body responsible for overseeing transportation systems and providing services and utilities to its residents. Over recent decades, there has been a remarkable surge in the population of cities worldwide. As of 2008, around half of the world's population, approximately 3.3 billion people, lived in urban areas. This figure is projected to rise to 70% by 2050 (Albino et al., 2015). Algeria follows this global trend, seen in Figure 1.1, which shows a sharp rise in urban population. It's anticipated that 85% of Algeria's population will live in urban areas by 2050.

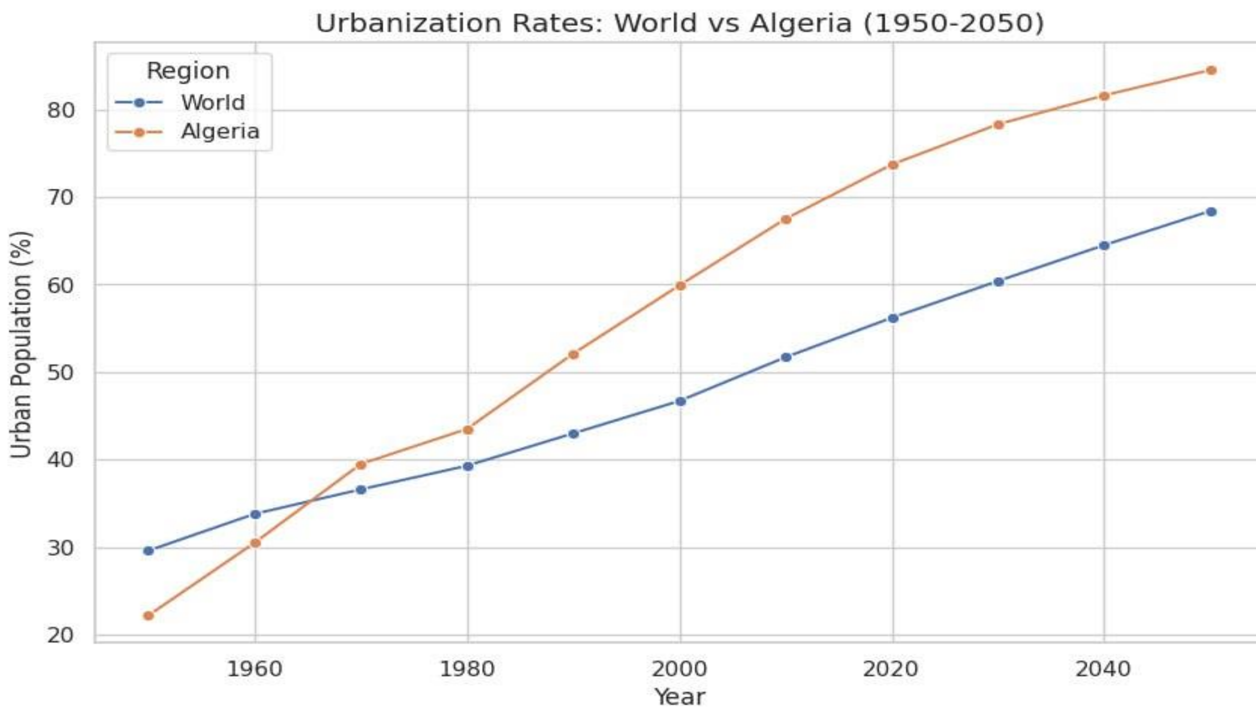

Figure 1.1 Algeria and the world urbanization estimations according to U.N World Urbanization Prospects The 2018 Revision

Despite covering less than 2% of the Earth's surface, cities consume a disproportionate amount, about three-quarters, of natural resources worldwide (Nam & Pardo, 2011b). This, coupled with ongoing population growth, presents new economic and environmental challenges that negatively impact citizens' quality of life. These challenges encompass air and water pollution, rising unemployment and poverty, and complex transportation systems, issues traditional cities have struggled to address.

The rapid urbanization aggravates problems like increased traffic congestion. This growing urban population demands new approaches to tackle transportation needs sustainably. Effective management of transportation networks, integrating smart technologies, and thoughtful urban planning become crucial in addressing congestion issues. Examining how technology, city infrastructure, and policies interact is essential to develop efficient and sustainable mobility solutions amidst this urban growth.

Addressing the congestion challenge demands innovative and 'smarter' approaches. Integration of Information and Communication Technologies (ICT) and Vehicular Ad Hoc Networks (VANETs) into existing infrastructures offers deployable solutions that significantly enhance the quality of life. Examples of such solutions encompass smarter traffic signal management and adaptable public transportation systems, specifically designed to meet societal and economic needs (Albino et al., 2015).

The concept of smart cities, gaining momentum over the past two decades, leverages ICT to optimize city operations by interconnecting various technologies through networks (Batty et al., 2012) These advancements in smart city initiatives, exemplified by enhanced traffic management and tailored transportation systems, mark significant strides in addressing contemporary urban challenges

In our thesis, our primary focus revolves around alleviating traffic congestion in smart cities. To address this challenge, we introduce the Self-Attention Multi-Agent Proximal Policy Optimization (SA-MAPPO) algorithm. Which take advantage of the wireless communication capabilities offered by VANET to control the traffic lights scheduling in order to reduce congestion. Initially, we implement this algorithm in a network comprising six intersections with varying traffic volumes, including both normal and high flows.

Upon successfully applying the algorithm to the smaller network, we expand our approach to a twelve-intersection network. Notably, this expansion requires minimal modifications to the algorithm. Moreover, this adaptation significantly reduces training costs and time compared to retraining the smaller network using the previous pre-trained algorithm.
Key contributions of our work include:

- Integration of attention mechanism alongside Proximal Policy Optimization (PPO) to extract crucial features from raw data. This facilitates a comparative analysis with the baseline fixed algorithm and PPO devoid of the attention mechanism.

- Demonstrating the scalability of the algorithm, evident in its effective performance in managing higher traffic flows within larger networks. Noteworthy is the achieved scalability while incurring minimal training costs, courtesy of utilizing a pre-trained PPO model paired with a streamlined encoder network.

Our thesis is structured as follows:

Chapter 2 We delve into the context and background of our thesis, starting with a comprehensive definition of a smart city. We explore the challenges it presents, providing examples, and then introduce Intillegent Transportation Systems (ITS) and VANET, defining their components and explaining their role in implementing our algorithm. Finally, we discuss Intelligent Traffic Management, explaining related terms and components, along with the challenges involved

Chapter 3: In this chapter, we will discuss adaptive traffic light control. We begin by providing a technical background on the approaches used in this field. Then, we will examine in detail the works that have addressed this problem, explaining the implementation details of each approach as well as its limitations and shortcomings.

Chapter 4: Here, we present our algorithm, SA-MAPPO, providing detailed insights into its architecture and hyperparameters. Additionally, we discuss the obtained results**.**

Chapter 5: In this concluding chapter, we summarize our thesis, outlining the limitations of our algorithm, and suggesting potential avenues for future research

1.1 Our published contributions:

- Oussama Chergui, Lamri Sayad**, "Mitigating congestion in multi-agent traffic signal control: an efficient self-attention proximal policy optimization approach**." *International Journal of Information Technology*, https://doi.org/10.1007/s41870-023-01545-8, 2023

- Oussama Chergui, Lamri Sayad**, "Adaptive traffic lights control using proximal policy optimization**." *National Conference on Applied Computing and Smart Technologies*, ACST'21 July 10, 2021, Ecole Supérieure en Informatique, Sidi Bel Abbès,

Chapter 2 Literature Review and Theoretical background

2.1 Introduction

This chapter provides an overview of the key concepts used in our thesis. we delve into the intersection of Smart Cities, Intelligent Transportation Systems (I.T.S), Vehicular Ad-Hoc Networks (VANET) and smart Traffic Management. For each component, we define the challenges, applications, and provide detailed explanations. Our focus is on understanding how these elements come together, particularly in the context of reducing congestion. We aim to explore how technology and networked intelligence can improve the efficiency and safety of urban mobility.

2.2 Smart City

2.2.1 Definitions

The term "smart city" lacks a universally agreed-upon definition and is open to various interpretations. This section aims to distill a fitting definition within the context of this thesis by scrutinizing multiple perspectives.

(Capdevila & Zarlenga, 2015) emphasize the importance of technological infrastructures and their accessibility, defining smart cities as involving the development of technological infrastructures that allow for the emergence of new businesses. They underscore the role of local governments as intermediaries, facilitating solutions between firms and citizens.

(Nam & Pardo, 2011) considered the combination, connection, and integration of systems and infrastructures as fundamental to a city being smart. They defined smart city innovation in terms of changing and upgrading technological tools to improve services, creating managerial and organizational capabilities for effective use of technological tools, and addressing institutional and non-technical urban problems.

(Albino et al., 2015) attributed the confusion surrounding the smart city concept to the fact that the term can be applied to two domains: the "hard" domain involving transportation, natural resources, and waste management, where Information and Communication Technology (ICT) plays a vital role, and the "soft" domain that includes non-physical entities such as governments and education, where ICT applications are not necessary.

(Chourabi et al., 2012) compared smart cities to an organism with a nervous system capable of behaving intelligently, where digital telecommunication networks serve as nerves, sensors and tags represent sensory organs, and software is the knowledge and cognitive competence.

(Benevolo et al., 2016) considered a smart city as one with a competitive economy, active citizen participation in governance, high quality of life, reduced pollution, smart mobility, and smart people.

All definitions stress the critical role that technology plays in a smart city, including ICT infrastructures, efficient transportation systems, and networks connecting all independent services within the city.

However, technology alone is not sufficient to make a city smart. Smart governance is crucial, where decision-makers foster a competitive economy, reduce pollution by using renewable energy, impose transparency and accountability policies, provide good healthcare systems functioning during unexpected events such as pandemics, push for innovation, help high-tech businesses and industries to create employment, and encourage citizens' active role in governance

Moreover, smart cities should not only attract but also cultivate smart people—talented and highly skilled individuals—by providing them with opportunities to develop their expertise and actively contribute to urban innovation. These smart people play a pivotal role in shaping the city by driving technological advancements, participating in smart governance, and fostering collaborative solutions. In summary, as illustrated Figure 2.1 a smart city combines technology, smart governance, and smart people to improve the quality of life and create a sustainable future.

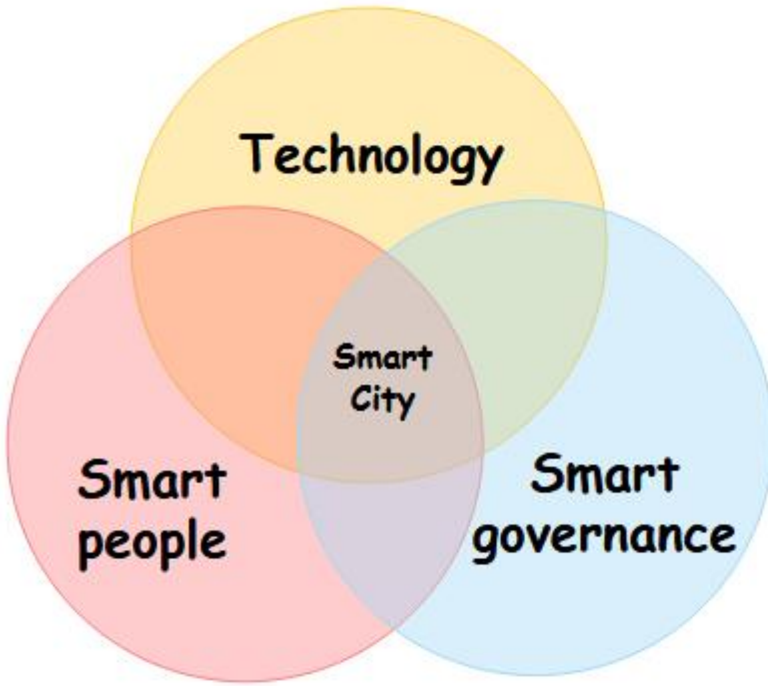

Figure *2*.1 the components of smart city

2.2.2 case studies

In recent decades, many cities around the world have been pursuing the goal of becoming a smart city, due to the potential benefits it promises. As of 2013, there were already 143 smart city projects underway globally, with 35 in America, 47 in Europe, 50 in Asia, 10 in Africa and the Middle East, and 10 in South America (Albino et al., 2015; Lee et al., 2014)

Singapore, ranked as the smartest city in the inaugural IMD Smart City Index, launched its Smart Nation program in 2014, which involved installing sensors throughout the city to facilitate data and information sharing. Singapore boasts the world's most open and competitive economy and has numerous projects that make it a smart city, including the first driverless taxi system and the use of drones by the police to fight crimes, with plans to introduce driverless buses by 2020 and integrate robots into the workforce (*6 Technological Advancements That Make Singapore a Smart City*, n.d.)

Barcelona, which was chosen as the European Capital of Innovation ("iCapital") in 2014, serves as an example of a city that has been successful in pursuing the smart city concept. Despite the global crisis affecting Spain, Barcelona has experienced an improvement in its economy and quality of life. This was achieved by implementing policies that encourage local innovations, data sharing and communication between universities, research centers, public and private partners, and providing services that revolve around information and communication technology (ICT) (Capdevila & Zarlenga, 2015).

Dubai is another city that has made significant strides towards becoming a smart city in recent years. In 2014, Dubai launched the Dubai Smart City project, which aims to transform the government into a completely paperless and digital system by 2021 Figure 2.2, and leverage emerging technologies such as blockchain and machine learning to meet the needs of its population. Dubai's success is reflected in its growing tourism industry, despite its location and history, with the city ranking as the seventh most popular destination for tourists worldwide in 2019 (Aisha Bint Butti et al, 2021; Khan et al., 2017)

Figure 2.2 Dubai smart city components source: (Smart Dubai 2021 Strategy **|** The Official Portal of the UAE Government, n.d.)

Amsterdam embraced the concept of a smart city long before its widespread adoption. The city demonstrates a strong commitment to leveraging technology and innovation through an open governance model to tackle urban challenges and enhance the residents' quality of life. Approximately 68% of citizens actively contribute to the city's economy. A notable manifestation of Amsterdam's dedication to smart city initiatives is the Energy Atlas project. Launched in 2012 by the Amsterdam Smart City (ASC) organization, a public-private partnership focused on developing and implementing smart city solutions, the Energy Atlas project aligns with the city's ambitious goal of reducing CO_2 emissions by 40% by 2025. This project offers valuable insights into factors such as solar and wind energy potential, along with the locations of existing renewable energy installations. The data provided by the Energy Atlas empowers businesses, residents, and policymakers to make well-informed decisions regarding investments in renewable energy (Putra & Van Der Knaap, 2019)

2.2.3 Challenges

Despite the optimism met with the idea of smart city. However, it has also been subjected to criticisms and questions. The confusion surrounding the definition of smart city has led many cities to label themselves as smart, which is often self-congratulatory, as there is no city that calls itself stupid (Hollands, 2008). Moreover, there is underestimation of the risks associated with adopting new technologies. Start-ups have a high failure rate of 90% (Patel, n.d.) and 85% of IT-related projects fail because of non-technical reasons, which also extends to the public sector that has less favorable conditions for innovation and risk-taking (Nam & Pardo, 2011b)

Smart city policies that encourage immigration may also cause social polarization and culture clashes. While attracting talented and smart people from around the world is beneficial, the influx of immigrants generates tension within the population, fueled by fake news that is widely spread on social media websites. The sharp increase in right-wing popularity in recent years is a result of uncontrolled immigration.

The heavy reliance on software in smart cities makes them vulnerable to hacking, and the consequences of such hacking can be disastrous. Moreover, there are ethical issues surrounding privacy with the selling and exploitation of personal data, as well as the emergence of controversial technologies such as facial recognition. Smart cities are facing a constant backlash, and they have yet to address these ethical issues. Additionally, the technologies employed in smart cities do not take into account the needs of people with disabilities. New technologies should be accessible to all people, not create new barriers (Woyke, 2019)

2.3 Intelligent Transportation System (ITS)

Intelligent Transportation Systems (ITS) represents an innovative approach to address urban mobility challenges by leveraging advanced technologies. ITS spans diverse technologies, responding to challenges in urbanization and homeland security concerns. ITS can be defined within road transport, encompassing infrastructure, vehicles, users, traffic management, mobility, and interfaces with other transport modes.

The evolution of ITS can be divided into three phases: preparation (1930-1980), feasibility study (1980-1992) and product development (1995-present). In the preparation phase, the concept of ITS was introduced but the technologies were not yet mature enough. The first ITS system was the electric traffic signals implemented in 1928. In the feasibility study phase, there was an explosion of development programs both industry and government subsidized. This phase resulted in the development of several ITS technologies, such as the AHS (Automated Highway System) in the US. In the product development phase, the focus shifted to creating feasible products. Several ITS systems were developed during this phase, such as the Chauffeur project in Europe (Figueiredo et al., 2001)

The developing world faces unique challenges, with some areas experiencing congestion due to motorization, while others are rapidly urbanizing. Intelligent transportation systems (ITS) offer promising solutions to address these challenges by improving traffic flow, safety, and efficiency. To achieve these goals, ITS leverage a range of technologies. These technologies can be broadly categorized into three main areas: wireless communication, computational technologies, and sensor technologies.

Wireless communications are pivotal in ITS, utilizing protocols like IEEE 802.11p for short-range and 5G for longer-range communication (Arena et al., 2020; Guevara & Auat Cheein, 2020). Computational technologies in vehicle electronics advance towards fewer, more capable processors with real-time operating systems, prominently featuring artificial intelligence.

Sensing technologies in ITS include inductive loop, video, Bluetooth, and radar detection, each offering distinct advantages. The fusion of data from multiple sensing modalities, incorporating acoustic, image, and sensor data, emerges as a critical aspect for accurately determining traffic states within ITS.

2.3.1 Applications

- **Traffic management**: ITS can be used to improve traffic flow by providing real-time traffic information to drivers, by controlling traffic signals, and by managing incidents (An et al., 2011).

- **Parking management**: ITS can be used to improve the efficiency of parking management by providing real-time information about parking availability, and by guiding drivers to available parking spaces.

- **Freight Management Systems**: leverage intelligent technologies to enhance the efficiency and effectiveness of freight and cargo transportation within the overall transportation network. (Qureshi & Abdullah, 2013)

- **Electronic Toll Collection (ETC)** allows vehicles to pay tolls instantly using a small tag with a radio transponder, interacting with electronic roadside antennas at toll stations (Shaheen & Finson, 2013)

- **Emergency response**: ITS can be used to improve emergency response by providing real-time information about accidents and other incidents, and by guiding emergency responders to the scene.

Accurate data collection and processing, facilitated by sensor technologies, is crucial for intelligent transportation systems. The computational cost of ITS-based software should scale with the growth of the transportation system. Furthermore, seamless integration with existing transportation systems is essential. Vehicular Adhoc NETwork (VANET) is considered a step in the right direction to meet these challenges.

2.4 VANET

Vehicular Adhoc NETwork (VANET) is an innovative technology that has been developed for use in intelligent transportation systems (ITS). The nodes in this network are represented by vehicles that can communicate with one another using wireless communication technology that is mounted onboard. In addition to communicating with one another, these vehicles can also exchange messages and data with other infrastructures in order to address modern transportation problems such as accidents, traffic congestion, and air pollution.

we define the Characteristics of VANET as (Tanuja et al., 2015)

- **active topology**: the speedy movements of the vehicles and the availability of the paths in VANET gives the network an active topology

- **predictable Motion Patterns**: as the vehicles move on roads that are already designed and the paths that a vehicle can take are already known, as opposed to MANET.

- **bigger Battery Power and Storage**: batteries and storage devices are placed inside a vehicle, which gives them a greater capacity.

2.4.1 Components

VANET has three major components:

- **Roadside Unit (RSU)**: The RSU component as seen in Figure 2.3 is the static node of VANET and is equipped with short-range 802.11p wireless communication technology. Its primary purpose is to extend the communication range of the vehicle and to distribute information received from On Board Unit **(**OBUs) to other OBUs and RSUs (Panjrath & Poriye, 2017). RSUs act as gateways for vehicles to connect to the internet and provide safety applications from accidents and other hazards by identifying vehicles using GPS and other authentication processes (Tanuja et al., 2015).

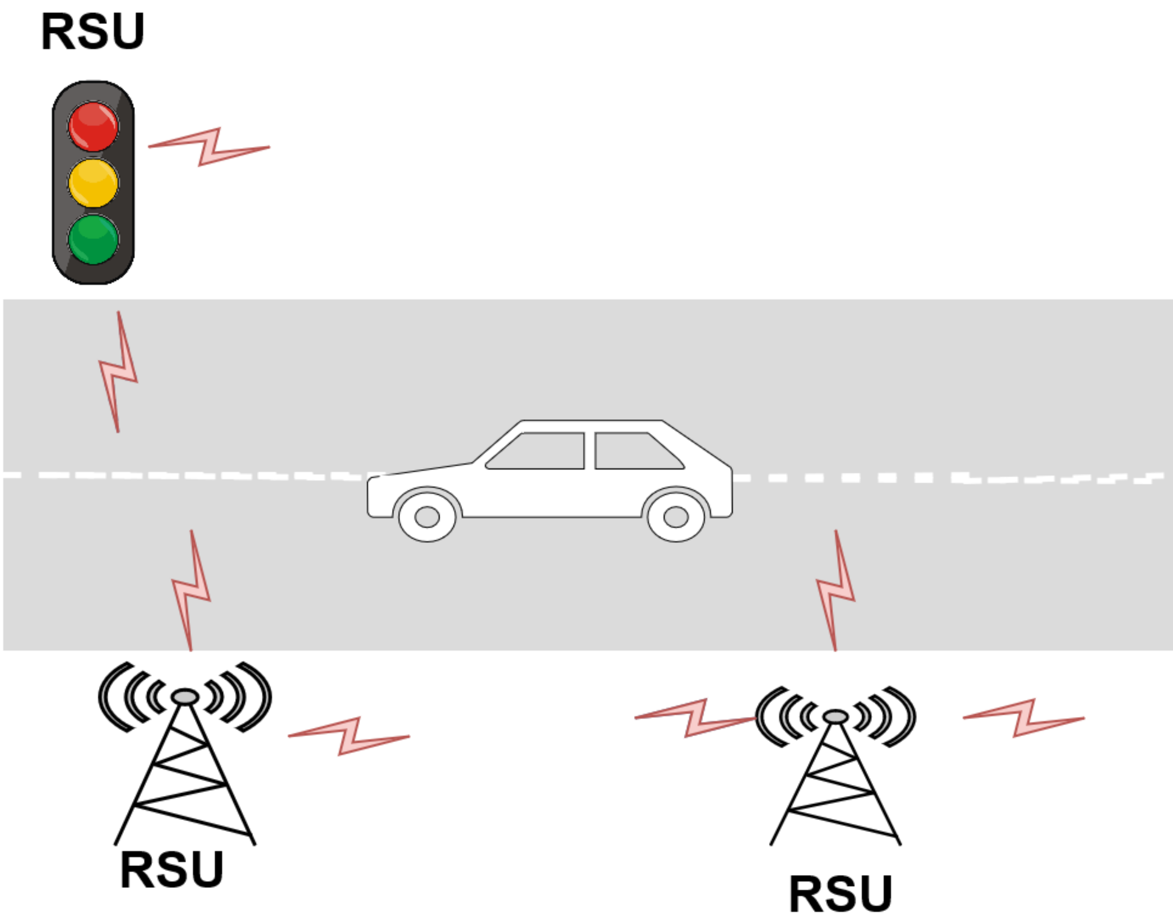

Figure 2.3 roadside units in Vanet network

- **On Board Unit (OBU)**: OBU is the mobile component of VANET, mounted on vehicles and equipped with wireless communication capabilities that communicate and exchange data with RSUs and OBUs of other vehicles (Tanuja et al., 2015). OBUs are connected to other nodes through wireless lines based on the IEEE 802.11p radio frequency channel (Panjrath & Poriye, 2017). OBUs consist mainly of memory for storage, GPS for position sharing, wireless communication devices, and a human interface that works best with a voice system to avoid driver distraction while driving.

- **Application Unit (AU)** AU is a device mounted inside the vehicle that allows it to use the applications provided by VANET technology providers. The AU communicates with the OBU through wireless or wired communication, and it can be used for security applications or as a personal digital assistant to run the internet (Tanuja et al., 2015). Figure 2.4 illustrates both OBUs and AUs.

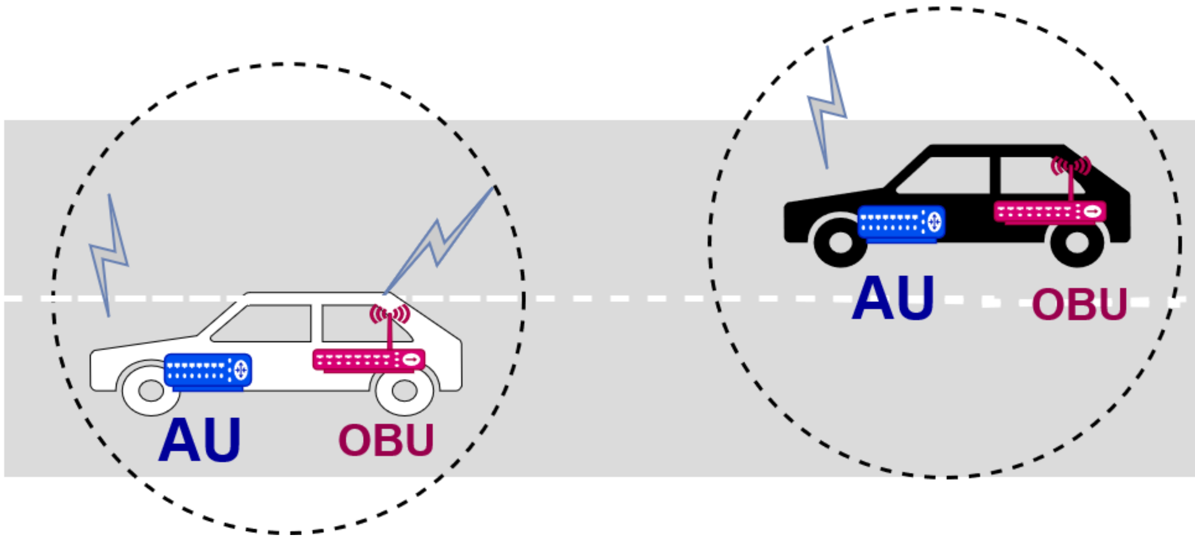

Figure 2.4 On Board Unit and application unit in VANET

2.4.2 Communication types

The communication within VANET framework can be achieved through:

- **Dedicated Short-Range Communications** (DSRC) primarily using IEEE 802.11p standard for vehicular based communications (Fitah et al., 2018).

- **Cellular Networks:** VANETs can also utilize existing cellular networks for communication. Vehicles establish connections with the cellular network, enabling a wider array of services, such as internet access, traffic information, and multimedia content (Y. Li et al., 2012).

As pictured in Figure 2.5, The communication between VANET components can be divided into four parts (Shandil, 2023):

In-vehicle communication: This architecture allows for the detection of the internal condition of the vehicle, as well as the health status of the driver in cases of fatigue. This is crucial for both the safety of the driver and the public.

Vehicle-to-vehicle communication (V2V): This type of communication involves sharing information between vehicles that can be useful to each other.

Vehicle-to-infrastructure communication (V2I): This is the architecture that enables the communication between vehicles and roadside units, which can provide data about the state of the road or traffic updates.

Vehicle-to-broadband communication (V2B): This architecture facilitates wireless communication between vehicles and the internet, providing drivers with more information than what roadside units can provide. V2B communication is particularly critical for driver assistance.

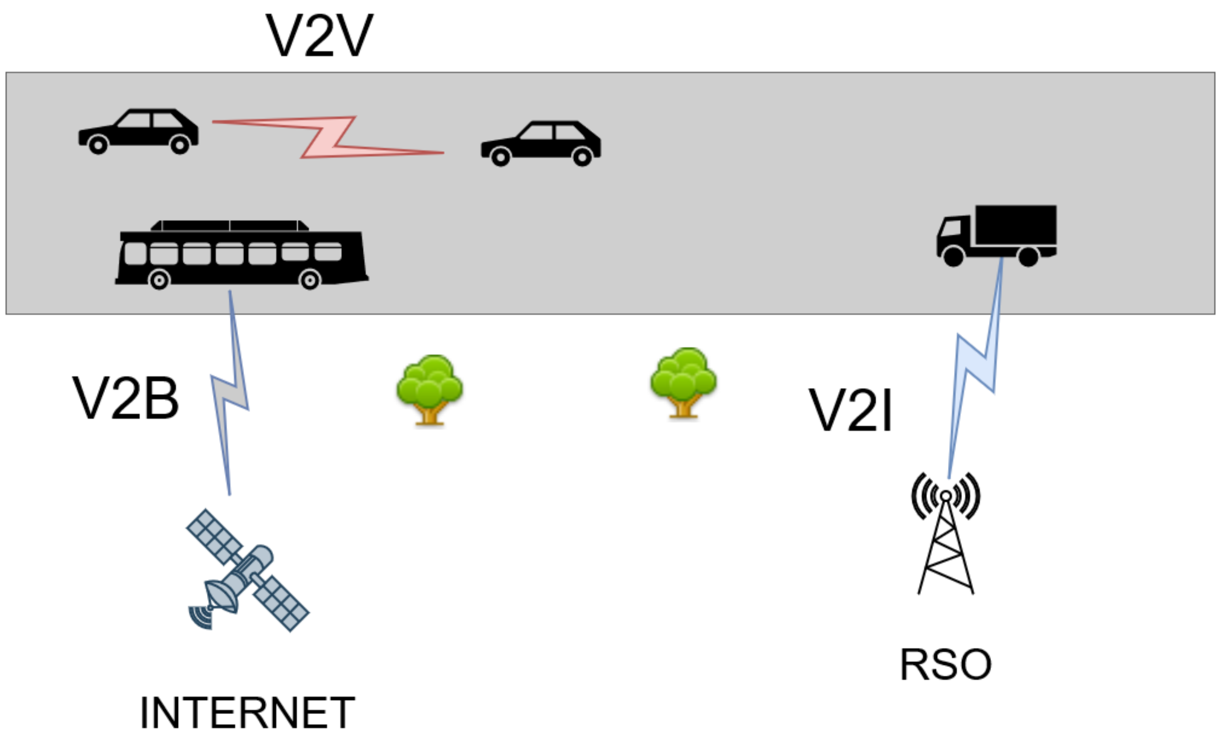

Figure 2.5 communication architectures in VANET

2.4.3 Applications

Traffic management: Traffic congestion is a significant problem that causes both economic and environmental issues. VANET is used to improve traffic management to reduce traffic and make travel smoother for vehicles. A vehicle stuck in congestion can broadcast messages to other vehicles to warn them and suggest alternate routes. The same messages can be sent to vehicles far away through roadside infrastructure for the same purpose. VANET technologies can also be used to optimize traffic signals to display the best light in the moment and escape traffic bottlenecks. Efficient traffic management can be helpful in reducing travel time and fuel consumption, benefiting the environment and the economy. (Vegni et al., 2013)

Safety:

Vehicular accidents are one of the leading causes of death worldwide. VANET has been developed with the primary aim of improving road safety. Real-time information is shared through V2V and V2I communications to minimize the number of accidents and vehicle collisions. Predictive models are used to identify potential accident situations, and warnings are broadcasted to drivers so that they can take necessary precautions to avoid accidents. Another safety measure that is used is vehicular platooning, where vehicles use the same speed without the need to change lanes. In post-crash situations, a vehicle involved in an accident can send a message to nearby vehicles to warn them of potential traffic congestion. (Dixit et al., 2016) (Vegni et al., 2013)

Entertainment and comfort-related applications :

VANET can be used to provide entertainment and comfort to drivers for a better driving experience. Applications that assist drivers in their needs, such as notifications for the nearest repair shop, or restaurants that provide their favorite food, can be easily accessed through VANET. The availability of internet access, provided by V2I communications, allows for many forms of entertainment, such as streaming services, where drivers can watch their favorite shows. Playing games using services like Fleetnet for gaming and peer-to-peer file transfer is another popular entertainment option. Another essential application is providing drivers with information on the availability of parking spaces near them. (Vegni et al., 2013)

2.4.4 Challenges

Although VANET technology is promising and beneficial, there are several challenges that researchers are currently addressing, the challenges are as follows:

Data storage and management: As a VANET network consists of numerous nodes, storing and managing the relevant data can be problematic. With the number of nodes reaching a level where it becomes an issue, concepts such as Big Data and Cloud computing have emerged in recent years as a means of addressing the issue. (Tomar et al., n.d.)

Security: Security is a significant concern in VANET networks due to the wireless communication and information shared between nodes. A fake node can integrate itself into the network and receive messages that can affect the privacy and safety of drivers. Therefore, developing a secure and robust VANET network should always be a priority.

Standardization: VANET nodes, including cars, buses, trucks, poles, and traffic lights, are quite different from each other. The heterogeneous nature of VANET nodes and their differences can be challenging to deal with in dense networks, which require standardization of communication protocols to address this issue. (Tomar et al., n.d.)

Scalability: VANET networks are expected to adapt to the density of nodes in a city and be flexible and functioning regardless of the position and location of the nodes.

Quality of service: The high dynamic topology and mobility of VANET networks make optimizing the available bandwidth and latency of messages difficult. Developing and maintaining good QOS in VANET remains an open challenge. (Dixit et al., 2016; Zeadally et al., 2012)

2.5 Intelligent Traffic Management

Extensive research in recent decades has focused on traffic management and congestion reduction, addressing the economic and environmental challenges arising from the growing number of vehicles in urban cities. Technological advancements have now made it possible to share real-time traffic information, opening up new possibilities for implementing Intelligent Transportation Systems (ITS) in urban areas.

Traffic Management Systems (TMSs) constitute a suite of applications and tools designed to enhance overall traffic efficiency and safety. Drawing data from diverse sources like vehicles, traffic lights, and sensors, TMSs play a crucial role in identifying and controlling traffic hazards, thereby improving overall traffic efficiency.

TMSs operate through three key phases: (1) information gathering, involving the collection of traffic-related data from various sources; (2) information processing, responsible for aggregating and analyzing the received data to identify traffic hazards;

and (3) service delivery, providing solutions to control traffic issues and enhance overall efficiency.

With the integration of Vehicle Ad-Hoc Network (VANET) communication technologies, Traffic Management systems enable the development of dynamic solutions to address traffic congestion in real-time. This is exemplified by the Adaptive Traffic Lights Control System (ATLCS), which monitors and accumulates data within a traffic network, offering a responsive approach to traffic congestion issues.

2.5.1 Important Terms definitions

some definitions of important terms related to traffic management and traffic lights control:

- **phase:** combination of green/red signals that allows the flow of vehicle with non-conflicting movements in the intersections

- **phase sequence:** is the ordering of a set of phases

- **Green Wave:** A coordinated timing strategy where consecutive traffic lights along a route turn green in sequence, facilitating uninterrupted vehicle flow.

- **Cycle length:** is the time required for the traffic light to complete a cycle, a cycle is a complete rotation of traffic signals from green, yellow, red to green again**.**

- **Pre-timed Signals:** Traffic signals operating on fixed timing plans without considering real-time traffic conditions**.**

- **lost time:** refers to the time when there are no movements in the intersection. For example: when a vehicle is yet to move despite the change of signal from red to green because of the driver slow reaction

- **Split**: The proportion of the cycle length allocated to each phase.

- **offset**: the time between the beginning of a green signal at the corresponding intersection and the beginning of green signal at the next (Armas et al., 2017)

- **Delay:** the time interval in which the vehicle was not moving in seconds.

- **Queue**: a set of vehicles in close distance that are waiting to move

- **Density**: The number of vehicles per unit length of road.

- **Bottleneck**: A point on the road network where traffic flow is significantly restricted, leading to congestion upstream.

- **flow rate**: is the number of vehicles that pass through a road during a time step, usually per hour.

- **Lane**: A section of the road for organized vehicle movement, regulated by traffic lights assigning specific phases for each direction.

- **user equilibrium**: also called Wardrop equilibrium. is the case where drivers can not improve their travel time regardless which route they choose. the drivers selfishly choose their route and their choice have no effect on the current traffic situation

- **system equilibrium**: concerned with minimizing the overall waiting time. the drivers cooperate to achieve the objective even if it means treated some drivers unfairly

2.5.2 Congestion

Traffic congestion, the primary challenge for smart traffic management, is a condition characterized by the slow movement or standstill of vehicles on roadways due to an excessive volume of traffic. To address this issue, integrated technologies, such as real-time traffic monitoring, predictive analytics, and intelligent traffic management systems, are employed to reduce congestion.

Congestion not only affects economic productivity (Cheng et al., 2020) but it also serves as a significant contributor to the rise in environmental issues as well as the public health of the population (Levy et al., 2010). This is evident in the increased greenhouse gas emissions resulting from congestion (Barth & Boriboonsomsin, 2009)

Efficient traffic signal control, dynamic route guidance, and data-driven insights are integral components contributing to a more streamlined and intelligent traffic flow. These measures aim to optimize the utilization of road infrastructure and enhance the overall mobility experience within urban areas.

These systems utilize data from diverse sources, including sensors, GPS devices, and social media, to dynamically adapt to varying traffic conditions, thereby contributing to a more responsive and efficient urban transportation network

2.5.3 Traffic management services:

Adaptive traffic lights control: TMSs can Utilize V2V and V2I communications to optimize traffic light control, enhancing traffic flow and contributing to congestion reduction. This, in turn, improves travel times and reduces fuel consumption.

Cooperative congestion detection: Leverages V2V communication for real-time detection of traffic congestion. Vehicles detecting congestion can share this information with neighboring vehicles, enabling them to proactively avoid congested areas.

Congestion avoidance: Utilizes VANET communication to recommend alternative routes to vehicles navigating around congested areas. This proactive approach aids in congestion reduction and enhances overall traffic flow.

Accident detection and warning: employs V2V communication to detect and alert vehicles about traffic accidents. This service aims to mitigate accident severity and enhance overall traffic safety.

Route suggestion: This service, facilitated by a central entity like a Traffic Management Center (TMC), suggests alternative routes to vehicles. The goal is to alleviate congestion, improve travel times, and reduce fuel consumption.

Speed adjustment: dynamically adjusts vehicle speeds to optimize traffic flow. The objective is to reduce congestion, enhance travel times, and minimize fuel consumption.(De Souza et al., 2017)

2.5.4 Challenges

Among the challenges of traffic management in smart cities:

Dynamic Traffic Patterns: Real-world traffic is dynamic, and the flows and volume can change rapidly and unexpectedly. imitating the realistic conditions of traffic can be hard to achieve.

Large Dimensionality Complexity: Managing the problems of large-dimensional systems, particularly the complexity introduced by multi-agents demanding high computation power.

Coordination at Intersections: Achieving optimal coordination between traffic lights at different intersections is complex, especially in large urban areas.

Limited Data Availability: Lack of comprehensive and real-time data can hinder the accuracy of traffic predictions and control strategies

Data quality and overhead: It is crucial to ensure the accuracy and currency of traffic-related data utilized by TMS. Additionally, TMSs often generate substantial data traffic, placing strain on communication infrastructure and impeding scalability, particularly in large deployments.

Security & Privacy: Protecting TMSs from cyberattacks and Ensuring that the privacy of individuals is protected(De Souza et al., 2017)

Non signalized intersections: in real life the fact that there may bo a non signalized intersection poses a problem in the real world (Qadri et al., 2020)

2.6 Conclusion

This chapter Explored the concepts of Smart Cities, Intelligent Transportation Systems (ITS), Vehicular Ad-Hoc Networks (VANET), and Intelligent Traffic Management, with a focus on traffic congestion problem and traffic lights control. Definitions of a "smart city" were explored, accompanied by case studies showcasing successful implementations in cities like Singapore, Barcelona, Dubai, and Amsterdam.

The discussion progressed to the evolution of ITS and VANET, emphasizing their roles in addressing urban mobility challenges. VANET characteristics, components, communication types, and applications were detailed, highlighting their contributions to traffic management, safety, and driver experience.

In the context of Intelligent Traffic Management, key terms and definitions were outlined, illuminating the complexities of traffic signal control, congestion, and the impact of technology on these challenges. Traffic management services, including adaptive traffic lights control, were explored.

However, challenges were acknowledged, ranging from data processing and security issues in VANET to the dynamic nature of real-world traffic patterns and coordination complexities in traffic management. Despite these issues, the chapter underscores the transformative potential of technology in shaping more efficient, sustainable, and livable cities for the future.

Chapter 3 Adaptive Traffic Light control

3.1 Introduction

Traffic light control in the context of ITS is divided into two categories: Fixed and adaptive systems. Fixed control follows predetermined timings and sequence for green, yellow, and red phases, often based on historical data or average traffic expectations. On the other hand, adaptive control utilizes real-time data and wireless communications to dynamically adjust signal timings, considering factors like traffic volume and congestion levels. Adaptive systems allow the employment of various algorithms to continuously optimize signal timings, ensuring responsiveness to changing traffic patterns. This chapter defines the approaches in adaptive traffic light control, followed by classifications of related works in smart city Adaptive traffic lights control systems

3.2 Definitions

3.2.1 Adaptive traffic lights control as an optimization problem

Adaptive traffic lights control is a problem where we aim to minimize congestion by optimizing the timing and sequencing of traffic lights, which make it an optimization problem. Traffic light control can be formulated as both a discrete and continuous optimization problem, depending on how the system is modeled. In a discrete optimization approach, the focus is on selecting specific states for the traffic lights (red, green, yellow) at discrete time intervals.

as a continuous optimization problem, adaptive traffic lights control involves treating the timing and duration of traffic light phases as continuous variables. Instead of having fixed, discrete states for the traffic lights (red, green, yellow), you allow for smooth transitions between these states, considering time as a continuous parameter.

3.2.1.1 Continuous optimization

Continuous optimization involves finding the maximum or minimum value of an objective function in which the variables of the function are real numbers. In order to minimize or maximize the objective function, we usually assume that the function is differentiable and convex. Common techniques for solving continuous optimization problems include methods like gradient-based optimization (e.g., gradient descent) and nonlinear programming. These methods involve iterative processes of adjusting variable values to converge towards the optimal solution. A continuous optimization problem can be modeled as:

$$Min(max)f(x) \tag{3.1}$$

Where decision variables x can take any real value within a specified ranges and f is the objective function Subject to:

$$h_i(x) = 0, i \dots\dots N$$

$$g_i(x) \leq 0, i \dots\dots N$$

Where h is the equality constraint and g is the inequality constraint (Andreasson et al., 2020)

3.2.1.2 Discrete optimization

Discrete optimization is a relatively recent mathematical field, formally defined in the 1950s. However, instances of Discrete optimization problems predate its formal establishment. The journey began in 1784 with the assignment problem, initially studied by Monge, who referred to it as the transportation problem and treated it as a continuous optimization challenge. Despite this, the first algorithm addressing assignment problems wasn't published until 1946 by Easterfield.

Discrete optimization draws from diverse roots, owing to its direct inspiration from real-life problems. For instance, the traveling salesman problem, aimed at finding the shortest paths, or job assignment problems, all contribute to this field. Essentially, discrete optimization involves mathematically modeling problems as the minimization or maximization of an objective function, with the feasible solutions being a finite set of discrete objects (Parker & Rardin, 2014)

$$Min(max) f(x) \tag{3.2}$$

$$Where\ x \in F$$

Where x here is and arrangement or a specific combination of objects, F is a set of combinations, and f is the objective function.

Similar to Continuous optimization, Discrete optimization may have constraints where the variables are restricted to discrete values or combinations.

3.2.2 Metaheuristics

Meta, as defined in oxford dictionary, is a word of Greek origin that means 'after' or 'across' and the 'meta' in metaheuristics means 'higher'. While the word Heuristic came from the Greek word 'heuriskein' which means 'Find'. By combining it we understand that metaheuristics is a higher level and more general heuristics, or heuristics about heuristics. Another difference between the two is while Metaheuristics are categorized to many classes (Gendreau & Potvin, 2005), we have:

- **Single solution-based metaheuristics** in which we take one solution from search space and improve it gradually. Examples: simulated annealing and Tabu search.

- **Population based metaheuristics** Population-based metaheuristics, primarily inspired by natural phenomena, operate by generating numerous solutions and iteratively modifying them until a predefined condition is met. Among the most renowned algorithms in this category are evolutionary algorithms, extensively studied in Discrete Optimization problems due to their efficient time complexity and ability to yield superior solutions compared to conventional approaches

Unlike heuristics, the term meta-heuristics refers to the approximation algorithms that are not designed to solve a specific problem but can be scaled to solve a variety of optimization problems.

Many metaheuristics problems are inspired by natural phenomens, such as evolution and the collective behavior of animals.

Those algorithms can be combined with each other by taking specific proprieties from two or more algorithm depending to what the specific algorithm require. This is what is called: hybrid metaheuristics (Blum, n.d.)

Metaheuristic also proved to be a more practical approach to solve combinatorial optimization problems than the classic approach such as dynamic programming and iterative methods, because the latter while guaranteeing finding the optimal solution the approach is only applicable in small sized problems. On the other hand Metaheuristics based approaches are capable of finding at least a good solution even if the problem is not small sized, and in a reasonable time (Bianchi et al., 2009)

Although researchers have been using ‘metaheuristic’s methods since the 1940s (Sorensen et al., 2017), The first one to coin the term meta-heuristics was Glover in 1986.

Next, we will talk about popular metaheuristic algorithms used in traffic light control and define them.

3.2.3 Genetic algorithms

Genetic algorithm is one of nature inspired population algorithms that use evolutionary techniques such as selection and mutation to generate new ‘better’ solutions based on the former solutions. Popularized by john Holland and his students in 1970s, genetic algorithms have remained in the theoretical area until 1980s, and have been since an important technique to solve a wide range of difficult optimization problems in a fast and reliable manner (Yang, 2010)

One of the important aspects of genetic algorithms is that it allows a lot of freedom in designing the algorithm, from the encoding of the population to the method of recombination and mutation or in choosing the different parameters that affect the functioning of the algorithm.

A simple genetics algorithm consists of:

- **Generating initial population**: First we randomly generate solutions, the population size should be chosen experimentally, there is no universal rule on the best value of this parameter. We represent the solution as chromosomes, this representation of a single solution differs depending on the problem at hand, usually we use byte encoding with binary values to encode the solutions.

- **Selection candidate solution**: to produce the next generation of solutions, we need to choose the best candidate (or the parents to breed the new generation of solution), while choosing the best – the most fit according to our fitness function-solutions is a good heuristic, sometime it is best to allows the inclusion of bad solutions to avoid being stuck on a local optima. There are many method of Selection, for example: roulette wheel, rank selection, elitist selection, tournament..etc.

- **Recombination and Mutation**: after choosing the parents, we start the process of ‘breeding’ between them, this is done usually by the crossover method. Where we choose a point to split the strings of two parents and exchange the values of those parents. Or 2-points crossover where we split the string into 2 parts. Adding more points of crossover is not recommended as it make it harder to produce good solutions (Whitley, 1994). mutation is the process of changing one bit or more of a solution using low probability, it is useful as it add more randomness to the new population of solutions and helps moving into the global optima and avoiding the premature convergence problem (Yang, 2010)

- after generating the new solution, we repeat the process iteratively until we meet our criteria of stopping (finding the optimal solution or reaching the maximum of iteration)

we see detains of GA implementation in the pseudocode in Figure 3.1

Genetic Algorithm

Objective function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, ..., x_d)^T$
Encode the solutions into chromosomes (strings)
Define fitness F (eg, $F \propto f(\boldsymbol{x})$ for maximization)
Generate the initial population
Initialize the probabilities of crossover (p_c) and mutation (p_m)
    while (t <Max number of generations)
        Generate new solution by crossover and mutation
        Crossover with a crossover probability p_c
        Mutate with a mutation probability p_m
        Accept the new solutions if their fitness increase
        Select the current best for the next generation (elitism)
        Update $t = t + 1$
    end while
Decode the results and visualization

Figure 3.1 a genetic algorithm (Yang, 2010)

3.2.4 Particle swarm Optimization

First developed by Kennedy and Eberhart in 1995, PSO takes inspiration from the social interactions of swarms, like fishes and birds, in search for food (Poli et al., 2007). It is quite popular because of its simplicity and effectiveness. Unlike GA, there is no need to encode the particles as strings and every individual is represented by a vector that denotes the current position and another vector that denotes the velocity. The swarm moves in the search space toward the optimum and every particle chooses its next direction based on its best position p_{best} (best local position) and the swarms' best position best global position or g_{best} for short, the best position is determined by the objective function.

Every particle has three components: the current position x_i, the current velocity v_i and the best position p_{best}. All represented by a D-dimensional vector where D is the dimensionality of the search space (Poli et al., 2007) The velocity v_i is updated by taking into account both the best position and the global best position by the equation:

$$V_i^{t+1} = V_i^t + \alpha€_1[g_{best}^t - x_i^t] + \beta€_2[p_{best}^t - x_i^t] \tag{3.3}$$

Where $€_1$ and $€_2$ are random vectors whom values are between 0 and 1. α and β are the accelerating constants, choosing the value of the constant depends on the problem but generally $\alpha \approx \beta \approx 2$. and after updating the velocity we update the position. In many variants of PSO, we multiply V_i^t by an inertia weights that take usually the value of 0.9 to make the motion of the swarms more stable and converge quickly (Yang, 2010)

$$x_i^{t+1} = x_i^t + V_i^t \tag{3.4}$$

and when a particle find itself in a better position x_i than its last best local position $pbest$ it updates the new position by the equation:

$$p_{best} = x_i \tag{3.5}$$

And the next global best position g_{best} is updated based on the best of all local best positions.

Like GA, the process is repeated iteratively until we arrive at the optimal solution or we reach the max iteration.

Figure 3.2 provides the pseudo code of PSO

Particle Swarm Optimization

Objective function $f(x)$, $x = (x_i, \dots x_d)$
Initialize positions x_i velocity v_i of n particles
Find g_{best} from $min\{f(x_1), \dots, f(x_n)\}$ (at $t = 0$)
while (criterion **)**
  for loop over all n particles and d dimensions
  generate new velocity v_i^{t+1} using equation (3.3)
  calculate new positions x_i^{t+1} using equation (3.4)
  evaluate objective functions at new positions x_i^{t+1}
  find the current best for each particle p_{best}
  end for
  find current global best g_{best}
  update $t = t + 1$ (pseudo time for iteration counter)
end while
output final results p_{best} and g_{best}

Figure 3.2 a PSO algorithm

3.2.5 Bi-Level Frameworks

It is the optimization of two interconnected problems that interact with each other to solve said problems (Stoilov et al., 2015). In bi-level framework we have an upper level (also called leader) and lower level (the follower). The optimization function of one affects the optimization function of the other.

In traffic congestion optimization, the upper level usually is concerned with optimizing traffic signal scheduling, while the lower level solves the traffic assignment and vehicle routing problems in aim to achieve network equilibrium.

Solving a bi-level network design problem using exact solution methods is very difficult because the problem is NP-hard (Ben-Ayed et al., 1988) which is why metaheuristic and other computational intelligent artificial methods are the preferred choices

3.2.6 Deep learning

Deep learning algorithms are a subset of machine learning algorithms that use multi-layers neural networks to learn new representations of the data with higher level of abstractions, or transforming the data non-linearly into a new space (Goodfellow et al., 2016)

The term 'deep' comes from the utilization of multiple network layers, as seen in Figure 3.3, each potentially dedicated to learning specific aspects of the data. For instance, in image processing, it can be generally stated that the initial layers capture edges, while subsequent layers focus on colors and more complex features.

Although deep learning dates back to 1967 when Alexey Ivakhnenko introduced it, but its widespread adoption was hindered by limited computational capabilities

Notably, LeCun utilized a deep learning algorithm in 1989 to recognize handwritten zip codes, a task that took three days (LeCun et al., 2015). However, recent advancements in computational power and data availability have propelled deep learning to the forefront of artificial intelligence in recent years. Ranging from convolutional deep networks to autoencoders, deep learning has significantly advanced the state-of-the-art in numerous well-known problems, whether supervised, unsupervised, or semi-supervised.

This chapter will focus exclusively on deep reinforcement learning, as it stands as the most utilized Deep learning approach in combinatorial optimization.

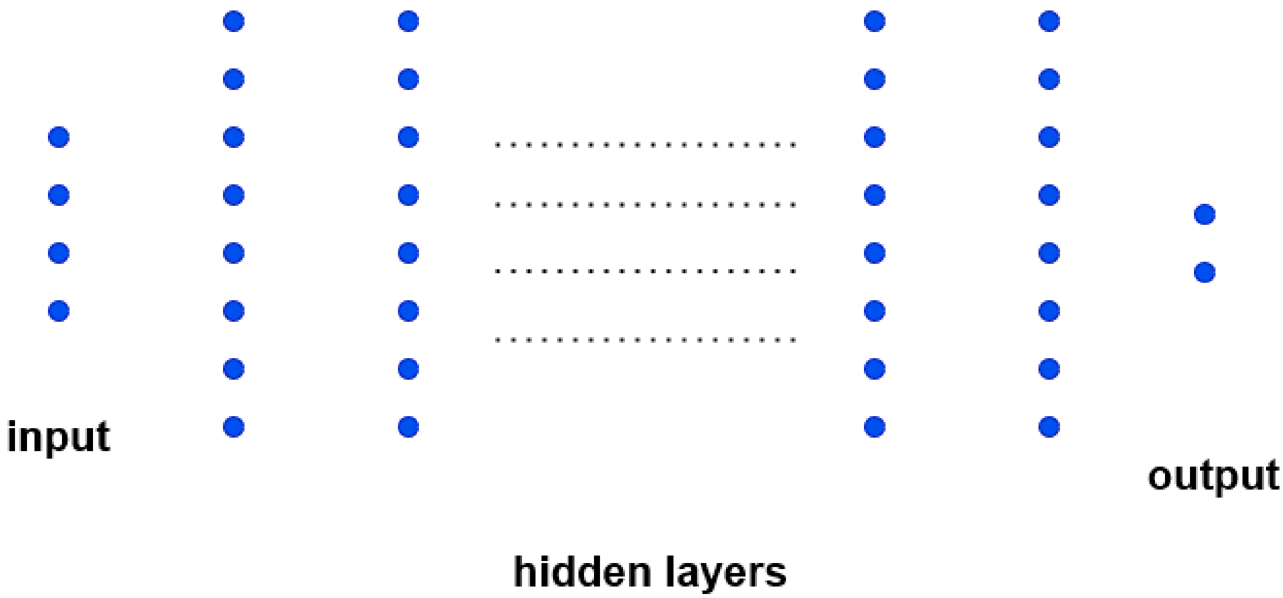

Figure 3.3 a deep neural network

3.2.7 Deep Reinforcement Learning

Reinforcement learning, a semi-supervised learning approach, finds widespread application in robotics, game theory, and combinatorial optimizations. In this paradigm, an environment is engaged with one or multiple agents, where these agents interact with the environment and receive feedback as rewards or penalties influencing subsequent actions as illustrated in Figure 3.4. RL achieves a balance between exploration and exploitation: agents exploit accumulated experience to decide their next actions, while also exploring by occasionally making random choices. Diversifying an agent's actions is particularly valuable in optimization scenarios, preventing the agent from being trapped in local optima

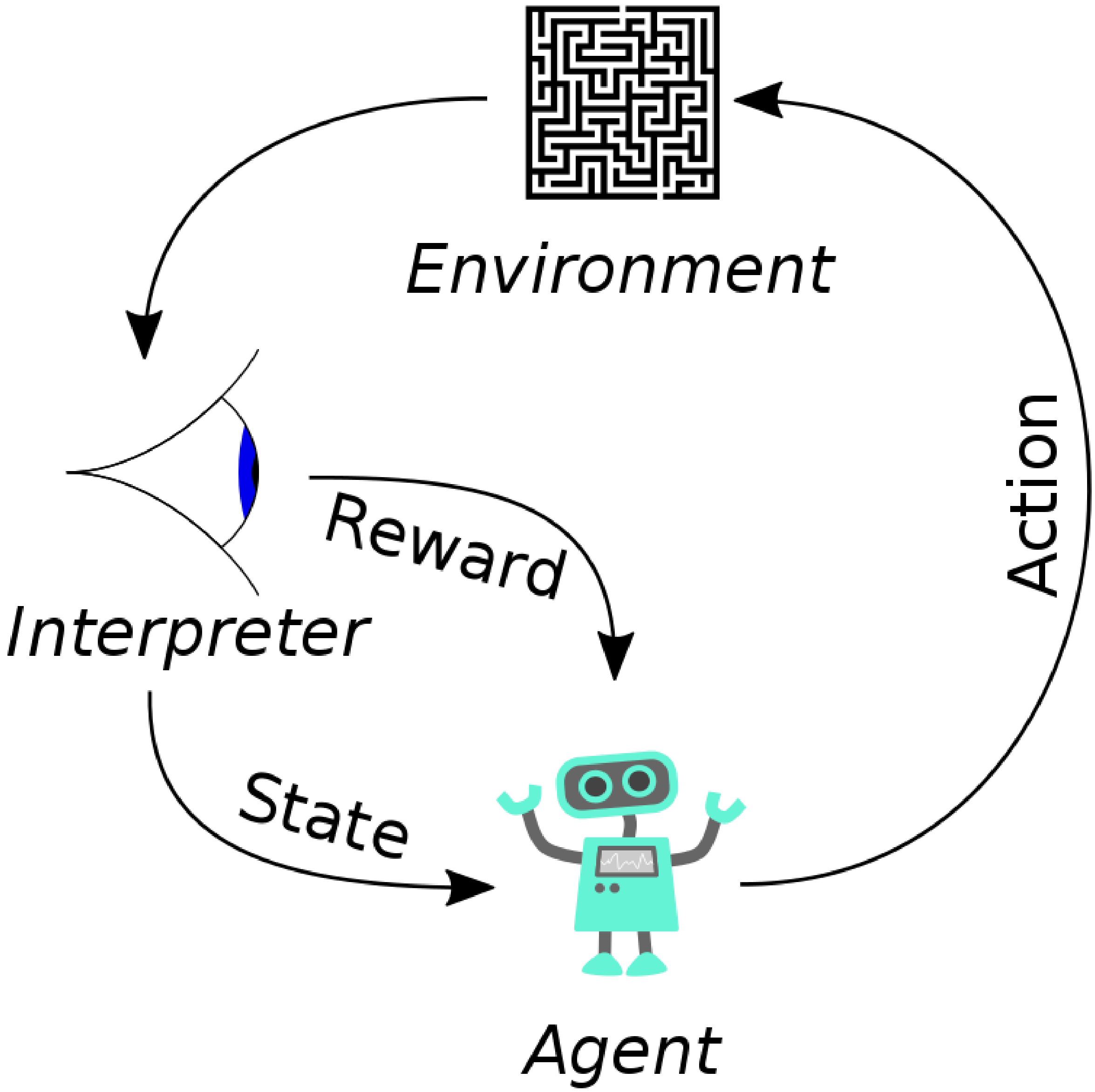

Figure 3.4 reinforcement learning visualization ("Reinforcement Learning," 2024)

- **Markov Decision Process**

To model reinforcement learning environments, we use Markov Decision Process (MDP). We say that a sequence of states is Markov if moving to state S_{t+1} depends only on state S_t. A MDP is a set (S, A, P, R, γ) where:

- ***S*** is a finite set of states
- ***A*** is a finite set of actions
- γ ∈ [0,1] is called a discount factor,
- **P** is the probability of transition from state s to s' after an action

$$Pa(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a) \tag{3.6}$$

- The reward of transition with action:

$$R: SxA \rightarrow R \tag{3.7}$$

In MDP, the transition from s_t to s_{t+1} depends on both the state and the current action A_t

Reinforcement learning have three sub-elements (Sutton & Barto, n.d.):

- A policy that specifies what action the agent is going to take given the state he is currently in.

$$\Pi_t(S_t = s | A_t = a) \tag{3.8}$$

- A reward R that takes the form of a single number, a reward is what gives an agent the clue if the actions it is taking are good or bad. The objective of an agent is to maximize the rewards by choosing the right actions.

The total reward is expressed as:

$$G = \sum_{k=0}^{H} \gamma^k R_{k+t+1} \tag{3.9}$$

- A value function V that differs from a reward in the sense that it gives the long-term benefits an agent can expect in its current state. Whereas the reward is all about immediate feedback.

3.3 Categorization of Adaptive traffic lights control

Efforts to alleviate congestion in smart cities through traffic management are typically classified into three segments: traffic lights control, vehicle routing, and a blend of both. Within the realm of traffic lights control, two main strategies exist: fixed control and adaptive control. This thesis focuses on methodologies associated with adaptive traffic lights control, specifically emphasizing the optimization of phase timings at intersections. The goal is to effectively manage the direction and flow of vehicles in incoming lanes, with the aim of mitigating congestion.

In the next section, we will delve into a discussion of prior works, exploring their approaches to traffic lights control

3.4 Related works

3.4.1 Metaheuristics based approaches

To render the proposed algorithms for traffic signal optimization as practical as possible, researchers employ network simulators designed to replicate real-life network traffic. Notable simulators include SUMO, NCTUns, PARAMICS, CityFlow, among others.

The evolution of traffic control algorithms began with statistical methods in the 1960s (Webster, 1958). However, the surge in computational power has ushered in numerous computational intelligence-based approaches, particularly with the growing popularity of biologically inspired algorithms over the last few decades. These algorithms combine efficiency with the capability to attain global solutions for complex problems.

The Genetic algorithm stands as a prevalent method in traffic control settings. (Foy et al., 1992) were the earliest in utilizing the Genetic algorithm for optimizing traffic signals in a fixed-flow four-intersection network. Their approach involved car volume on lanes as input and decision variables such as green time, first phase direction, cycle length, and intersection offsets.

(Jee-Hyong Lee & Hyung Lee-Kwang, 1999) implemented a fuzzy controller for adaptive phase length and sequence optimization in intersections, while (Chou & Teng, 2002) presented a more comprehensive fuzzy controller system, considering factors like vehicle and street lengths, number of junctions, and lanes.

(Ceylan & Bell, 2004) used Genetic algorithm to solve a bi-level traffic optimization problem where the upper level is signal timing and the lower level is finding equilibrium link flows based on the stochastic effects of drivers' routing.

(Srinivasan et al., 2006) utilized a distributed multi-agent Neural Network coordinated by the PARAMICS simulator, optimizing delay and stoppage times in Singapore's congested areas. Unlike other approaches, they used simultaneous perturbation stochastic approximation to update neural network parameters instead of genetic algorithms

Their experiments were run times each one with different duration (3 hours, 6 hours, 4 hours), the hybrid multi-agent neural network performed remarkably well in the 24 hours scenario compared to other tested methods. Their algorithm managed to reduce the mean delay and the stoppage time by 78% and 85% respectively.

(Teo et al., 2010) employed a genetic algorithm at a T shaped intersection with three traffic phases. Their algorithm aim was to minimize queue length by determining the optimal green time taking into consideration the cycle time and amber time.

Many research methods used a combination of algorithms. (Sanchez-Medina et al., 2010) approach was a Combination of Genetic algorithm, and inside the fitness function a CA-based simulator to simulate overtaking and multiple lanes, and Beowulf cluster as a parallel computing system MIMD to make the system scalable. The objective was to

Optimize the traffic-light Cycle, meaning optimizing the duration of each traffic signal (green, red, or yellow) in seconds in order to achieve the objective. They represented the solutions (chromosomes) as gray binary encoding rather than binary encoding and used a combination of elitism and truncation for selection.

The authors tested four different fitness functions: number of vehicles that left the network during the simulation interval, average travel time of the vehicles, time of occupancy (TOC), and global mean speed. The reason for trying different objective functions is to cover all possible scenarios and make their model as realistic as possible.

Their approach was Highly scalable and can be applied in a large congestion network. they used the traffic network of Satna de Cruz city in Spain as their case study.

(Zhen-Jin Huang et al., 2010) used a neural network controlled by a Genetic algorithm to minimize average waiting time by regulating green light and cycle durations.

A vehicle-to-vehicle communication based approach was used in (Maslekar et al., 2011) who proposed an adaptive traffic control using VANET by computing the density of vehicles in an intersection and using this information in timing the traffic lights accordingly using NCTUns simulation tools. The method calculated the optimal cycle time by modifying Webster's formula. results showed improvements over pre-timed method both in waiting time and vehicle stopping.

in (García-Nieto et al., 2012), a PSO algorithm optimized a cycle program using the number of vehicles passing through the network as the fitness function. They favored PSO due to its rapid convergence and straightforward implementation, adjusting green and red times based on traffic volume at intersections in cities such as Malaga and Sevilla, Spain.

Two approaches were applied in (Putha et al., 2012) where the authors used both the genetic algorithm and Ant Colony Optimization (ACO), they applied both the algorithms, to optimize traffic signals settings to maximize the number of passing vehicles in the network in an oversaturation period. The algorithms were applied to two different models, a simple model and a more realistic and complex model. By comparing the results to reach the conclusion that ACO performed better and has less variance in randomized trials.

Vanet was also used in (Kwatirayo et al., 2013) who proposed a custom adaptive traffic control algorithm, modifying green light time in real time based on accumulated information about lanes and traffic density used SUMO simulator, the results showed improvements over Pre-timed approach in reducing waiting time in their test on city of Moncton.

The authors in (Pandit et al., 2013) represented the traffic control problem as a job scheduling problem. Their adaptive traffic control by optimize green time using data from vehicles such as speed and position with the help of VANET.

They created an algorithm with the name oldest arrival first OAF that uses the speed and position of the vehicle to extend green time. The used INET/OMNET+ simulator

The authors extended the Webster's equation to calculate the optimal green time to accommodate to their adaptive approach.

(Hajbabaie & Benekohal, 2015) Tried to make the model as realistic as possible by adding variable and constraints that represents traffic accidents and unforeseen events, using the Used CORSIM simulator, the authors proposed a bilevel framework for traffic timing and traffic assignment with the genetic algorithm to dynamically optimize signal settings

A combination of cellular automaton mechanism and particle swarm optimization (IOCA-PSO) was proposed in (W. Hu et al., 2016) to optimize traffic optimization in real-time By controlling the timing of the signals and the phase sequence to minimize waiting time. The authors choose particle swarm optimization due to the PSO being a faster computing speed and its ability at finding good global solutions. They tested their approach and found that the algorithm performed better than GA and simple PSO in the case of increasing the number of vehicles or the number of intersections.

In (Fujdiak et al., 2016), the authors applied a genetic algorithm to manage static and dynamic traffic in the city of Brno, Czech Republic. Utilizing data sensors to share traffic information, they found the algorithm to be less effective under high-density traffic, especially when using average delay time as the cost function.

The authors in (Armas et al., 2017) utilized an evolutionary algorithm incorporating elitism and mutation with an integer-based representation in their solution. Their approach was tested using the MATSim simulator, revealing the importance of a higher mutation probability for green time compared to cycle and offset.

Another hybrid approach was used in on (Srivastava & Sahana, 2017), where the authors combined Genetic Algorithm and Ant Colony Optimization where The GA is nested inside ACO to speed up the convergence process in the upper level and the lower layer is for stochastic user equilibrium for which road network is designed using Petri Net (PN).

(Liu et al., 2020) used bilevel optimization framework to minimize the average travel time where the upper-level adaptive evolutionary differential technique for signal timing and the lower level a combination of ant colony algorithm for vehicle routing and frank-wolf for flow shifting. They Used different shapes of networks with varying number of intersections.

Same goes in (Stoilova & Stoilov, 2020) where the lower level is concerned with minimizing the queue length while the upper level aims to optimize Cycle timing using realistic constraints, rather than fixed values, this paper propose adapting the timing of traffic signal in real time.

3.4.2 Reinforcement learning based Approaches

Reinforcement learning has been applied in traffic management scenarios since 1994 (Mikami & Kakazu, 1994). Its inherent characteristics, involving the modeling of an environment and its interaction, align well with the complexities of traffic control, particularly in real-time adaptive traffic signal control.

Central to reinforcement learning lies the challenge of devising optimal representations for states, actions, and rewards.

For instance, (Abdulhai et al., 2003) depicted states as the elapsed phase time and queue length from different lanes. The decision-making process pertained to altering the current signal, with rewards tied directly to the delay experienced by vehicles in the network. Their utilization of the Q-learning algorithm showcased significant advantages over pre-timed approaches, notably excelling in managing variable traffic flows across three diverse scenarios: uniform, constant, and variable.

Alternatively, (Arel et al., 2010) adopted a more simplified representation where the state comprised a vector denoting relative delays for all lanes within an intersection at each simulation step. Actions involved selecting subsequent phases, while rewards were determined by the disparity in delay between consecutive points. Employing a neural network for Q-learning approximation within discrete environments across five intersections, their study exhibited promising results in key performance metrics compared to their prior research endeavors.

In the pursuit of refining reinforcement learning methodologies for traffic control (El-Tantawy et al., 2014) experimented with diverse formulations for states, actions, and rewards. They discovered that augmenting the representation vector with queue lengths during red phases and vehicle arrival times during green phases yielded superior results in minimizing delay, a key objective in stochastic traffic control.

Their comparison of fixed and dynamic phase sequences revealed the efficacy of the former in achieving optimal policies, attributing this advantage to the ability to focus on policy optimization and extend green phases. The adaptive approach showcased favorable results against pre-existing fixed and actuated traffic control strategies.

Similarly, (Mannion et al., 2016) explored various RL approaches centered on the Q-learning algorithm. Their state representation, a vector of dimensions (2+P), incorporated maximum queue lengths for each phase, alongside indices denoting the current phase and its elapsed time. Evaluating actions concerning phase extension or transition, their findings demonstrated enhancements in managing stochastic vehicle flows compared to pre-timed schemes.

In a multi-agent context, (Van der Pol & Oliehoek, 2016) proposed coordinating multi-agent deep Q-learning for traffic control across multiple intersections. They mapped intersection images to an 8x8 matrix, feeding them into a convolutional network. To address delayed rewards, they formulated a weighted reward function integrating delay, vehicle waiting times, emergency stops, traffic jams, and traffic light changes.

Expanding on the success of deep learning in image processing, the integration of deep neural networks into traffic control has become a prevailing trend within artificial intelligence. (Mousavi et al., 2017) incorporated policy gradient techniques to optimize policy and action-value functions. Their state representation consisted of intersection snapshots at discrete time steps, leveraging policy gradients within a framework based on the SUMO simulator.

Similarly utilizing images as input to their algorithm, (Wei et al., 2018) employed deep Q-learning (DQN) where the state representation encompassed vehicle counts, waiting times, queue lengths, images from current traffic conditions processed through convolutional layers, and phase information. Their innovative strategies, including phase gates to ensure fair treatment of all phases and the application of the Memory Palace theory to combat overfitting, outperformed established pre-timed and actuated methodologies in both synthetic and real-world data scenarios.

(Liang et al., 2019) explored real-time traffic control via Deep Q-learning, utilizing a tensor representation comprising two matrices depicting traffic light conditions, vehicle positions, and speeds. Their novel double dueling deep Q network (3DQN) significantly outperformed conventional approaches in both reward optimization and learning efficiency.

Furthermore, (Wei et al., 2019) introduced PressLight, a unique reward representation in RL for traffic lights, emphasizing differences between incoming and outgoing traffic in lanes. Their state representation involved lane vehicle counts and current phase, while actions focused on phase transitions.

In their work (X. Hu et al., 2020) a novel methodology integrating Graph Neural Networks (GNNs) was employed to retain structural information within the network, distinguishing itself from conventional approaches. Their utilization of reinforcement learning (RL) aimed to optimize green timing for traffic lights by leveraging real-time traffic data. An essential element of their study was the adoption of a reward metric derived from prior research: the pressure of a lane, calculated using the formula:

$$P_i = N_{in} * \left(1 - \frac{N_{out}}{N_{max}}\right) \tag{3.10}$$

Here, p signifies the traffic pressure, N_{in} represents the number of incoming vehicles, N_{out} signifies outgoing vehicles, and N_{max} denotes the maximum vehicle capacity of a lane.

In their work presented in (Shijie & Shangbo, 2023), the authors introduced the 'Friend-Deep Q Network,' a cooperative multi-agent approach for traffic signal control in 2–4 junction scenarios using SUMO simulations. This approach demonstrated superior performance compared to both independent Q-networks and fixed networks.

3.4.3 Limitations

In Table 3.1 we provide the drawbacks of previous works in Adaptive traffic lights control

Table 3.1 Criticism of related works

Paper	Simulation tool	Limitations
(Foy et al., 1992)	Custom simulation model	Limited simulation model: a basic traffic simulation model that does not take into account traffic signal coordination between intersections. leading to less accurate real-world results. Scalability: inability to scale to large traffic networks with many intersections and complex traffic patterns.
(Mikami & Kakazu, 1994)	Custom simulation model	Scalability: The algorithm relies on a central controller to gather performance data from all agents and generate new parameters. make it harder for scaling to large networks Simplified Traffic Model: ex: vehicle speed is constant
(Jee-Hyong Lee & Hyung Lee-Kwang, 1999)	Custom simulation model	performance: performance drops in high traffic flow small improvement over compared approach (vehicle actuated method) (max of 13.5%)
(Chou & Teng, 2002)	Pappis & Mamdani fuzzy traffic control model	Limited simulation: their fuzzy Controller is applied on a simplistic environment with inability to keep its performance in different traffic

		situations
(Abdulhai et al., 2003)	custom simulation model	Scalability to large networks: The algorithm applied on a single intersection and doesn't provide concrete evidence of the algorithm's scalability to large networks with numerous intersections
(Ceylan & Bell, 2004)	TRANSYT	Scalability: their approach as applied on 6 intersections network but no larger network
(Srinivasan et al., 2006)	PARAMICS	Scalability: no proof of scalability to larger networks. Their approach requires re-training the neural network
(Guojiang Shen and Xiangjie Kong 2009)	Custom simulation model	performance: performance degraded in high traffic flow no mention of scalability to bigger networks
(Teo, Kow, and Chin 2010)	Custom simulation model	Limited scope: The algorithm is only designed to optimize traffic flow at a single intersection.
(Zhen-Jin Huang et al., 2010)	GLD (Green Light District)	Limited simulation: Only applied on a single intersection
(Sanchez-Medina, Galan-Moreno, and Rubio-Royo 2010)	Custom simulation model	Limited simulation: The Cellular Automata -based traffic simulator used in the algorithm is a simplified model of real-world traffic behavior computations costs: the set of results took 12 days to obtain.

(Arel et al., 2010)	Discrete model in Matlab	scalability: The algorithm was tested on a specific network topology consisting of five intersections, with one central intersection and four outbound intersections. no application to larger networks
(Maslekar et al., 2011)	NCTUns	Simplified traffic modeling: The algorithm uses a simplified model of traffic flow that might not capture all real-world traffic dynamics Limited evaluation: no evaluation on larger road networks
(García-Nieto, Alba, and Carolina Olivera 2012)	MATSim	Computational cost: the algorithm has been applied to large network and require validation after each iteration, which make it difficult to use the algorithm in real-time applications
(Putha, Quadrifoglio, and Zechman 2012)	Custom simulation model	Scalability: the ACO algorithm relies on pheromone trails to guide ants towards optimal solutions. In larger networks, the number of possible paths and the interactions between them can become more complex
(Kwatirayo, Almhana, and Liu 2013)	SUMO	Scalability: The algorithm is only tested on a single intersection in Moncton, Canada. It is not clear how well it would perform on larger or more complex intersections.
(Pandit et al. 2013)	INET/OMNET++	Scalability: The algorithm is only tested on a single intersection.

(El-Tantawy et al., 2014)	Paramics	benchmark comparison: their algorithm applied on a single intersection and was compared to fixed and actuated traffic control, which are very simple implementation and do not showcase the strength of their approach
(Hajbabaie and Benekohal 2015)	CORSIM	Scalability: the reliance on central hub to collect and process all traffic data for decision-making is only suitable for smaller networks
(Fujdiak et al., 2016)	Discrete Time Simulations DES	Limited scope: The algorithm is only designed to optimize traffic flow at a single intersection.
(W. Hu et al., 2016)	VISSIM	Limited Scalability: Their algorithm is designed to optimize traffic light scheduling for a specific time period and set of traffic conditions. It may not be able to adapt to changes in traffic flow. in Addition, their algorithm has a number of parameters that need to be tuned in order to achieve optimal performance which limit its scalability
(Mannion et al., 2016)	SUMO	scalability: The algorithm was tested on a specific network topology consisting of nine intersections. no application to larger networks

(Van der Pol & Oliehoek, 2016)	SUMO	their DQN algorithm does not keep its superior performance going from single intersection to 4 intersections
(Armas et al. 2017)	MATSim	Computational cost and scalability: the EA is designed for short-term evolution with a small population size and few generations due to the computational expense of microscopic traffic simulation, which makes it harder to apply on larger networks.
(Mousavi et al., 2017)	SUMO	scalability: only applied to single intersection. Limited state representation: The algorithm uses raw pixel values of images as the state representation. which may not capture all of the relevant information about the traffic state, such as the speed and direction of vehicles
(Srivastava & Sahana, 2017)	Custom simulation model	Simplified road network model: their simulation does not account for varying traffic flow or different vehicles.

(Wei et al., 2018)	SUMO	scalability: their approach was applied on a single intersection
(Liang et al., 2019)	SUMO	scalability: the model is only applied to single intersection
(Wei et al., 2019)	CityFlow	computation costs: using to different larger networks require retraining their RL algorithm
(Liu et al. 2020)	Cityflow	computation costs: their implementation of the ADE algorithm and the Ant Colony algorithm require repeated evaluation of candidate solutions and require storing a large amount of information about potential routes for each origin-destination pair, which can be memory-intensive
(Stoilova & Stoilov, 2020)	YALMIP	scalability: applied on 5 intersections but no application on larger networks
(X. Hu et al., 2020)	CityFlow	computation costs: The Graph Neural Network (GNN) used for traffic prediction can be computationally expensive, especially for large-scale road networks. in addition, using different larger networks require retraining their RL algorithm

(Shijie & Shangbo, 2023)	SUMO	scalability: applied to only 4 intersections. scaling to larger network with retraining

3.5 Conclusion

In this chapter, we discussed the formulation of Adaptive Traffic Light Control as either discrete optimization problems or continuous optimization. We provided detailed explanations of popular approaches used in these problems, ranging from metaheuristics like genetic algorithms and PSO to the more recently popular reinforcement learning. Additionally, we included a section on related works using the approaches we defined earlier. Finally, we outlined the limitations and shortcomings we observed within these approaches.

Chapter 4 Proposed Algorithm and obtained results

4.1 Introduction:

In the context of a smart city environment assuming the integrating of V2V and V2I communication within a VANET network, our implementation, Self-attention multi-agent proximal policy optimization (SA-MAPPO), is a reinforcement learning-based algorithm that has demonstrated effectiveness in addressing our problems and delivering satisfactory solutions.

In this chapter, firstly we will define components of our deep reinforcement learning approach and then provide a detailed explanation of our implementation, from the hyperparameters of our algorithm to the architecture of our SAMAPPO model. Finally, we will present the results of our implementation and discuss in detail on those results in both 6 intersections and 12 intersections networks and varying traffic flow.

4.2 Description of the algorithm's architecture and methodology

We can break down our algorithm into three main components: Multi-Agent Proximal Policy Optimization network (MAPPO), Self-Attention network (SA), and Convolutional Neural Network (CNN).

The MAPPO network is responsible for decision-making, determining the actions agents should take. We adopt the Centralized Learning with Decentralized Execution Paradigm for our approach. During training, MAPPO has access to all simulation data through the value function or critic network input. The policy network weights are shared across all agents during training, simplifying implementation and reducing computation costs. Instead of training all agents separately, we focus on training two networks: the policy network and the value function network. This approach offers flexibility for tweaking the hyperparameters of our model (Yu et al., 2021).

The Self-Attention network transforms raw data into useful features before feeding it to the MAPPO network. SA extracts information directly from the environment as a matrix, identifies relevant temporal and spatial connections among agents, and produces an output matrix of the same size for MAPPO.

The CNN serves as the encoder algorithm, aiming to reduce the size of the input matrix without losing significant information. This network is used solely when training on 12 intersections map. In the 12 intersections map, the encoder and SA parts are the only part getting trained, while MAPPO keeps its parameters that were trained from the 6 intersections network in what is called transfer learning

In the next section we will define the technical details of our implementations

4.2.1 Algorithm components

Policy gradients: are a class of reinforcement learning algorithms that harness the power of neural networks to approximate policies that define the agent's behavior within an environment. The primary goal of policy gradients is to optimize these policies using gradient descent to maximize cumulative rewards received by the agent over time.

Actor-critic methods, a subset of policy gradients, combines both the actor and critic components within a single architecture. While the actor is responsible for learning the policy $\pi(a \mid s)$ or strategy, the critic evaluates this policy by estimating the value function associated with state-action pairs.

The advantage function, denoted as $A(s,a)$, represents the relative advantage of taking an action a in a state s compared to the expected value under the current policy π: the advantage function is used to reduce the variance of the estimation of the value function

$$A(s,a) = Q(s,a) - V(s) \tag{4.1}$$

Where $Q(s,a)$ estimates the value of taking a specific action a in a state s following the policy. And $V(s)$ estimation helps in determining the value of being in state s following the policy π, without considering any specific action. By providing feedback on the expected returns, the critic guides the learning process of the actor, facilitating more informed and refined policy updates.

Actor-critic methods offer a significant advantage over other policy gradient approaches by incorporating both policy approximation and value function estimation. This integration enables more stable and efficient learning as the critic's evaluations provide a gradient signal that aids in the continual refinement of the policy by the actor. Through this combination, actor-critic methods aim to achieve a balance between exploration and exploitation, leading to more effective and adaptive decision-making in dynamic environments (Mnih et al., 2016).

Proximal policy optimization (PPO) stands out as a robust actor-critic based algorithm specifically engineered to address the instability issues prevalent in training caused by two key factors: sampling inefficiency and large gradient steps in conventional policy gradient methods.

PPO tackles sampling inefficiency by leveraging a technique that enables more efficient utilization of collected experiences. Instead of discarding or underutilizing valuable data, PPO maximizes the efficiency of these samples by utilizing them effectively throughout the learning process. This approach ensures that experiences are leveraged optimally, contributing to more data-efficient learning.

Moreover, PPO mitigates the impact of large gradient steps that could lead to training instability. It achieves this by imposing a constraint, limiting the size of policy updates during training. This restriction ensures that the policy changes gradually, preventing drastic alterations that might destabilize the learning process. By constraining the policy updates, PPO maintains a balance between exploration and exploitation, enhancing the stability of learning without sacrificing performance (Schulman et al., 2015, 2017). The objective function of PPO is given by Eq. (4.2):

$$L^{CLIP}(\theta) = E_t[min(r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)] \tag{4.2}$$

$$r(\theta) = \frac{\pi_\theta(s)}{\pi_{\theta_{old}}(s)} \tag{4.3}$$

r represents the ratio between the old policy and the new policy, measuring the difference between them as shown in Eq. (4.3). The objective function is obtained by multiplying the ratio with the advantage.

To discourage large updates in case the ratio is too large, the $CLIP$ function is used in the objective function in Eq. (4.2) to choose between the objective function or the clipped version of the objective function. This ensures the algorithm's stability during convergence.

Self-Attention: Self-Attention, a key component of transformer-based neural networks, has gained significant prominence primarily within language models (Dhrisya et al., 2020) yet its utility extends across diverse fields and applications. This mechanism stands out for its ability to process input matrices and effectively allocate attention across different input vectors, enabling the identification of intricate patterns and dependencies within the data.

The self-Attention mechanism operates by performing calculations on input matrices to selectively focus attention on specific input vectors while considering relationships across the entire sequence. By calculating the key, query, and value matrices through weighted transformations of the input matrix, self-Attention effectively learns the relevance and importance of different elements within the sequence. The keys k, queries Q, and values V are computed by applying specific weights W to the input X, allowing the network to filter out essential features and associations within the data (Shaw et al., 2018):

$$k_i = X_i W^k \tag{4.4}$$

$$Q_i = X_i W^q \tag{4.5}$$

$$V_i = X_i W^v \tag{4.6}$$

Next, we perform scaled dot products with the keys and the queries (Vaswani et al., 2017):

$$e_{ij} = \frac{K_i Q_j^T}{\sqrt{d_k}} \tag{4.7}$$

The scaling factor $\sqrt{d_k}$, which is the dimension of K, helps to stabilize the attention scores and prevent numerical instabilities during training. Without this scaling, the dot products can grow very large in magnitude for larger values of d_k, causing the softmax function to saturate and making the gradients very small or zero.

Finally, the $SOFTMAX$ function is applied to obtain attention scores, which are then combined with the values to yield the final output.

$$y_i = \sum_{j=1}^{n} SOFTMAX(e_{ij})V_j \tag{4.8}$$

Encoder: When scaling to the 12 intersections network**,** before the application of the self-attention layer, the subsequent step in our model architecture involved the incorporation of a Convolutional Neural Network (CNN) layer serving as an encoder. CNNs are renowned for their capacity to conduct operations between filters and input data, effectively generating transformed tensors by detecting and extracting meaningful features within the data.

In our specific implementation, the CNN was strategically employed to adjust the dimensions of the input layer to align with the requirements of the Pre-trained MAPPO model. This adaptation aimed to ensure compatibility between the input and the Pre-trained MAPPO architecture without compromising crucial information. The role of the CNN, in this context, extended beyond simple resizing; it performed a compression operation on the input layer. This compression facilitated the transformation of the input data into a format compatible with the self-attention layers and the Pre-trained MAPPO model, thus preserving essential information crucial for subsequent stages of the model's processing.

Effectively, the CNN acted as an intermediary, reshaping the raw input obtained from the environment while retaining pertinent information necessary for the continuity and coherence of the subsequent model operations. This encoding process not only ensured seamless integration between different components of the model but also optimized the input data to align with the size of self-attention layers and the Pre-trained MAPPO model for more effective and streamlined processing.

4.2.2 Implementation

The primary objectives of this implementation involve maintaining high performance across varying traffic volumes, ensuring effectiveness in both low and high traffic scenarios. Additionally, we aim to enhance scalability to accommodate a larger number of agents with minimal adjustments and reduced training costs.

This section will define in detail the representation algorithms central to our approach, offering a comprehensive delineation of our implementation. Subsequently, we will delve into discussions regarding our environments and explain the outcomes resulted from our training process.

4.2.2.1 Representations

The representation of state, actions, and rewards in a reinforcement learning (RL) algorithm is crucial for success, enabling the modeling of the environment and appropriate incentivization of behavior. Our representation is as follows:

State: The input of the self-attention network is a matrix $K(N \times M) = (S_0 \dots \dots \dots .. S_N)$ where N is the number of intersections (or agents) in the simulation and M is the number of features of each agent. Each vector S is represented by:

$$S_i = (currentphaseId, duration, H, X_i) \tag{4.9}$$

Current traffic phase id is the id of the current combination of green, red and yellow lights. $duration$ represents how long the current phase has been active at this intersection. H is the number of halting vehicles in every lane connected to intersection i. , $X_i = (x_{i0} \dots x_{ij} \dots .. x_{in})$ is a vector denoting neighborhood connections, signifies the presence $x_{ij} = 1$ or absence $(x_{ij} = 0)$ of edge connection between intersections i and j.

After passing through the self-attention network, each vector S_i of the matrix K is passed to the policy network separately. as for the value function input, matrix K is transformed into a flattened vector, to which we append another vector $A = (a_0 \ldots\ldots a_N)$ that signifies the current actions of all agents.

Actions: Each agent in the network has two possible actions:

- 1: Extend the current phase by 4 steps.

- 0: jump to the next phase

a constraint was imposed where Each phase cannot have a duration shorter than 4 steps or longer than 60 steps.

Rewards: We use a global reward function that represents the whole network

$$R = (W \times J) \tag{4.10}$$

W is the average accumulated waiting time of all vehicles in the simulation, and J is the sum of average vehicle lengths across all lanes in the previous step. These metrics provide valuable insights into the traffic from the simulation's beginning to the current time step. W captures all waiting time for all vehicles since entering the simulation, whereas J offers a more current perspective. Combining these metrics optimally informs our traffic control algorithm's decision-making for next actions.

For further insight into our implementation, please refer to the pseudocode in Algorithm 1. This particular implementation is applied on the initial network upon which the algorithm is initially trained as visualized in Figure 4.1. When transitioning to a larger network, we adopt a strategy where we freeze all the layers of the MAPPO model and use their pre-trained weights for the larger network. To adapt to the increased input size of this larger network, we introduce an additional encoder layer, implemented as a Convolutional Neural Network as seen in Figure 4.2. This encoder serves the purpose of reducing the input dimensions to align with the original MAPPO network, ensuring compatibility without sacrificing critical information. This approach allows us to effectively leverage the knowledge acquired from the initial network and apply it to more complex networks.

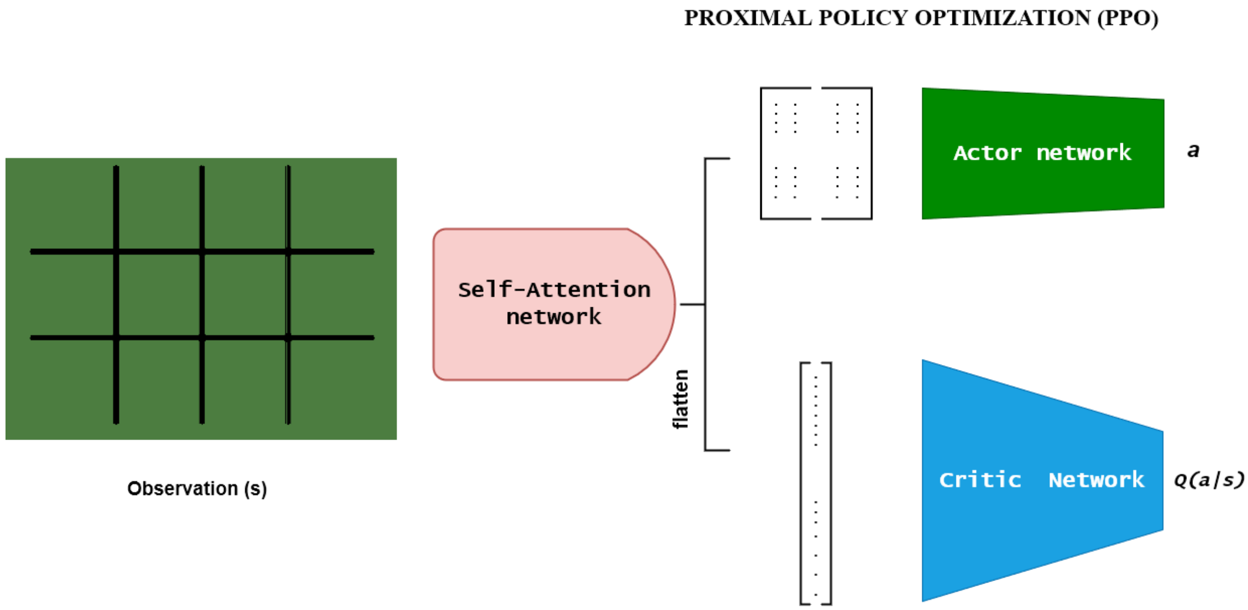

Figure 4.1 Algorithm architecture visualization for 6 intersections network

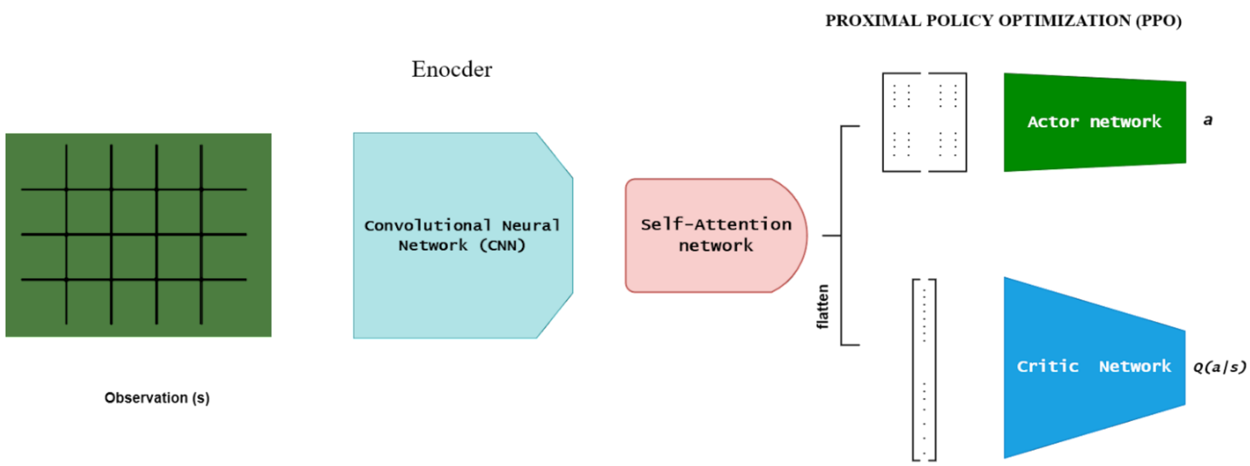

Figure 4.2 Algorithm architecture visualization for 12 intersections network

Algorithm 1 SA-MAPPO

Initialization:

Initialize the Self Attention network parameters *SA*, policy network θ and value network φ.
max episodes *M*
max steps *MS*
memory buffer

for each episode 1 ... *M* **do**
    for each step t….MS **do**
        Collect the states o_t and get the policy network input o'_t from the self-attention network Eq.(4.8)
        for each agent i in environment **do**
            Get next action of this agent a_{ti} from the policy network $\pi(a_{ti} \vee o'_{ti})$
        end for
        Collect the rewards r_t Eq. (4.10) and next states o_{t+1}based on actions a_t.
        Get $V(o_t, a_t)$ from the critic network
        Calculate Advantage estimate Eq. (4.1)
        Add (o_t, a_t, o_{t+1}, r_t) to the memory buffer
    end for
    Calculate the objective function from Eq. (4.2) and Update the SAMAPPO algorithm using Adam Optimizer
    Clear the memory buffer
end for

4.2.3 Hyperparameters:

We define the hyperparameters of our model in Table 4.1

Table 4.1 Values of the hyperparameters used in our proposed models

Hyperparameters	**Value**
Learning rate	1e-4
Optimizer	Adam
Optimizer epsilon	1e-5
Clip coefficient	0.1
Number of layers in MAPPO	2
Feature size in MAPPO	128
Layer normalization in MAPPO	TRUE
Number of heads in the self attention	2
Dropout Vale in SA	0.1
Entropy coefficient	0.01
Value function coefficient	0.1
maximum gradient norm	0.5
generalized advantage estimate lambda	0.9
Gamma	0.8

Mini batch size	60
Update epochs	20
Timesteps	3600
Update frequency	Every 900 time steps
Episodes	20000

4.3 Simulation

4.3.1 Simulation tool

In our study, we utilize the SUMO-RL library, built upon the widely recognized 'Simulation of Urban MObility' (SUMO), a reputable traffic simulation tool. SUMO enables the creation of realistic simulations that mirror various traffic scenarios, including grid networks with different road shapes and varying traffic flows involving different types of vehicles and passengers. This selection is made with the objective of ensuring seamless integration and compatibility with widely used reinforcement learning algorithms.

4.3.2 Environment settings

We examine two networks: a six-intersection and a 12-intersection network illustrated in Figure 4.3 and Figure 4.4. The depicted maps represent the simulations, with the upper section showcasing the six-intersection network, while the lower section exhibits the 12-intersection network. Each intersection in the simulation connects to four edges and encompasses eight incoming lanes

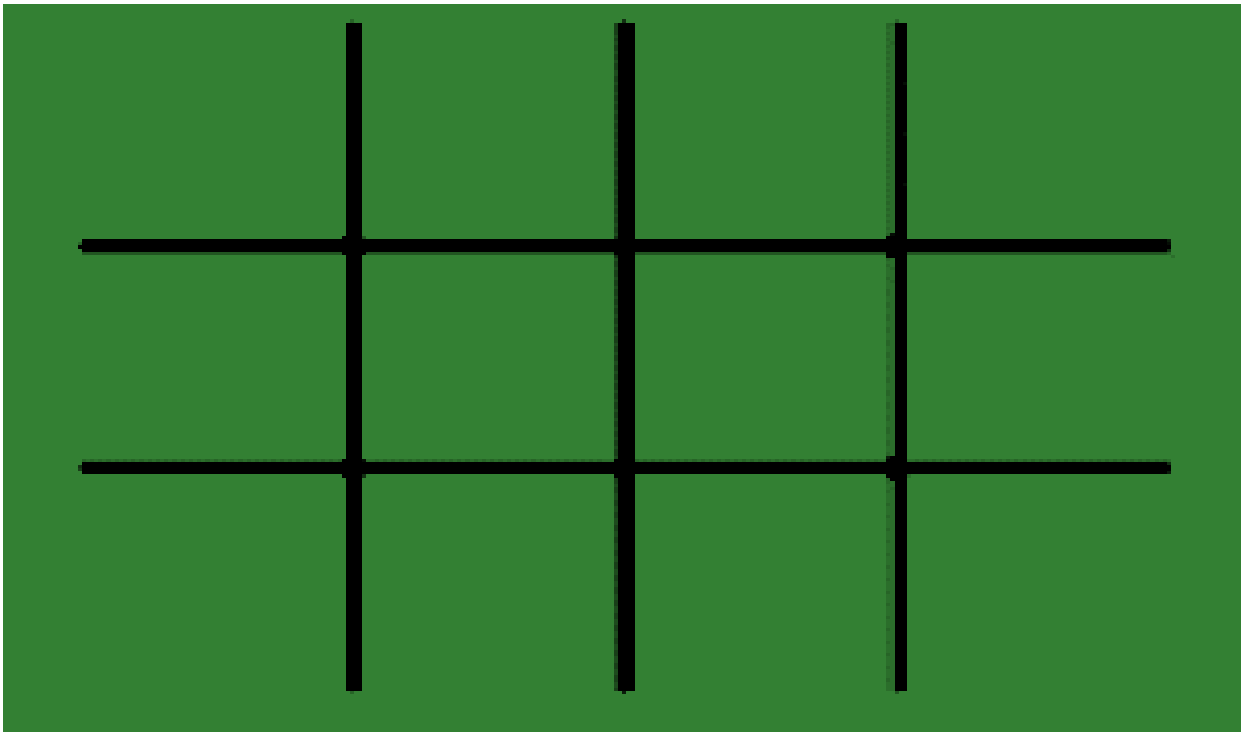

Figure 4.3 six intersections grid network

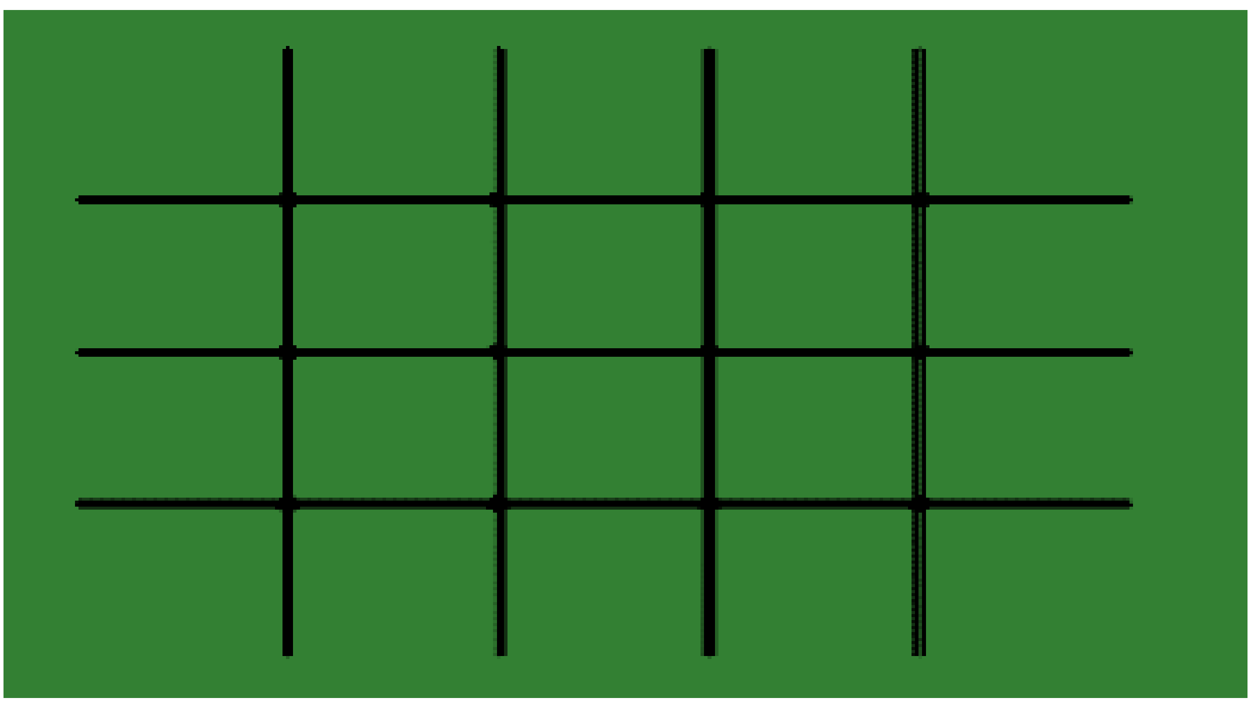

Figure 4.4 twelve intersections grid network

We use two types of vehicles as expressed in Table 4.2

Table 4.2 Proprieties of the vehicles used in the simulation

Type	**Max speed**	**Length**	**probability**
car	35	4.5	0.9
bus	25	12.6	0.1

Lastly, and to simulate rush hours we used two traffic scenarios to test our algorithm, the default scenario here is we have 1 vehicle every 2.5 timesteps, and the higher flow where we have 1 vehicle per 1.7 timesteps

4.4 Evaluation metrics

In our evaluation, we assess the proposed algorithm using four metrics:

- **Accumulated Waiting Time:** This metric aggregates the accumulated waiting time of vehicles in all lanes, recorded at each time step.

- **Average. Waiting:** Calculated as the average waiting time of all vehicles throughout the simulation, computed by summing the average waiting time every step.

- **Average Speed:** This metric represents the mean speed of all vehicles during the simulation, divided by the number of steps.

Each simulation has a fixed duration of 3600 seconds, equivalent to 900 steps. To benchmark our algorithm (SAMAPPO), we compare its performance against the following algorithms:

- **MAPPO:** A multi-agent PPO without self-attention layers.

- **PRESSURE:** SAMAPPO, with pressure-based reward functions, following the approach introduced in (W. Li et al., 2021; Wei et al., 2019)

- **FIXED:** This serves as the baseline algorithm, implementing fixed timing for traffic lights as configured by default in the SUMO simulation tool.

4.5 Training

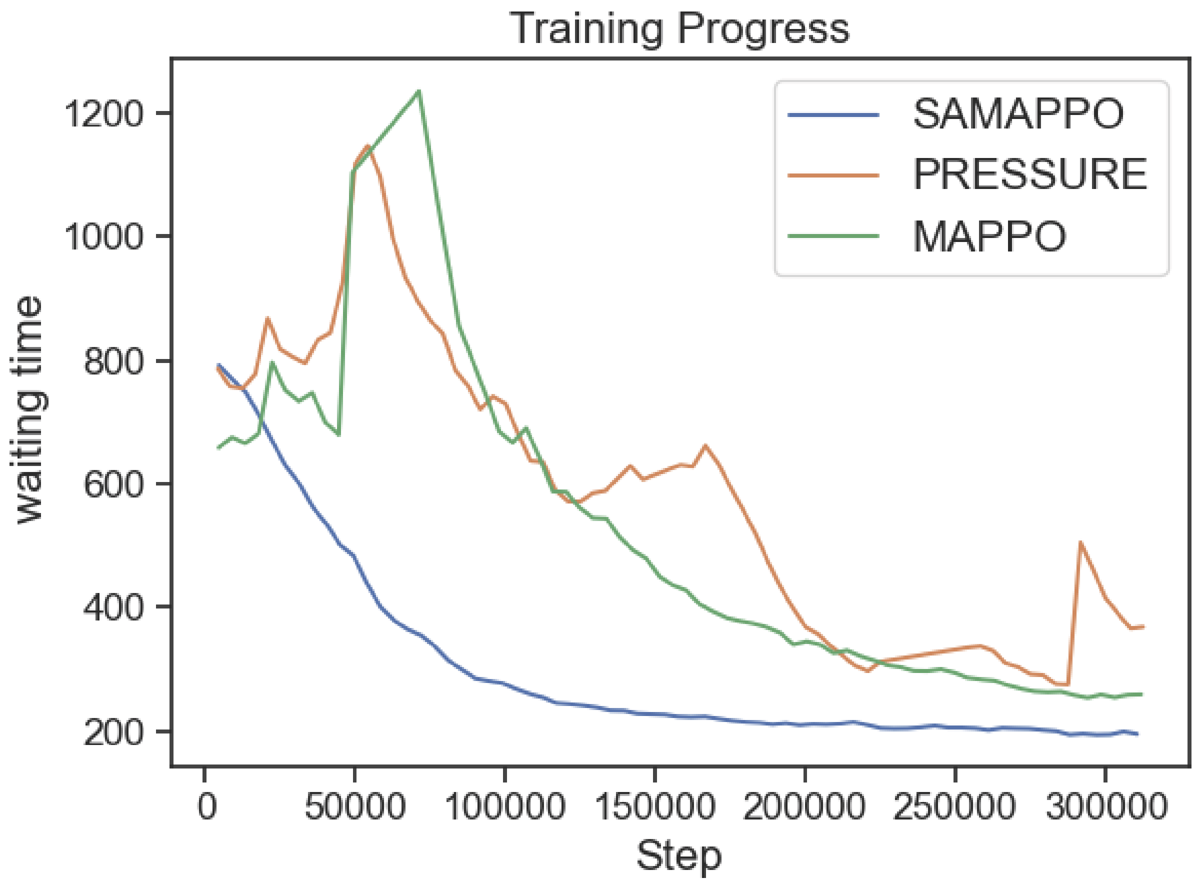

Figure 4.5 training line plot for waiting time in 6 intersections

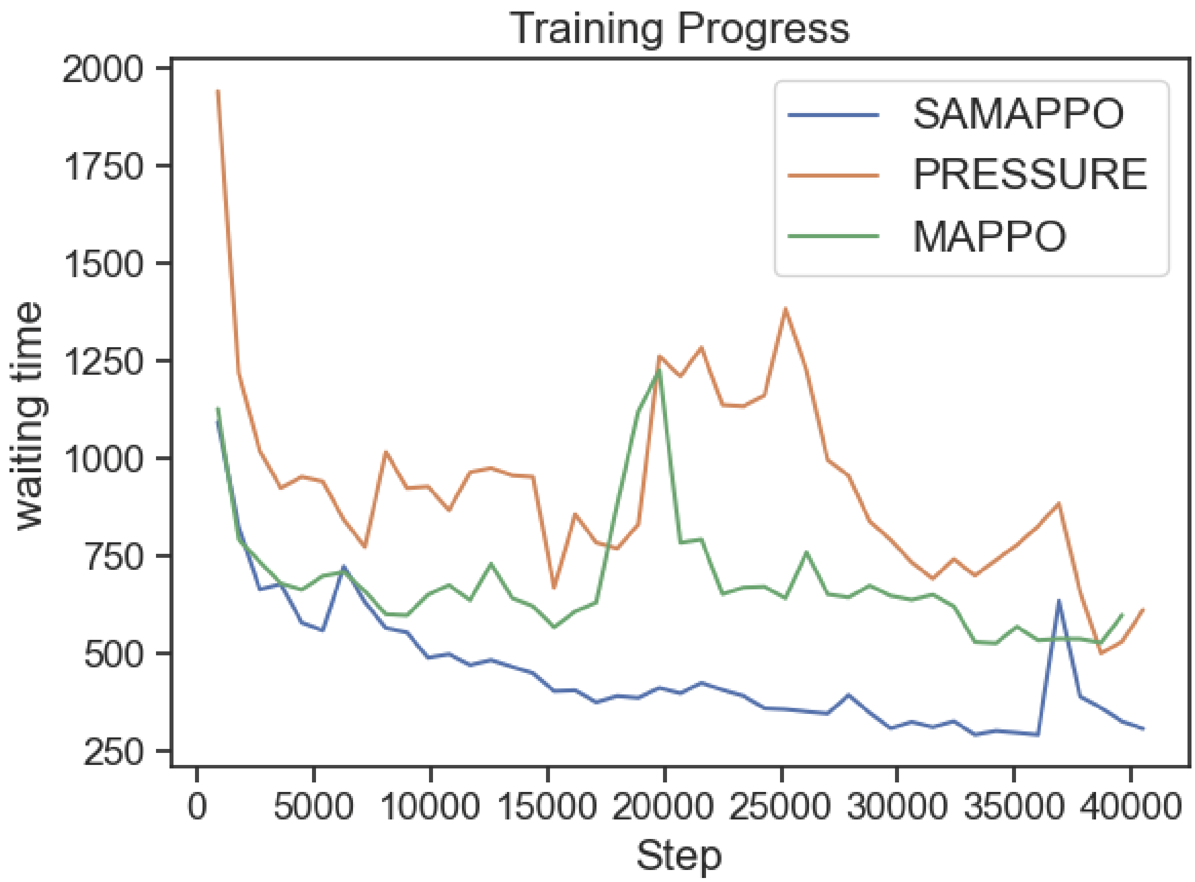

Figure 4.6 training line plot for waiting time in 12 intersections

In our evaluation of training outcomes, it's important to note that the models under assessment employ different reward functions. To gauge their effectiveness, we rely on measuring waiting times throughout the training process, depicted in the training curve plots (Figure 4.5 and Figure 4.6). Our analysis uncovers a consistent trend across all models: a gradual decrease in waiting times as training unfolds. Particularly noteworthy is the SAMAPPO algorithm's superior performance, consistently achieving reduced waiting times and showcasing a more stable training pattern across both the 6-intersection and 12-intersection networks.

An interesting revelation emerges from our observations concerning the waiting time curves in the 12-intersection network. Here, we witness a notably swifter convergence towards the minima compared to the 6-intersection network, requiring less than 8 times the duration. This accelerated convergence is due to our strategic transfer learning approach. By leveraging pre-trained models from the 6-intersection network, our focus lies primarily on only optimizing the encoder and preprocessing layers during training, while keeping the PPO network layers static. This method facilitates efficient feature construction in the 12-intersection scenario, contributing significantly to the expeditious convergence observed.

4.6 Results and discussion

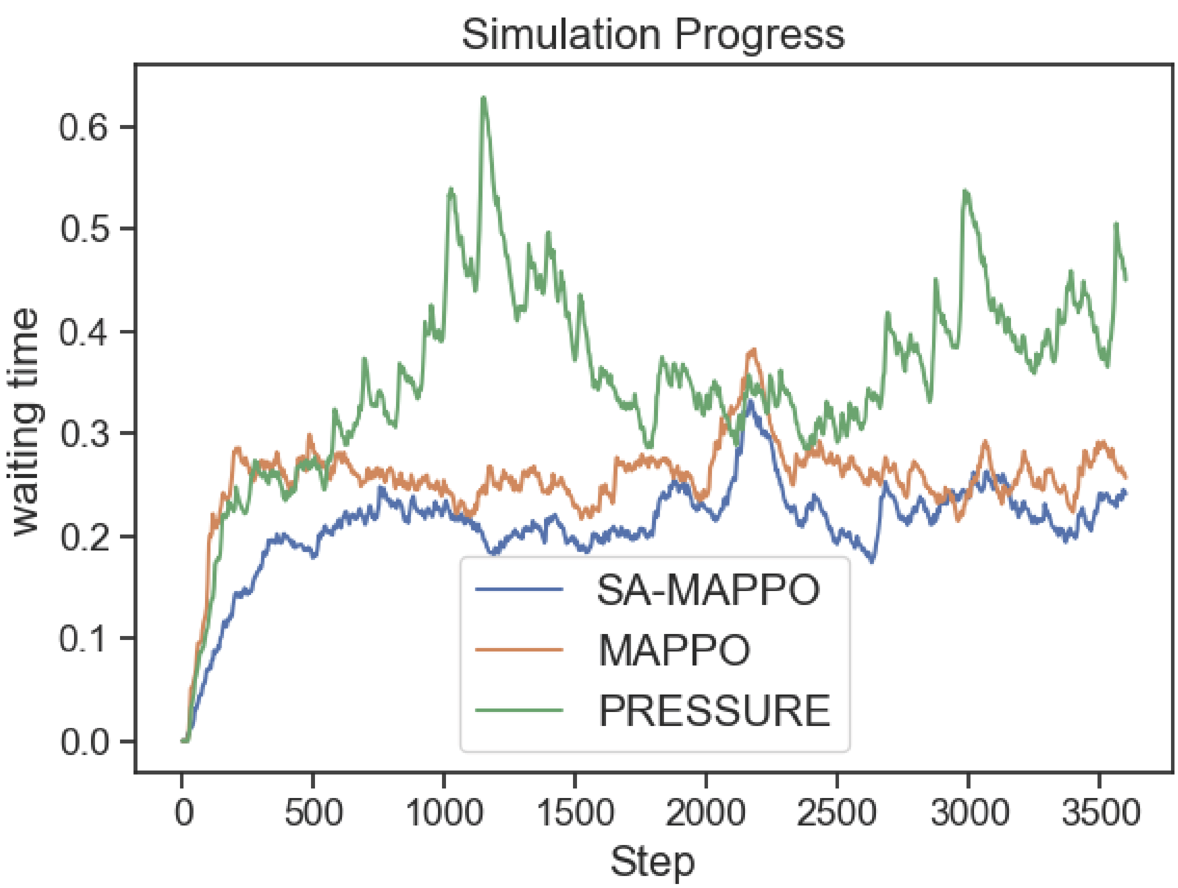

Figure 4.7 waiting time during the simulation in 6 intersections in normal traffic flow

Table 4.3 results on 6 intersections and normal traffic flow

	Acc. waiting	Avg speed	Avg waiting
SAMAPPO	**19743**	**9.18**	**201**
PRESSURE	26466	8.95	327
MAPPO	21920.3	9.11	249
FIXED	181460	6.4	2614

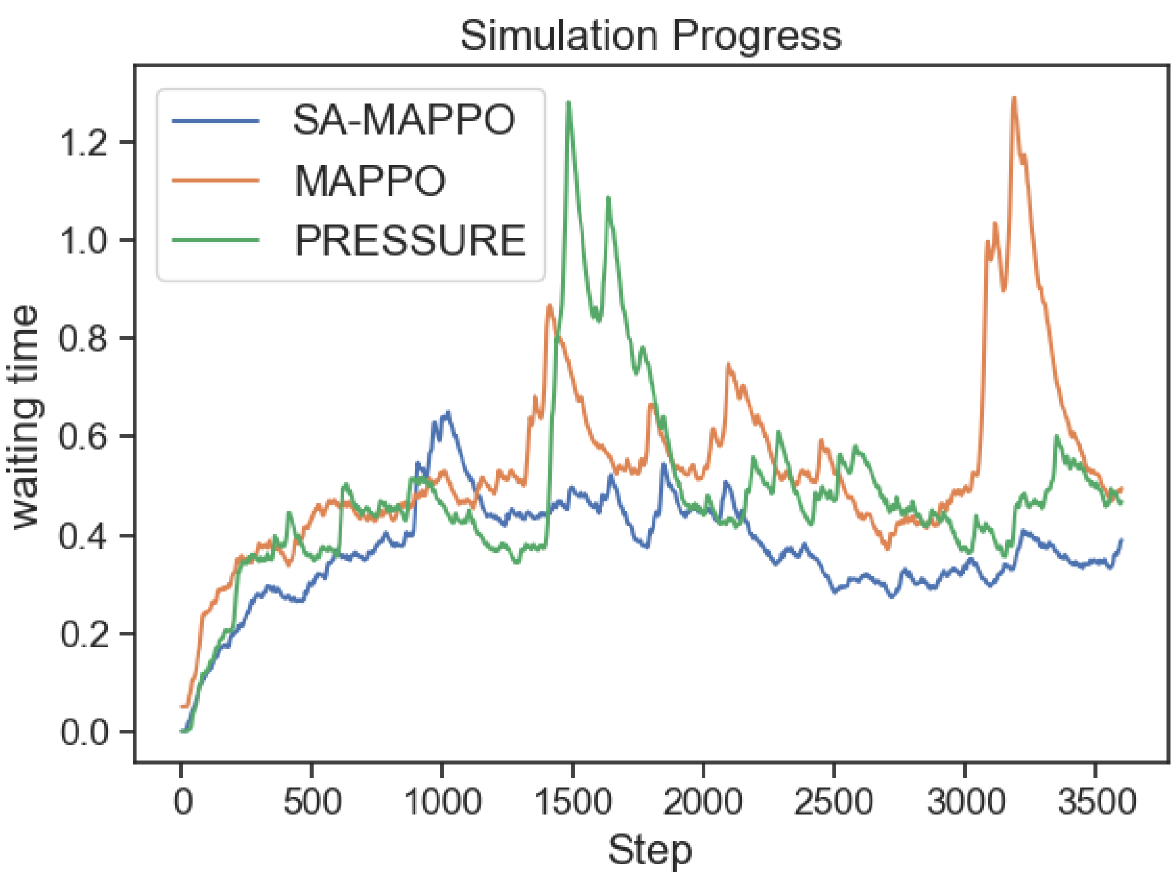

Figure 4.8 waiting time during the simulation in 6 intersections in high traffic flow

Table 4.4 results on 6 intersections and high traffic flow

	Acc. waiting	Avg speed	Avg waiting
SAMAPPO	**42420**	**8.34**	**314**
PRESSURE	50069	8.07	465
MAPPO	49674	8.20	451
FIXED	233840	6.13	4239

In the context of 6-intersection networks (Figure 4.7), our SAMAPPO algorithm consistently outperforms its closest competitor, MAPPO, across benchmark metrics. It achieves an approximate 19% improvement in mean waiting time. Notably, our selected reward function surpasses the Pressure reward function by 38% under typical traffic flow conditions as expressed in Table 4.3

In high traffic flow scenarios, algorithms incorporating self-attention layers exhibit greater resilience compared to non-self-attention-based MAPPO as seen in Table 4.4. SAMAPPO, for instance, experiences a 35% performance decrease in higher flow conditions, whereas MAPPO's performance deteriorates by almost 45%, as evident in Figure 4.8, where SAMAPPO's curve displays greater stability compared to MAPPO's.

This resilience against higher traffic flow was also observed in the PRESSURE algorithm which employs self-attention layers, where the performance dropped by only 30%, reinforcing the efficacy of self-attention mechanisms in managing high traffic flow situations

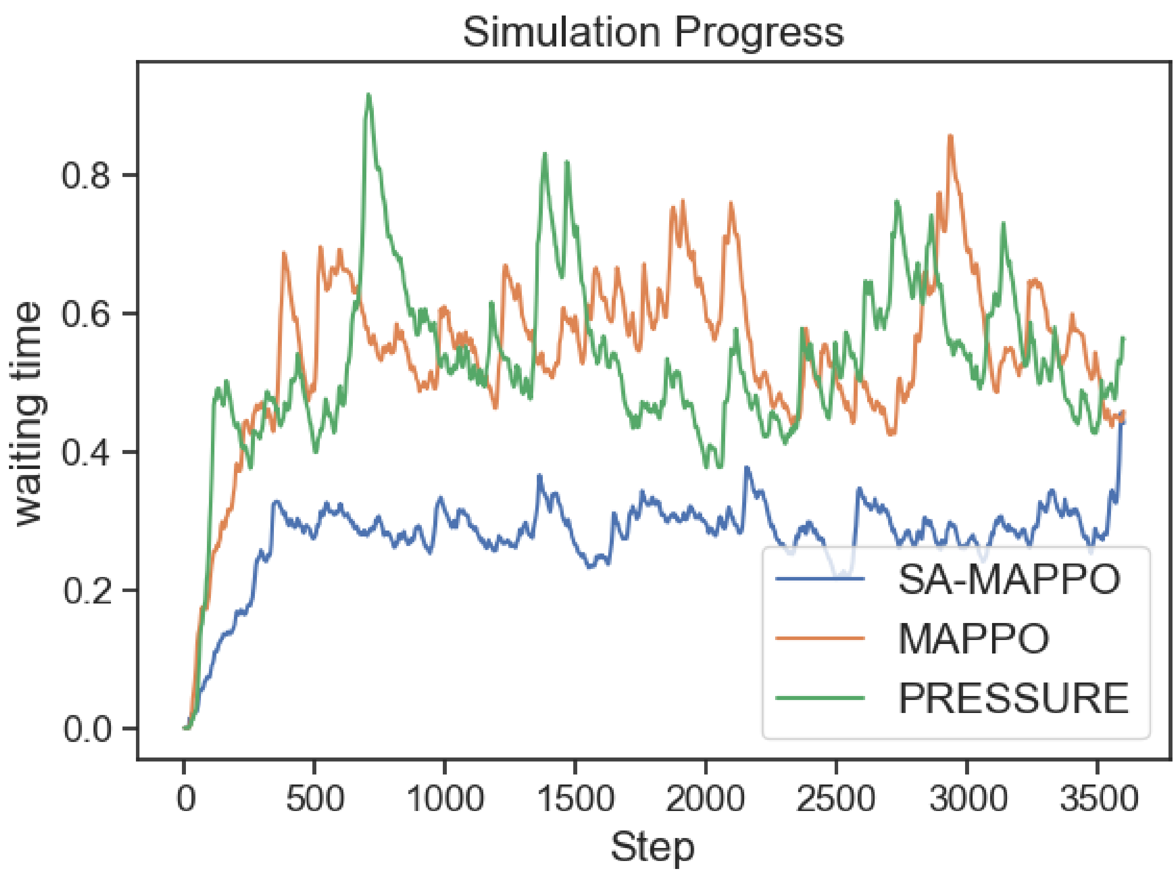

Figure 4.9 waiting time during the simulation in 12 intersections in normal traffic flow

Table 4.5 results on 12 intersections and normal traffic flow

	Acc. waiting	Avg speed	Avg waiting
SAMAPPO	**33411**	**9.39**	**263**
PRESSURE	46707	9.07	493
MAPPO	47836	9.09	505
FIXED	228844	7.59	3590

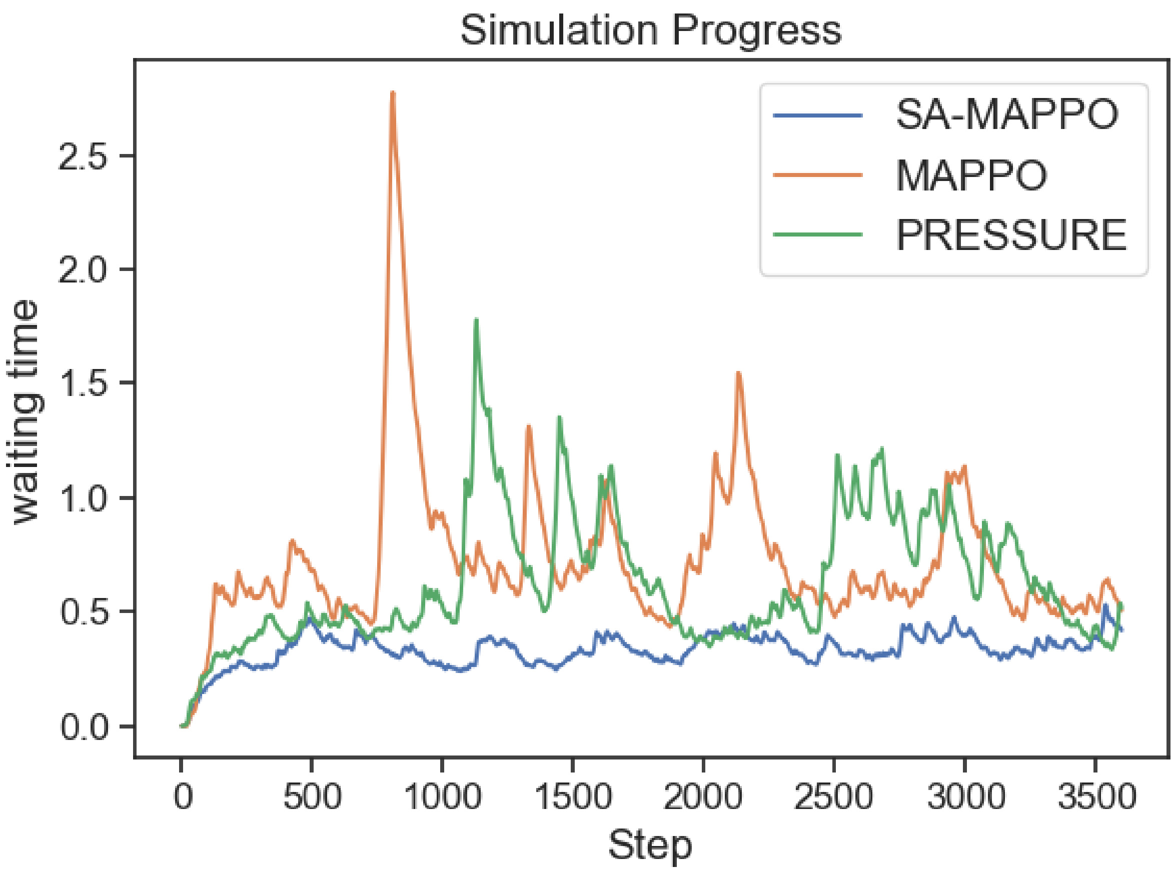

Figure 4.10 waiting time during the simulation in 12 intersections in high traffic flow

Table 4.6 results on 12 intersections and high traffic flow

	Acc. waiting	Avg speed	Avg waiting
SAMAPPO	**59192**	**8.98**	**376**
PRESSURE	70886	8.79	573
MAPPO	80884	8.6	657
FIXED	316929	7.25	4761

In our assessment of the 12-intersection network, our trained algorithms consistently maintain their performance levels, leveraging transfer learning alongside encoding mechanisms (Figure 4.9 and Figure 4.10). They showcase a sustained superiority over the fixed approach, mirroring the margins observed in the 6-intersection network. Notably, from the results presented in Table 4.5 our approach maintains its dominance over rival algorithms, exhibiting a substantial lead of 46% and 47% over Pressure and MAPPO, respectively, in normal traffic flow scenarios. In high traffic flow scenarios, our approach maintains its strong performance, outpacing Pressure and MAPPO by 34% and 42%, respectively, as measured by the average waiting time (Table 4.6).

It's noteworthy that the PRESSURE reward-based algorithm demonstrates improved waiting time performance compared to MAPPO across both traffic flow scenarios in the context of the 12-intersection network, deviating from the 6-network results. This variance can be attributed to PRESSURE's integration of self-attention layers, facilitating the extraction of crucial features from multiple agents. This mechanism proves particularly advantageous in larger networks, allowing PRESSURE to excel in waiting time performance due to its enhanced ability to capture essential multi-agent interactions and dependencies

4.7 Conclusion

In this chapter, we take a close look at our algorithms. We start by explaining the reasoning behind choosing each component of our model as well as a detailed explanation of their logic. Then, we dive into the simulation environment where our algorithm is examined, and we outline the metrics we use to compare it with other approaches. Finally, we provide evidence supporting the benefits of including self-attention. This evidence shows that our approach performs better than non-self-attention algorithms across various network sizes and traffic situations. Plus, our reward design proved more effective when compared to the well-regarded pressure-based rewards.

Chapter 5 General conclusion

Our thesis addresses the pressing challenge of traffic congestion, a direct consequence of rapid urbanization and the substantial increase in vehicle volume aimed at accommodating this urban growth. This traffic congestion problem imposes significant negative economic and environmental effects, prompting an urgent need for effective solutions.

We provided context for the global rise of smart cities, attributing it to substantial technological progress in recent decades. Among the pivotal strategies identified for mitigating congestion in a smart city is the implementation of adaptive traffic light control algorithms. Utilizing wireless communication within VANET networks enables dynamic adjustments to traffic light timing and phases, leveraging real-time data sharing and analysis to make informed decisions aimed at congestion reduction.

We explained that multi-agent adaptive traffic light control can be a discrete or a continuous optimization problem, historically approached using biologically inspired algorithms or ad hoc mathematical and computational approaches. However, the recent success of deep learning and rapid advancements in reinforcement learning algorithms brought about a revolution in how optimization problems are approached, expanding the scope of methodologies applied to such problems. Hence, our introduction of the multi-agent proximal optimization algorithm, SA-MAPPO, equipped with Self-Attention mechanisms to enhance the representation of raw traffic data.

We showcased SA-MAPPO significantly outperforms non self-attention model in a 6-intersection setup. additionally, our reward design surpasses the effectiveness of pressure-based rewards. Notably, our algorithm exhibits scalability in handling higher traffic flows, surpassing both MAPPO and pressure-based SA-MAPPO

To validate scalability in larger networks, we applied our algorithm to a 12-intersection network with a pretrained SA-MAPPO from the 6 intersections network but supplemented by encoder layers—a CNN network to accommodate larger inputs.

Our algorithm maintains superiority in higher flows over MAPPO and pressure-based SAMAPPO, while training time was reduced to about 8x faster than the initial 6-intersection training despite the expanded network size.

Our algorithm's training occurred in a homogeneous network, warranting future exploration in heterogeneous networks with varied topologies and phase cycles. Moreover, optimizing V2V and V2I communications by integrating vehicle routings into the algorithm's actions presents a promising avenue for improvement.

In summary, our algorithm showcases its superiority by alleviating traffic congestion in higher flow scenarios and maintaining performance in larger networks, establishing it as an effective solution for congestion mitigation in smart cities.

REFERENCES

6 Technological Advancements That Make Singapore a Smart City. (n.d.). Retrieved January 21, 2024, from https://www.iotforall.com/singapore-smart-city

Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement Learning for *True* Adaptive Traffic Signal Control. *Journal of Transportation Engineering*, *129*(3), 278–285. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:3(278)

Aisha Bint Butti, B. B., & Okan, G. (2021). *Smart Dubai—Towards becoming the happiest city on Earth*.

Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, *22*(1), 3–21.

An, S., Lee, B.-H., & Shin, D.-R. (2011). A Survey of Intelligent Transportation Systems. *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*, 332–337. https://doi.org/10.1109/CICSyN.2011.76

Andreasson, N., Patriksson, M., & Evgrafov, A. (2020). *An introduction to continuous optimization: Foundations and fundamental algorithms*. Courier Dover Publications.

Arel, I., Liu, C., Urbanik, T., & Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, *4*(2), 128. https://doi.org/10.1049/iet-its.2009.0070

Arena, F., Pau, G., & Severino, A. (2020). A Review on IEEE 802.11p for Intelligent Transportation Systems. *Journal of Sensor and Actuator Networks*, *9*(2), Article 2. https://doi.org/10.3390/jsan9020022

Armas, R., Aguirre, H., Daolio, F., & Tanaka, K. (2017). Evolutionary design optimization of traffic signals applied to Quito city. *PLoS ONE*, *12*(12). https://doi.org/10.1371/journal.pone.0188757

Barth, M., & Boriboonsomsin, K. (2009). *TRAFFIC CONGESTION AND GREENHOUSE GASES*. *35*.

Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., & Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, *214*(1), 481–518.

Ben-Ayed, O., Boyce, D. E., & Blair, C. E. (1988). A general bilevel linear programming formulation of the network design problem. *Transportation Research Part B: Methodological*, *22*(4), 311–318. https://doi.org/10.1016/0191-2615(88)90006-9

Benevolo, C., Dameri, R. P., & D'Auria, B. (2016). Smart mobility in smart city. In *Empowering Organizations* (pp. 13–28). Springer.

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*(2), 239–287.

Blum, C. (n.d.). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, *35*(3), 41.

Capdevila, I., & Zarlenga, M. (2015). Smart city or smart citizens? The Barcelona case. *The Barcelona Case (March 26, 2015)*.

Ceylan, H., & Bell, M. G. H. (2004). Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B: Methodological*, *38*(4), 329–342. https://doi.org/10.1016/S0191-2615(03)00015-8

Cheng, Z. (Aaron), Pang, M.-S., & Pavlou, P. A. (2020). Mitigating Traffic Congestion: The Role of Intelligent Transportation Systems. *Information Systems Research*, *31*(3), 653–674. https://doi.org/10.1287/isre.2019.0894

Chou, C.-H., & Teng, J.-C. (2002). A fuzzy logic controller for traffic junction signals. *Information Sciences*, *143*(1–4), 73–97. https://doi.org/10.1016/S0020-0255(02)00199-8

Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J. R., Mellouli, S., Nahon, K., Pardo, T. A., & Scholl, H. J. (2012). Understanding smart cities: An integrative framework. *2012 45th Hawaii International Conference on System Sciences*, 2289–2297.

De Souza, A. M., Brennand, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., & Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, *13*(4), 155014771668361. https://doi.org/10.1177/1550147716683612

Dhrisya, K., Remya, G., & Mohan, A. (2020). Fine-grained entity type classification using GRU with self-attention. *International Journal of Information Technology*, *12*(3), 869–878. https://doi.org/10.1007/s41870-020-00499-5

Dixit, M., Kumar, R., & Sagar, A. K. (2016). VANET: Architectures, research issues, routing protocols, and its applications. *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 555–561.

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2014). Design of Reinforcement Learning Parameters for Seamless Application of Adaptive Traffic Signal Control. *Journal of Intelligent Transportation Systems*, *18*. https://doi.org/10.1080/15472450.2013.810991

Figueiredo, L., Jesus, I., Machado, J. T., Ferreira, J. R., & De Carvalho, J. M. (2001). Towards the development of intelligent transportation systems. *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, 1206–1211.

Fitah, A., Badri, A., Moughit, M., & Sahel, A. (2018). Performance of DSRC and WIFI for Intelligent Transport Systems in VANET. *Procedia Computer Science*, *127*, 360–368. https://doi.org/10.1016/j.procs.2018.01.133

Foy, M. D., Benekohal, R. (Ray) F., & Goldberg, D. E. (1992). SIGNAL TIMING DETERMINATION USING GENETIC ALGORITHMS. *Transportation Research Record*, *1365*. https://trid.trb.org/view/371411

Fujdiak, R., Masek, P., Mlynek, P., Misurec, J., & Muthanna, A. (2016). Advanced optimization method for improving the urban traffic management. *2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*, 48–53. https://doi.org/10.1109/FRUCT-ISPIT.2016.7561507

García-Nieto, J., Alba, E., & Carolina Olivera, A. (2012). Swarm intelligence for traffic light scheduling: Application to real urban areas. *Engineering Applications of Artificial Intelligence*, *25*(2), 274–283. https://doi.org/10.1016/j.engappai.2011.04.011

Gendreau, M., & Potvin, J.-Y. (2005). Metaheuristics in combinatorial optimization. *Annals of Operations Research*, *140*(1), 189–213.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Guevara, L., & Auat Cheein, F. (2020). The Role of 5G Technologies: Challenges in Smart Cities and Intelligent Transportation Systems. *Sustainability*, *12*(16), Article 16. https://doi.org/10.3390/su12166469

Hajbabaie, A., & Benekohal, R. (2015). A Program for Simultaneous Network Signal Timing Optimization and Traffic Assignment. *IEEE Transactions on Intelligent Transportation Systems*, *16*, 1–14. https://doi.org/10.1109/TITS.2015.2413360

Hollands, R. G. (2008). Will the real smart city please stand up? Intelligent, progressive or entrepreneurial? *City*, *12*(3), 303–320.

Hu, W., Wang, H., Yan, L., & Du, B. (2016). A swarm intelligent method for traffic light scheduling: Application to real urban traffic networks. *Applied Intelligence*, *44*(1), 208–231. https://doi.org/10.1007/s10489-015-0701-y

Hu, X., Zhao, C., & Wang, G. (2020). A Traffic Light Dynamic Control Algorithm with Deep Reinforcement Learning Based on GNN Prediction. *arXiv:2009.14627 [Cs]*. http://arxiv.org/abs/2009.14627

Jee-Hyong Lee & Hyung Lee-Kwang. (1999). Distributed and cooperative fuzzy controllers for traffic intersections group. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *29*(2), 263–271. https://doi.org/10.1109/5326.760570

Khan, M. S., Woo, M., Nam, K., & Chathoth, P. K. (2017). Smart city and smart tourism: A case of Dubai. *Sustainability*, *9*(12), 2279.

Kwatirayo, S., Almhana, J., & Liu, Z. (2013). Adaptive Traffic Light Control using VANET: A case study. *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 752–757. https://doi.org/10.1109/IWCMC.2013.6583651

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, J. H., Hancock, M. G., & Hu, M.-C. (2014). Towards an effective framework for building smart cities: Lessons from Seoul and San Francisco. *Technological Forecasting and Social Change*, *89*, 80–99. https://doi.org/10.1016/j.techfore.2013.08.033

Levy, J. I., Buonocore, J. J., & Von Stackelberg, K. (2010). Evaluation of the public health impacts of traffic congestion: A health risk assessment. *Environmental Health*, *9*(1), 65. https://doi.org/10.1186/1476-069X-9-65

Li, W., Cai, Y., Dinesha, U., Fu, Y., & Di, X. (2021). CVLight: Deep Reinforcement Learning for Adaptive Traffic Signal Control with Connected Vehicles. *arXiv:2104.10340 [Cs, Eess]*. http://arxiv.org/abs/2104.10340

Li, Y., Ying, K., Cheng, P., Yu, H., & Luo, H. (2012). Cooperative data dissemination in cellular-VANET heterogeneous wireless networks. *2012 4th International High Speed Intelligent Communication Forum*, 1–4. https://doi.org/10.1109/HSIC.2012.6212991

Liang, X., Du, X., Wang, G., & Han, Z. (2019). Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, *68*(2), 1243–1253. https://doi.org/10.1109/TVT.2018.2890726

Liu, J., Zhang, H., Fu, Z., & Wang, Y. (2020). Learning Scalable Multi-Agent Coordination by Spatial Differentiation for Traffic Signal Control. *arXiv:2002.11874 [Cs]*. http://arxiv.org/abs/2002.11874

Mannion, P., Duggan, J., & Howley, E. (2016). *An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control* (pp. 47–66). https://doi.org/10.1007/978-3-319-25808-9_4

Maslekar, N., Boussedjra, M., Mouzna, J., & Labiod, H. (2011). VANET Based Adaptive Traffic Signal Control. *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 1–5. https://doi.org/10.1109/VETECS.2011.5956305

Mikami, S., & Kakazu, Y. (1994). Genetic reinforcement learning for cooperative traffic signal control. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 223–228.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Harley, T., Lillicrap, T. P., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 1928–1937.

Mousavi, S. S., Schukat, M., & Howley, E. (2017). Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, *11*(7), 417–423. https://doi.org/10.1049/iet-its.2017.0153

Nam, T., & Pardo, T. A. (2011a). Conceptualizing smart city with dimensions of technology, people, and institutions. *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, 282–291.

Nam, T., & Pardo, T. A. (2011b). Smart city as urban innovation: Focusing on management, policy, and context. *Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance*, 185–194.

Pandit, K., Ghosal, D., Zhang, H. M., & Chuah, C.-N. (2013). Adaptive Traffic Signal Control With Vehicular Ad hoc Networks. *IEEE Transactions on Vehicular Technology*, *62*(4), 1459–1471. https://doi.org/10.1109/TVT.2013.2241460

Panjrath, N., & Poriye, M. (2017). A Comprehensive Survey of VANET Architectures and Design. *International Journal of Advanced Research in Computer Science*, *8*(5), 2099–2103. Applied Science & Technology Full Text (H.W. Wilson).

Parker, R. G., & Rardin, R. L. (2014). *Discrete optimization*. Elsevier.

Patel, N. (n.d.). *90% Of Startups Fail: Here's What You Need To Know About The 10%*. Forbes. Retrieved March 6, 2024, from https://www.forbes.com/sites/neilpatel/2015/01/16/90-of-startups-will-fail-heres-what-you-need-to-know-about-the-10/

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, *1*(1), 33–57.

Putha, R., Quadrifoglio, L., & Zechman, E. (2012). Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions. *Computer-Aided Civil and Infrastructure Engineering*, *27*(1), 14–28.

Putra, Z. D. W., & Van Der Knaap, W. (2019). A smart city needs more than just technology: Amsterdam's Energy Atlas project. In *Smart City Emergence* (pp. 129–147). Elsevier. https://doi.org/10.1016/B978-0-12-816169-2.00006-7

Qadri, S. S. S. M., Gökçe, M. A., & Öner, E. (2020). State-of-art review of traffic signal control methods: Challenges and opportunities. *European Transport Research Review*, *12*(1), 55. https://doi.org/10.1186/s12544-020-00439-1

Qureshi, K. N., & Abdullah, A. H. (2013). *A Survey on Intelligent Transportation Systems*.

Reinforcement learning. (2024). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Reinforcement_learning&oldid=1250213557

Sanchez-Medina, J. J., Galan-Moreno, M. J., & Rubio-Royo, E. (2010). Traffic Signal Optimization in "La Almozara" District in Saragossa Under Congestion Conditions, Using Genetic Algorithms, Traffic Microsimulation, and Cluster Computing. *IEEE Transactions on Intelligent Transportation Systems*, *11*(1), 132–141. https://doi.org/10.1109/TITS.2009.2034383

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust Region Policy Optimization. *Proceedings of the 32nd International Conference on Machine Learning*, 1889–1897. https://proceedings.mlr.press/v37/schulman15.html

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

Shaheen, S., & Finson, R. (2013). *Intelligent Transportation Systems*. https://escholarship.org/uc/item/3hh2t4f9

Shandil, P. (2023). A Survey of Different VANET Routing Protocols. *Evergreen*, *10*(2), 976–997. https://doi.org/10.5109/6793653

Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-Attention with Relative Position Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 464–468. https://doi.org/10.18653/v1/N18-2074

Shijie, W., & Shangbo, W. (2023). A Novel Multi-Agent Deep RL Approach for Traffic Signal Control. *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops)*, 15–20. https://doi.org/10.1109/PerComWorkshops56833.2023.10150326

Smart Dubai 2021 Strategy | The Official Portal of the UAE Government. (n.d.). Retrieved January 21, 2024, from https://u.ae/en/about-the-uae/strategies-initiatives-and-awards/strategies-plans-and-visions/strategies-plans-and-visions-untill-2021/smart-dubai-2021-strategy

Sorensen, K., Sevaux, M., & Glover, F. (2017). A History of Metaheuristics. *arXiv:1704.00853 [Cs]*. http://arxiv.org/abs/1704.00853

Srinivasan, D., Choy, M. C., & Cheu, R. L. (2006). Neural Networks for Real-Time Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, *7*(3), 261–272. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2006.874716

Srivastava, S., & Sahana, S. K. (2017). Nested hybrid evolutionary model for traffic signal optimization. *Applied Intelligence*, *46*(1), 113–123.

Stoilov, T., Stoilova, K., Papageorgiou, M., & Papamichail, I. (2015). Bi-Level Optimization in a Transport Network. *Cybernetics and Information Technologies*, *15*. https://doi.org/10.1515/cait-2015-0015

Stoilova, K., & Stoilov, T. (2020). *Bi-level Optimization Application for Urban Traffic Management*. 327–336. https://doi.org/10.15439/2020F18

Sutton, R. S., & Barto, A. G. (n.d.). *Reinforcement Learning: An Introduction*. 352.

Tanuja, K., Sushma, T. M., Bharathi, M., & Arun, K. H. (2015). *A survey on VANET technologies*.

Teo, K. T. K., Kow, W. Y., & Chin, Y. K. (2010). Optimization of Traffic Flow within an Urban Traffic Light Intersection with Genetic Algorithm. *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, 172–177. https://doi.org/10.1109/CIMSiM.2010.95

Tomar, R., Prateek, M., & Sastry, G. H. (n.d.). *Vehicular Adhoc Network (VANET)—An Introduction*. 7.

Van der Pol, E., & Oliehoek, F. A. (2016). Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*. https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Vegni, A. M., Biagi, M., & Cusani, R. (2013). Smart Vehicles, Technologies and Main Applications in Vehicular Ad hoc Networks. *Vehicular Technologies - Deployment and Applications*. https://doi.org/10.5772/55492

Webster, F. V. (1958). TRAFFIC SIGNAL SETTINGS. *Road Research Lab Tech Papers /UK/*, Article No 39. https://trid.trb.org/view/113579

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

Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496–2505. https://doi.org/10.1145/3219819.3220096

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85.

Woyke, E. (2019, January 9). *Smart cities could be lousy to live in if you have a disability | MIT Technology Review*. https://www.technologyreview.com/2019/01/09/137899/smart-cities-coule-be-lousy-if-you-have-a-disability/

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., & Wu, Y. (2021). The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv:2103.01955 [Cs]*. http://arxiv.org/abs/2103.01955

Zeadally, S., Hunt, R., Chen, Y.-S., Irwin, A., & Hassan, A. (2012). Vehicular ad hoc networks (VANETS): Status, results, and challenges. *Telecommunication Systems*, *50*(4), 217–241.

Zhen-Jin Huang, Chun-Gui Li, & Zeng-Fang Zhang. (2010). Traffic signal control based on genetic neural network algorithm. *2010 International Conference on Intelligent Computing and Integrated Systems*, 31–34. https://doi.org/10.1109/ICISS.2010.5655002

Abstract

As urban landscapes evolve rapidly, marked by exponential growth in both population and vehicles, this thesis addresses the pressing issue of traffic congestion through the implementation of an innovative multi agent adaptive control algorithm for traffic lights In a smart city. Leveraging VANET technology for real-time communication and data-sharing across multiple signalized intersections and vehicles, our algorithm, named Self-Attention Multi-Agents Proximal Policy Optimization (SAMAPPO), aims to alleviate congestion in intersections with diverse traffic flows and traffic network map sizes.

Utilizing the simulation tool SUMO for realistic traffic simulation, our algorithm demonstrates its efficacy in varying traffic flows as well as small and large network maps. An additional strength of our algorithm lies in its scalability, showcasing superior performance in larger networks without requiring retraining the whole model, thanks to the incorporation of transfer learning, which reduces the computation costs associated with training. Our implementation proves to be a practical solution for congestion in smart cities, offering scalability to accommodate higher traffic flows and larger network maps. The success of our algorithm suggests its potential to address the traffic congestion challenge posed by evolving urban traffic scenarios.

Key words: smart city, vehicular ad hoc network (VANET), Adaptive traffic lights control, reinforcement learning, multi-agents, Self-attention, Proximal Policy Optimization

Résumé

Alors que les paysages urbains évoluent rapidement, marqués par une croissance exponentielle de la population et des véhicules, cette thèse aborde la question de la congestion routière à travers la mise en œuvre d'un algorithme innovant de contrôle adaptatif multi-agents des feux de circulation dans une ville intelligente. Utilisant la technologie VANET pour la communication en temps réel et le partage de données entre plusieurs intersections et véhicules signalés, notre algorithme, nommé Auto-Attention Multi-agents Optimisation de la Politique Proximale (AAMOPP), vise à réduire la congestion aux intersections avec différents flux de circulation et tailles de réseau routier.

En utilisant l'outil de simulation SUMO pour une simulation réaliste du trafic, notre algorithme démontre son efficacité dans différents flux de trafic ainsi que dans les petites et grandes cartes de réseau. Un avantage supplémentaire de notre algorithme réside dans son évolutivité, montrant des performances supérieures dans des réseaux plus grands sans nécessiter de recyclage de l'ensemble du modèle, grâce à l'incorporation de l'apprentissage par transfert, qui réduit les coûts de calcul associés à la formation. Notre mise en œuvre s'avère être une solution pratique pour lutter contre la congestion dans les villes intelligentes, offrant une évolutivité permettant de s'adapter à des flux de trafic plus élevés et à des cartes de réseau plus vastes. Le succès de notre algorithme suggère son potentiel pour relever le défi de la congestion routière posé par l'évolution des scénarios de trafic urbain.

MOTS CLÉS : Ville Intelligente, réseau Ad-Hoc *de* véhicules, Contrôle Adaptatif des Feux de Circulation, Apprentissage par Renforcement, Multi-agents, Auto-Attention,

ملخص

مع التطور السريع في المشهد الحضري وزيادة عدد السكان والسيارات بشكل كبير، تتناول هذه الأطروحة قضية ازدحام المرور من خلال تطبيق خوارزمية مبتكرة للتحكم التكيفي في إشارات المرور باستخدام الوكلاء المتعددين في المدن الذكية. بالاعتماد على تقنية الشبكات اللاسلكية للمركبات (VANET) للإتصال الفوري وتبادل البيانات بين التقاطعات المرورية المضيئة والمركبات، تهدف خوارزميتنا المسماة تحسين السياسة القريبة للوكلاء المتعددين ذات الانتباه الذاتي (SAMAPPO) إلى تخفيف الازدحام في التقاطعات المرورية ذات تدفقات المرور المتغيرة وأحجام خرائط الشبكات المرورية المختلفة.

باستخدام أداة المحاكاة SUMO لمحاكاة حركة المرور الواقعية، تُظهر خوارزميتنا فعاليتها في تدفقات مرور متنوعة بالإضافة إلى خرائط شبكات صغيرة وكبيرة. وتتميز خوارزميتنا بقابليتها للتوسع، حيث تقدم أداءً متفوقًا في الشبكات الكبيرة دون الحاجة إلى إعادة تدريب النموذج بالكامل مرة اخرى، وذلك بفضل دمج تقنية نقل التعلّم، مما يقلل من تكاليف الحوسبة المرتبطة بالتدريب. يثبت تطبيقنا أنه حل عملي لمشكلة الازدحام في المدن الذكية، حيث يوفر قابلية للتوسع لاستيعاب تدفقات مرور أعلى وخرائط شبكية أكبر. نجاح خوارزميتنا يشير إلى إمكانيتها في معالجة التحديات المرتبطة بازدحام المرور في ظل السيناريوهات المرورية الحضرية المتطورة.

الكلمات المفتاحية: المدينة الذكية، الشبكات اللاسلكية للمركبات (VANET) ، التحكم التكيفي في إشارات المرور، التعلم المعزز، الوكلاء المتعددين، الانتباه الذاتي، تحسين السياسة القريبة.