Title
Agent-Based Urban Traffic Management for Connected and Automated Vehicles

Author
Liu, Shangrui

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Agent-Based Urban Traffic Management for Connected and Automated Vehicles

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Electrical Engineering

by

Shangrui Liu

December 2021

Thesis Committee:
Dr. Matt Barth, Chairperson
Dr. Guoyuan Wu
Dr. Wei Ren

The Thesis of Shangrui Liu is approved:

Committee Chairperson

University of California, Riverside

ABSTRACT OF THE THESIS

Agent-Based Urban Traffic Management for Connected and Automated Vehicles

by

Shangrui Liu

Master of Science, Graduate Program in Electrical Engineering
University of California, Riverside, December 2021
Dr. Matt Barth, Chairperson

Traffic congestion has always been a serious problem in metropolitan areas. Fortunately, the emergence of autonomous vehicles (AVs), vehicle-to-everything communication (V2X), and advanced machine learning algorithms have unlocked uncountable opportunities to improve transportation system management and vehicle operations in terms of safety, mobility, efficiency, and environmental sustainability. In this dissertation, agent-based traffic management strategies have been developed to mitigate traffic congestion for three representative scenarios, including: 1) signalized intersection control using deep reinforcement learning (DRL); 2) signal-free intersection management based on first come first served policy; and 3) reservation-based network traffic management with a multi-agent system (MAS) approach.

The first part focused on the signal timing problem at an isolated intersection by using deep reinforcement learning approach. The system is designed and tested in a realistic

transportation simulation platform (Simulation of Urban, n Mobility) and the intersection configuration is also developed from a real-world scenario (University Ave @ Chicago Ave, Riverside, California). The method significantly achieved improvement in terms of both congestion mitigation and energy saving when compared with traditional signalized intersection management methods. As for the second part, an autonomous centralized intersection management strategy was developed and realized in multiple Raspberry Pi based vehicle models to determine the optimal passing sequence at signal-free intersections in an artificial urban network. The last part moved from small scale scenarios (intersection) to large scale scenarios (network). A reservation–based network traffic management method using MAS was proposed to route individual CAV traversing a given network in terms of minimizing its arrival time. The results show that our system can reduce travel time in the range of 8 – 12%, compared with the state-of-the-practice strategy.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Overview

This thesis focuses on transportation systems, applying the latest advances in Intelligent Transportation Systems (ITS) to mitigate the environmental and energy issues associated with the movement of goods and people [1]. To be more specific, this thesis focuses more on the traffic management problem under the fully Connected and Automated Vehicles (CAVs) environment to improve transportation system management and vehicle operations in terms of safety, mobility, efficiency, and environmental sustainability.

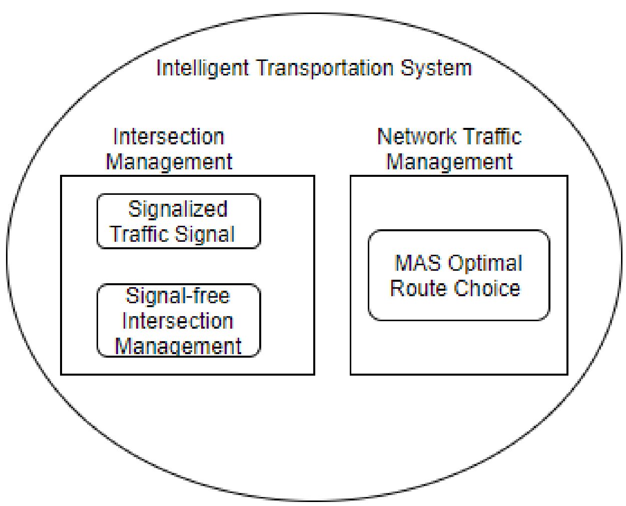


Figure 1. 1 Overview of the study area for this thesis

The traffic management problems are studied in microscopic and macroscopic scenarios, as shown in Figure 1.1. As for the microscopic scenarios, this thesis focuses on managing the traffic at isolated intersections which include signalized intersection and signal-free intersection; As for the macroscopic scenario, the work is focused on routing an individual CAV to traverse a given network in terms of minimizing its travel time. Overall, the research in this thesis belongs to the advanced transportation area and gradually moves from a small scale (an isolated intersection) to a large scale (a grid network).

1.2 Motivation

1.2.1 Traffic Congestion

Rising traffic congestion is an unavoidable condition in large and growing metropolitan areas across the world. Valuable time is wasted by vehicles stuck in traffic. A 2017 report from Texas A&M Transportation Institute [2] shows that the annual travel delay cost by congestion for each commuter in Los Angeles is 119 hours, and 47.9% of the travel delay (caused by congestion) occurred during peak hours in downtown area. Moreover, traffic congestion increases vehicle emissions and degrades ambient air quality. Some previous studies [3] developed scenarios to examine associations between traffic volume, exposures, and health risk, showing that traffic jams may significantly increase the level of environmental pollution. Figure 1.2(A, B) shows concentrations predicted for various emission estimates, traffic volume, and rush hour periods in the freeway scenario. Figure 1.2(C, D) shows predicted concentrations in the arterial scenario. Concentration levels increased nearly linearly to about 3000 vph (vehicle per hour) and then increased sharply.

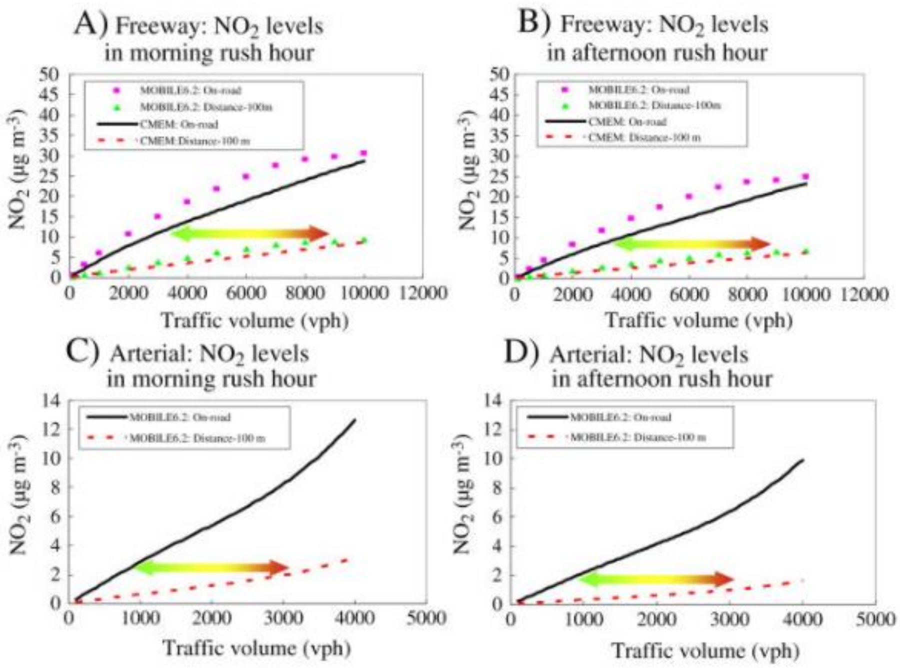


Figure 1. 2 Predicted speed and NO_2 concentrations versus traffic volume in the freeway and arterial scenarios [3]

1.2.2 Autonomous Vehicles (AVs)

Fortunately, the emergence of Autonomous Vehicles (AVs) and Vehicle-to-everything (V2X) communications [54] have unlocked uncountable opportunities to improve transportation system management and vehicle operations in terms of safety, mobility, efficiency, and environmental sustainability.

AVs can detect surroundings by using information from radar, laser light, Global Positioning System (GPS), odometry, and computer vision. Thus, AVs can detect the road segment and everything around them and operate in isolation from other vehicles. Figure 1.3 shows a brief explanation of the different levels of driving automation [4]. An active and engaged driver is required if a vehicle has a level 0, level 1, or level 2 driver support system. Drivers are always responsible for the vehicle's operation, always supervise the technology, and take complete control of the vehicle when necessary. To the current position, various car manufacturers could produce vehicles with automation level 2 or level 2.5. In the coming future, if a vehicle has level 3, level 4, or level 5 automated driving system, the technology takes complete control of the driving without human supervision. However, with level 3, if the vehicle alerts the driver, and requests driver takes control of the vehicle, he or she must be prepared and able to do so.

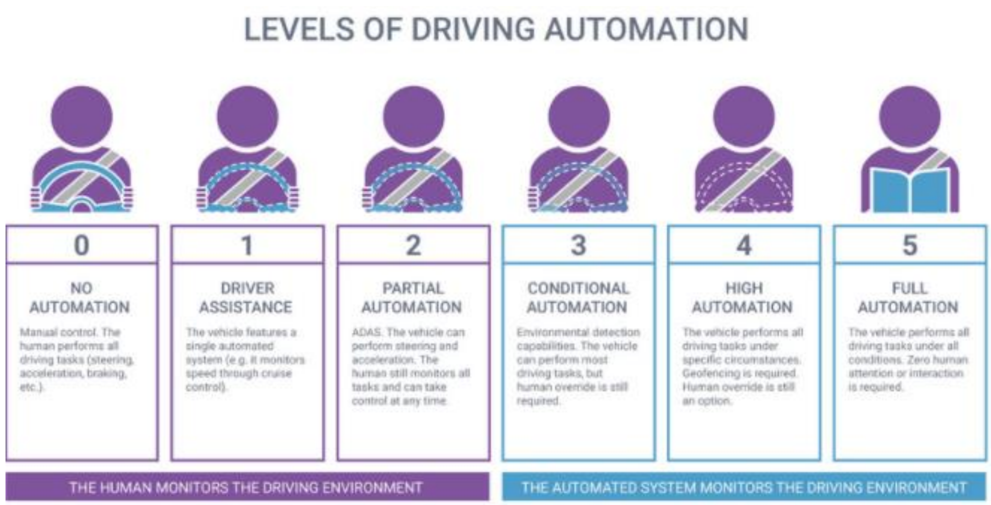


Figure 1. 3 Levels of Driving Automation [56]

1.2.3 Vehicle-to-everything Communications (V2X)

By having richer traffic data, it is critical to transmit them to other parts of the traffic system to manage the traffic better. Vehicle-to-everything (V2X) is a vehicular communication system that supports transferring information from a vehicle to any parts of the traffic system that may affect the vehicle. The primary purpose of V2X technology is to improve road safety, energy savings, and traffic efficiency on the roads. The critical components of V2X technology include vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), which are shown in Figure 1.4 [5]. V2V allows vehicles to communicate with other vehicles on the road, while V2I enables vehicles to share vehicle dynamics with external entities, such as traffic lights, parking spaces, cyclists, and pedestrians. The external entities can integrate all those received information into the server, which helps improve road safety, increase road mobility, reduce fuel consumption, and enhance the experience between drivers and other road users. For example, the roadside infrastructure can estimate the time-dependent traffic volume by receiving the vehicle's information on the road segment, which is critical when solving the traffic management problem. More than sending data to the infrastructure, AVs equipped with V2X systems may also receive the information or guidance from the infrastructure. Although the V2X technique is still in its early stages, most manufacturers have started incorporating the technology, and vehicles are increasingly becoming connected to other vehicles and infrastructure. As a result, the users benefit from safer, more efficient journeys with reduced carbon emissions.

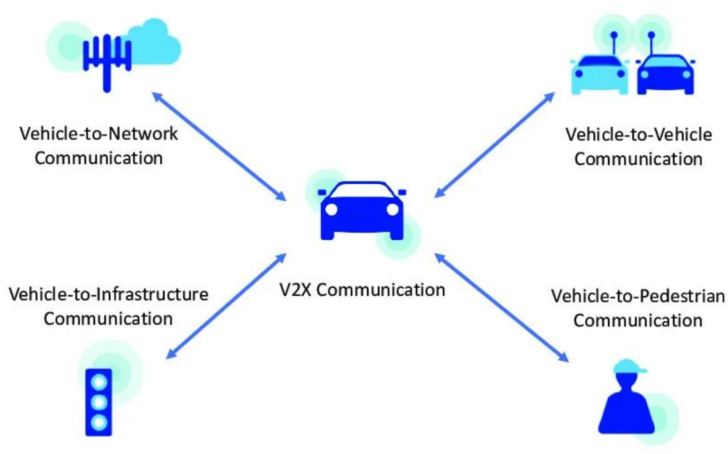


Figure 1. 4 Framework of Vehicle-to-everything (V2X) communication

In general, with the emergence of AVs and V2X communication, richer real-time information is becoming available, and higher definition control (at the individual vehicle level) can be applied to mitigating traffic congestion. Studies in this thesis take advantage of the new technology in intelligent transportation systems and try to solve the intersection management and traffic network management problems from a new perspective.

1.2.4 Artificial Intelligence

Machine learning (ML) has great applicability in the transport industry. Transportation systems have been influenced by the growth of machine learning, particularly in intelligent transportation systems (ITS). Through deep learning, ML explores the complex interactions among road uses, infrastructure, and other environment elements. ML also has great potential in daily traffic management and the collection of traffic data. Figure 1.5 illustrates the growth of applying machine learning approaches to solve a variety of problems in ITS [6].

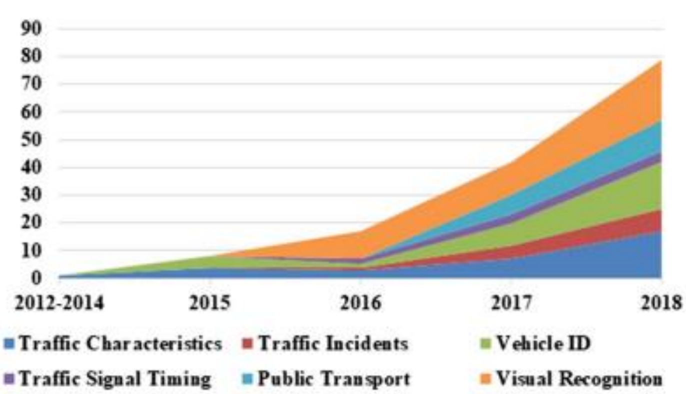


Figure 1. 5 Year-wise publication growth in ITS domains using ML [6]

Reinforcement learning (RL), a branch of machine learning, is applied in this thesis to solve the signalized intersection management problem. The reason for using RL is these techniques can directly learn from the observed data without making any strong assumptions. Different from other types of ML techniques, RL does not require any sampled data. Instead, the RL must first take action to change the signal plans and then learn from the outcomes. This trial-and-error approach is also the core idea of RL. The detailed information on RL will be discussed in the following part of this thesis.

1.3 Work Summary

This section briefly introduces all the works in the thesis and summarizes major contributions of my study. First, the study area belongs to the Intelligent Transportation System (ITS), which applies the latest advances to the mitigation of the environmental and energy issues associated with the movement of goods and people. To be more specific, this thesis focuses more on the traffic management problem under the fully Connected and Automated Vehicle (CAVs) environment to improve system performance in terms of safety, mobility, efficiency, and environmental sustainability.

There are in total three main topics in the thesis, including:

1. Signalized intersection control using deep reinforcement learning (DRL).
2. Signal-free intersection management based on first come first served (FCFS) policy.
3. Reservation-based network traffic management with a multi-agent system (MAS) approach.

In general, agent-based traffic management strategies are developed to mitigate traffic congestion for these three representative scenarios. The studies are moved from a small scale (an isolated intersection) to a large scale (a grid network), from contemporary to futuristic.

The first part focuses on the signal timing problem at an isolated intersection using a deep reinforcement learning approach. The system is designed and tested in a realistic transportation simulation platform (Simulation of Urban MObility). The intersection configuration is also developed from a real-world scenario (University Ave @ Chicago Ave, Riverside, California). The method significantly improved from the perspective of both congestion reduction and energy saving compared with traditional signalized intersection management methods. The main contributions are:

- Develop a flexible traffic light state dual-ring controller with 64 traffic states. This controller provides extra flexibility in the traffic light transitions process, which gives the DRL agent more freedom to learn the optimal policy from the environment.
- Develop a DRL model for the TLC at a single intersection. In the model, traffic lights are controlled by an independent agent. In addition, a knowledge-assisted decision-maker algorithm is embedded in the DRL model to prevent frequent traffic light transitions, and a Deep Q-Network (DQN) algorithm is utilized to help the agent find the optimal policy.
- Validate and compare the proposed algorithm with classical traffic signal control methods (fixed time and actuated) under scenarios of varying traffic demands. The

results of simulation experiments confirm the superior performance of the proposed algorithm over other state-of-the-art methods in terms of average travel delay, emission (CO, CO_2, NO_X), and fuel consumption.

As for the second part, an autonomous (signal-free) intersection management module is installed in four Raspberry Pi-based vehicle models to solve the passing sequence at signal-free intersections in an artificial urban network. The major contributions are:

- Propose a centralized agent-based autonomous intersection management strategy by following the first come first served (FCFS) policy.
- Test the strategy in both simulation environment and real-world environment.

The last part moved from a small scale scenario (an isolated intersection) to a large scale scenario (a grid network). A reservation-based network traffic management method using MAS is proposed to route individual CAV traversing a given network to minimize its arrival time. The results show that proposed strategies can reduce travel delay in the range of 8 – 12%, compared with the state-of-the-practice strategy. The major contributions are:

- Develop a multi-agent system (MAS) based framework for network management of fully CAVs environment, where the network management agent (NMA), link agents (LAs) and vehicle agents (VAs) would cooperate with each other, and their functionalities are clearly defined in a scalable manner.
- Propose a link-level reservation-based routing algorithm for each CAV, considering

its origin/destination and entry time. The first-come-first-served (FCFS) protocol is applied, when reserving the link occupancy.

- Evaluate the mobility and network usage rate performance of different MAS based network traffic management strategies.

1.4. Structure of Thesis

The thesis is structured as follows; Chapter 1 introduces the whole thesis, including the scope, motivation, a summary of the thesis, and major contributions, followed by a detailed literature review and essential background information for some previous related works in Chapter 2. Then Chapter 3 to Chapter 5 discuss three major topics studied in the thesis, which are signalized intersection control using deep reinforcement learning, signal-free intersection management based on first come first served policy, and reservation-based network traffic management with a multi-agent system (MAS) approach. Each chapter introduces the motivation, methodology and experiment results regarding the different topic. Chapter 6 concludes the thesis with further discussion and possible future work. The detailed structure of the thesis is shown in Figure 1.6.

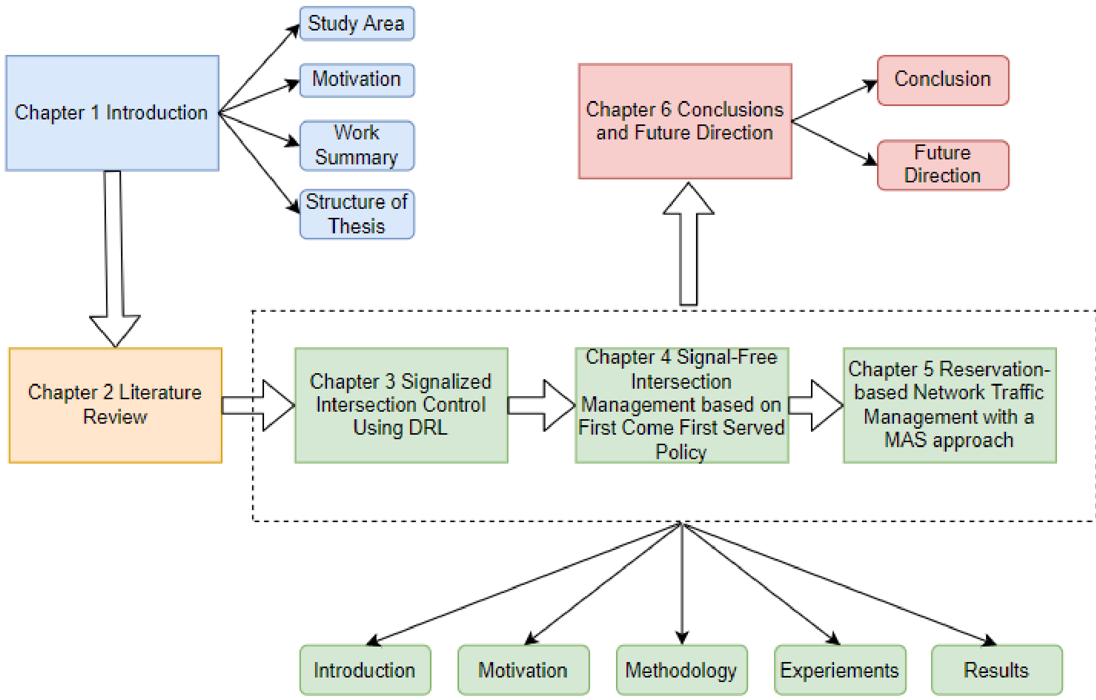


Figure 1. 6 Structure of thesis

Chapter 2 Literature Review

In the past few decades, many efforts have been made on intersection management and network traffic management problems. This chapter reviews essential background information and related previous works.

2.1 Deep Reinforcement Learning

In this thesis, reinforcement learning (RL) and deep learning (DL) work together to solve the signalized traffic signal control problem. Those two techniques are both belonging to the machine learning area. In recent years, ML techniques have become a part of smart transportation. Transportation systems have been influenced by machine learning growth, particularly in intelligent transportation systems (ITS).

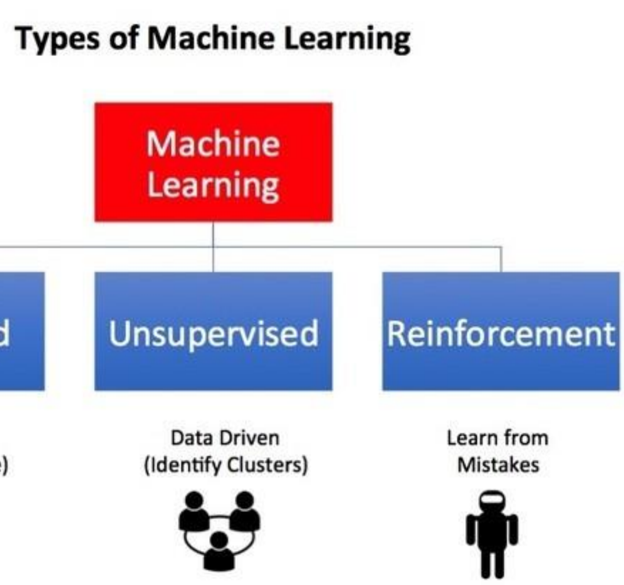


Figure 2. 1 Three types of machine learning

These are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning, which are shown in Figure 2.1. In terms of reinforcement learning, it is a machine learning training method based on rewarding desired behaviors and/or punishing undesired behaviors. In general, a reinforcement learning agent can perceive and interpret its environment, take actions, and learn through trial and error. It is rapidly growing and producing a huge variety of learning algorithms that can be used for various applications.

The reason for using reinforcement learning to solve the signalize intersection management problem is that it does not require any sampled data for the training process. And the labeled training data are hard to obtain in the transportation area since the traffic environment is highly dynamic and none-stationary. Differs from supervised learning, RL is directly learning from the interaction with environment, which is shown in Figure 2.2. An agent interacts with the environment to generate state, action, and reward signals, and its policy is then adjusted from another round of interaction. The final goal of RL is to find the optimal policy in terms of maximizing the long-term reward.

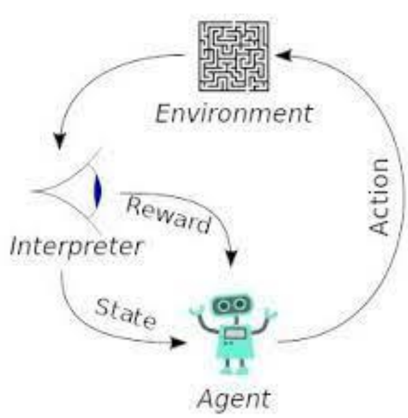


Figure 2. 2 Brief framework for reinforcement learning

The formulation of RL is first introduced, usually a single-agent RL problem is modeled as a **Markov Decision Process (MDP)** equation $< S, A, P, R, \gamma >$, where S, A, P, R, γ are the set of state representations, the set of action, the probabilistic state transition function, the reward function, and the discount factor respectively. The definitions are given as follows:

- $\boldsymbol{S}$: At time step t, the agent observes state $s^{\,t} \in S$.
- $\boldsymbol{A}, \boldsymbol{P}$: At time step t, the agent takes an action at $a^t \in A$, which induces a transition in the environment according to the state transition function:

$$P(s^{t+1}|s^t, a^t): S \times A \rightarrow S \tag{2.1}$$

- $\boldsymbol{R}$: At time step t, the agent obtains a reward $r^t$by a reward function.

$$R(s^t, a^t): S \times A \rightarrow R \tag{2.2}$$

- $\boldsymbol{\gamma}$: The goal of an agent is to find a policy that maximizes the expected return, which is the discounted sum of rewards:

$$G^t \coloneqq \textstyle\sum_{i=0}^{\infty} \gamma^i \times r^{t+i} \tag{2.3}$$

  where the discount factor $\gamma \in [0.1]$ controls the importance of immediate rewards versus future rewards.

Besides RL, deep learning (DL) is also applied to solve the signalized traffic signal problem in this thesis. Like RL, DL is also a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain, allowing it to learn from large amounts of data.

While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. The popular deep learning algorithms are Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs), Recurrent Neural Networks (RNNs), Generative Adversarial Network (GANs), and so on. They have different advantages and are used in various research areas. This thesis uses CNNs, which were designed to map image data to an output variable. Figure 2.3 shows the architecture of a general CNN. Three layers make up the CNN: the convolutional layers, pooling layers, and fully connected layers. When these layers are stacked, a CNN architecture is formed.

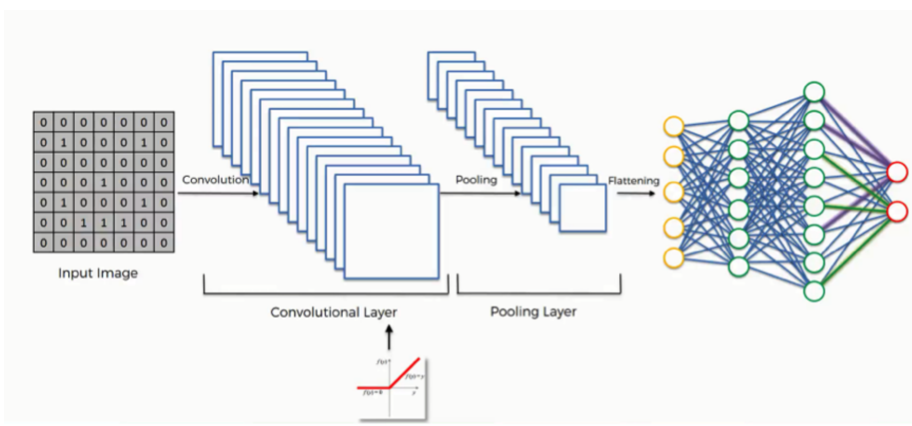


Figure 2. 3 Architecture of Convolution Neural Network (CNN)

As for the convolutional layer, it is the first layer which is used to extract the various features from the input images. In this layer, mathematical operation of convolution is performed between the input image and a filter of a particular size. In most cases, a convolutional layer is followed by a pooling layer. The primary aim of the pooling layer is to decrease the size of the convolved feature map to reduce computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. The pooling layer usually serves as a bridge between the convolutional layer and the fully connected layer. As for the fully connected layer, it consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. The fully connected layer is usually placed before the output layer and forms the last few layers of a CNN Architecture. The input images from the previous layers are flattened and fed to the fully connected layer. The flattened vector then undergoes few more layers where the operations of the mathematical function usually take place. In this stage, the classification process begins to take places.

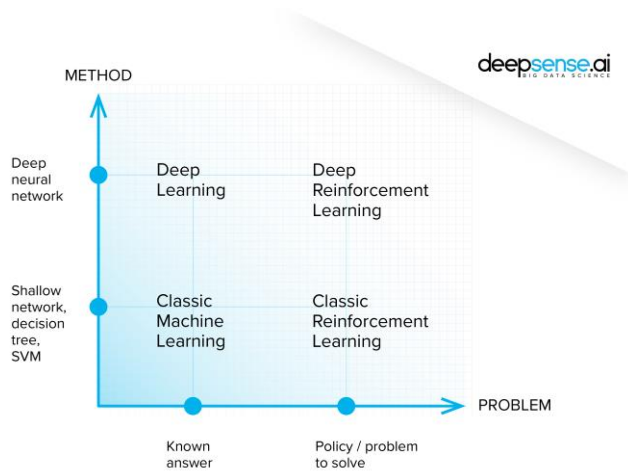


Figure 2. 4 Generation of Deep Reinforcement Learning (DRL) [55]

In addition to these three layers, there are two more important components: the dropout layer and the activation function. These two components are not discussed in detail in this thesis. In general, the integration of RL and DL is Deep Reinforcement Learning (DRL). DRL incorporates deep learning into the solution, allowing agents to make decisions from unstructured input data without manual engineering of the state space. DRL algorithms can take in very large inputs (features of traffic data) and decide what actions to perform to optimize an objective. In recent years, some studies in transportation have already used DRL as a tool, especially in signalized traffic signal problems.

2.2 Traffic Signal Control

Traffic signal control at road intersections allows vehicle movement to be controlled by allocating time intervals, during which separate traffic demands for each approach of the intersection can make use of the available road space. Intersections are bottlenecks of traffic flow and a major cause of traffic accident. According to National Highway Traffic Safety Administration (NHTSA) [7], about 40 percent of the estimated 5,811,000 crashes that occurred in the United States in 2018 were intersection-related crashes. Moreover, major traffic congestion in urban areas is caused by inefficient management methods at intersections.

2.2.1 Background Information

Here are some basic notations and definitions for traffic signal control problems:

- **Approach & lane**: The approach is the roadway meeting at an intersection. The approaches can be defined as incoming approaches (vehicles enter the intersection) and outgoing approaches (vehicles leave the intersection). Figure 2.5 (a) shows a typical intersection with four incoming approaches and outgoing approaches. For example, the westbound incoming approach is denoted in this figure as the vehicles which are traveling in the westbound direction.

Similar to the definition of approach, there are two types of lanes: incoming lanes and outgoing lanes. Each approach consists of a set of lanes. Figure 2.5(a) shows there are three lanes from each coming approach, which include one dedicated right-turn lane, one dedicated through lane and one dedicated left-turn lane. However, in some other intersections, right-turn/through and left-turn/through share the same lane.

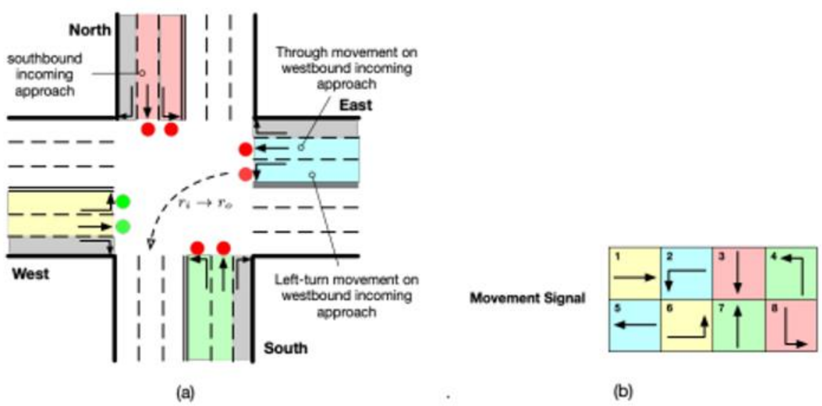


Figure 2. 5 Definitions of traffic movements and traffic signal phase [50]

- **Traffic movement and movement signal:** A traffic movement represents vehicles moving from an incoming approach to an outgoing approach. In general, there are three types of traffic movements from each approach, which include right turn, though, and left turn. Figure 2.5(a) gives an example of traffic movement: $(r_i \rightarrow r_o)$, where r_i is an incoming lane and r_o is an outgoing lane.

A movement signal is defined on the traffic movement, with a green signal indicating the corresponding movement is allowed and a red signal indicating the movement is prohibited. For the four-leg intersection, there are eight movement signals in use.

Phase and phase sequences: A phase is a combination of non-conflict movement signals. Figure 2.6 shows an intersection with four legs and corresponding phases [8]where the right turn on each leg is unrestricted. Ring 1 and Ring 2 are two conflicting movement sets; for example, movement 1 and movement 5 can be compatible in the same phase, while movement 1 and movement 2 cannot be compatible in the same phase. A phase sequence is a sequence of phases that defines a set of phases and their order of change.

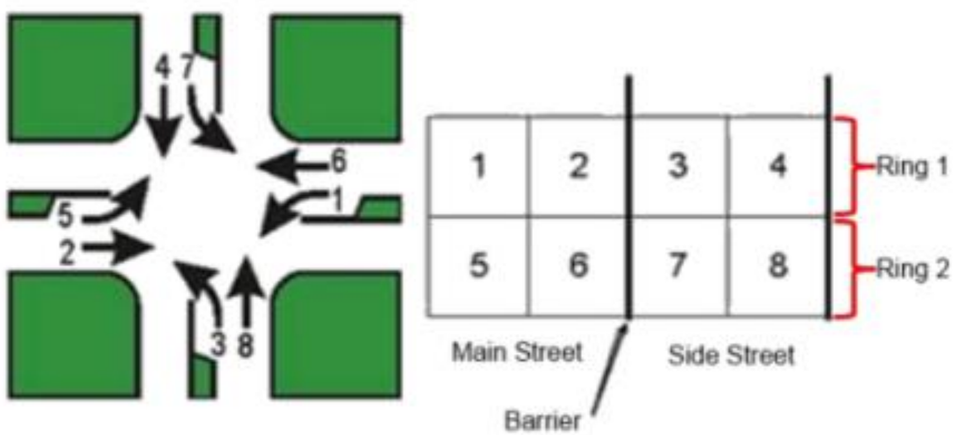


Figure 2. 6 An intersection with four legs and corresponding phases [8]

- **Signal Plan:** A signal plan for a single intersection is a sequence of phases and their corresponding starting time. A signal plan is denote as (p_1, t_1) (p_2, t_2) ... (p_i, t_i)..., where p_1 and t_1 stand for a phase and its starting time.

The objective of traffic signal control is to facilitate the safe and efficient movement of vehicles at the intersection. Safety is achieved by defining phases and appropriate yellow light time and all-red time. As for the efficiency, various measures have been proposed to quantify the efficiency of the intersection from different perspectives:

- **Travel time**: In traffic signal control, travel time of a vehicle is defined as the time difference between the time one car enters the system and the time it leaves the system. One of the most common goals is to minimize the average travel time of vehicles in the network.
- **Queue length**: The queue length of the road network is the number of queued vehicles in the road network.
- **Number of Stop:** The number of stops of a vehicle is the total occurrences of stopping (or speed being less than a threshold) that a vehicle experiences.

2.2.2 Fixed-time Traffic Signal Control (FTTSC)

Traffic lights are an integral part of the modern-day traffic infrastructure that controls traffic flow by adjusting signal timings. A poorly designed signal control could cause traffic jams and incidents at intersections. In the past few decades, many efforts have been made to design an efficient traffic signal control method to reduce vehicle delays. Traditional traffic signal approaches are mainly rule-based. Experts developed mathematical models to determine the optimal signal timings using historical traffic observations which simply change signal phases based on traffic state thresholds.

For a single (isolated) intersection, the most traditional traffic signal control method is fixed-time traffic signal control, which usually consists of a pre-timed cycle length, fixed cycle-based phase sequence, and phase split. Webster's method [9] is one of the widely used methods to calculate the cycle length and phase split for a single intersection. Assuming the traffic flow is uniform during a certain period (i.e., 5 or 10 minutes), it has a closed-form solution shown in Eq. (2. 4) and (2. 5) to generate the optimal cycle length and phase split for a single intersection that minimizes the travel time of all vehicles passing the intersection.

The calculation of the desired cycle length is relying on Webster's Equation:

$$C_{des}(V_c) = \frac{N \times t_L}{1 - \frac{V_c}{\frac{3600}{h} \times PHF \times (v/c)}} \tag{2.4}$$

where N is the number of phases; t_L is the total loss time per phase, which can be treated as a parameter related to the all-red time and the acceleration and deceleration of vehicles; the parameter h is saturation headway time (seconds/vehicle), which is the smallest time interval between successive vehicles passing a point; PHF stands for peak hour factor, which is a parameter measuring traffic demand fluctuations within the peak hour; and the parameter v/c is desired volume-to-capacity ratio, which indicates how busy the intersection is in a signal timing context. These parameters usually vary in different traffic conditions and are usually selected based on observations. Once the cycle length is decided, the green split is then calculated to be proportional to the ratios of critical lane volumes served by each phase, as indicated in Eq. (2.5):

$$\frac{t_i}{t_j} = \frac{V_c^i}{V_c^j} \tag{2.5}$$

where t_i and t_j stand for the phase duration for phase i and j, respectively.

Also, there are other wildly used methods to calculate the optimal cycle length and green split, including GreenWave [10], Maxband [11], and so on, but they would not be discussed in detail in this thesis.

In summary, FTTSC strategies are based on observed daily patterns of traffic flow parameters such as volumes and speeds. The signal timing for each phase is already predefined based on mathematical models and historical traffic observations. FTTSC cannot respond to the presence of vehicles or pedestrians at the intersection, and is recommended in downtown areas, central business districts, and urban areas in which pedestrians are anticipated or traffic demand along each phase is high. The advantage of FTTSC is that it is the most stable traffic signal control method and performance well under the steady and heavy traffic conditions. However, the disadvantages of FTTSC are also obvious, it is lacking the flexibility and non-sensitive to the real-time traffic volume. FTTSC is now wildly used in some developing countries, like China and India, where the traffic is usually heavy and mixed with pedestrians and other non-motorized vehicles.

2.2.3 Fully Actuated Traffic Signal Control (FATSC)

Fully actuated traffic signal control (FATSC) is another type of traditional control method. Different from FTSC, the signal timing is not fixed and is entirely influenced by traffic volumes. The traffic condition is detected through sensors at all the approaches. Qing *et al.* [16] formulated a mixed-integer linear program (MILP) that explicitly accommodates multiple priority requests from different modes of vehicles and pedestrians while simultaneously considering coordination and vehicle actuation. Another work [17] proposed to use vehicular ad hoc networks (VANETs) to collect and aggregate real-time speed and position information on individual vehicles to optimize signal control at traffic intersections. Moreover, Khayatian et al. [18] described a new approach to control traffic signals at isolated intersections by capturing vehicles' delay times and utilize them to adjust the green times.

2.2.4 Adaptive Traffic Signal Control (ATSC)

Excepted from FTSC and FATSC, there are also many studies on adaptive traffic signal control (ATSC) method. Cools et al. [19] and Gershenson et al. [20] proposed the Self-Organizing Traffic Signal Control (SOTSC), where the main difference between SOTSC and FATSC is on the definition of request on the current phase: in FATSC, the request on the current phase will be generated whenever there is a vehicle approaching the green signal, while in SOTSC, the request will not be generated unless the number of vehicles approaching the green signal is larger than a predefined threshold.

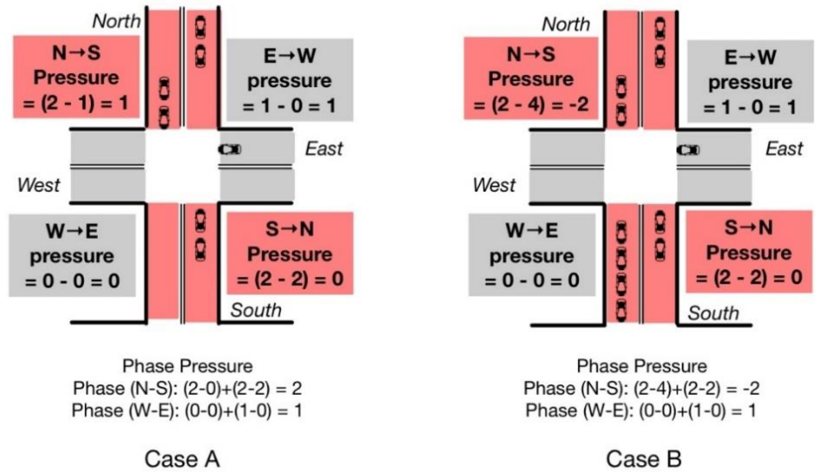


Figure 2. 7 Illustration of max pressure control in two cases. In both cases, there are four movement signals: North→South, South→North, East→West and West→East and there are two phases: *Phase* $(N - S)$ which sets green signal in the North→South and South→North direction, and *Phase* $(N - S)$ which sets green signal in the East→West and West→East direction. In Case A, MPSC selects *Phase* $(N - S)$ since the pressure of *Phase* $(N - S)$ is higher than *Phase* $(W - E)$; in Case B, Max-pressure selects *Phase* $(W - E)$. [21]

Another ATSC is called max-pressure traffic signal control (MPTSC) method. MPTSC [21] aimed to reduce the risk of over-saturation by balancing queue length between neighboring intersections (or by minimizing the maximum "pressure" of the phases for an intersection). The concept of pressure is illustrated in Figure 2.7. Normally, the pressure of a movement signal can be defined as the number of vehicles on incoming lanes (of the traffic movement) minus the number of vehicles on the corresponding outgoing lanes; the pressure of a phase is defined as the difference between the total queue length on incoming approaches and outgoing approaches. By setting the objective as minimizing the pressure of phases for individual intersections, MPSC is proved to maximize the throughput of the whole road network.

Traditional traffic signal methods, including fixed-time signal control, fully actuated signal control, semi-actuated signal control, self-organizing signal control, and the max-pressure signal control, are well developed. Some of them are wildly used in the real world. However, the traffic environment is highly dynamic and complex. These traditional signal control approaches can hardly adapt well to the real-world traffic conditions.

2.2.5 Reinforcement Learning-based Traffic Signal Control (RLTSC) for Isolated Intersection

Thrope [29] first attempted to apply the state-action-reward-state-action (SARSA) RL algorithm to control traffic signals for an isolated intersection in 1997 and found it outperformed the fixed-time controller in terms of reducing vehicle waiting time. Since then, several RL-based traffic signal control algorithms have been proposed to optimize signal control strategies. For example, Abdulhai et al. [30] used the Q-learning to control an intersection with two signal phases. They defined the state as the queue length and reward function as the total delay caused by vehicles between two successive decision points. The results showed that the proposed controller can significantly reduce the average travel delay under variable traffic flows compared with the fixed-time controller.

When researchers tried to solve TSC using the RL, the most popular method was the Q-learning. However, given the highly varied traffic environment, it is impossible to trace all state-action pairs for traffic signal control problems. Currently, combining RL and deep learning algorithms becomes a paradigm to solve intractable high-dimensional problems. In the past years, various deep neural network structures, such as convolution neural networks (CNN) and recurrent neural networks (RNN), have been used to automatically extract traffic state information for the RL to effectively learn traffic signal control policies. For example, Li et al. [31] applied the deep-stacked auto-encoders neural network to estimate the state-action function. It took the queue length on each lane as inputs and output the Q-value for actions of either keeping the current phase or changing to another phase. The experimental results showed that the DRL method outperforms the conventional Q-learning method. Mousavi et al. [32] proposed two types of DRL algorithms: a value-based and a policy-based approach to control signals at a single intersection. Both methods used an image-like state representation based on the positions and speeds of vehicles, and applied a CNN model to compress those high-dimensional traffic states.

2.2.6 Reinforcement Learning-based Traffic Signal Control (RLTSC) for Multiple Intersections

Many efforts have been devoted to designing agent-based RL algorithms to coordinate signals at multiple intersections. The joint control method [33] used a single agent to manage multiple intersections. The agent learned the joint actions based on the global traffic state information. This method is highly sensitive to the data quality at intersections. If the data collection sensors at one intersection fails, it might lead to control performance deterioration at other stations or even model collapse. Aslani et al.

[34] and Gong et al. [35] applied a fully independent control method using an independent RL agent. Each controller decided it action solely based on the local traffic state information without considering partial state information of neighboring intersections. This method is easy to scale up to a road network compared with the joint control method. In [36], [37], the partial environment information and control strategy of neighboring intersections are shared to target intersections to assist agents in making decisions.

In general, the traditional RL and advanced DRL methods have been successfully applied to traffic signal control at both single intersection and multiple intersections scenarios. However, challenges remain for designing RL-based traffic signal control algorithms to improve the flexibility of traffic light transitions and intersection mobility

2.3 Autonomous (Signal-free) Intersection Control

Advances in AVs and intelligent transportation systems indicate a rapidly approaching future in which intelligent vehicles will automatically handle the process of driving. However, increasing the efficiency of transportation infrastructure will require intelligent traffic control mechanisms that can help AVs passing through intersections smoothly. By applying vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, the behaviors of multiple connected vehicles can be coordinated at the intersection area. A highly cited paper on autonomous intersection management is [24]. In the paper, the authors suggested an alternative mechanism for coordinating the movement of autonomous vehicles through intersections. In this multiagent system, intersection uses a **reservation-based** approach built around a detailed communication protocol. The main idea for the reservation-based management is that the AVs must make requests to reserve the traversing of intersections from both the spatial and temporal perspectives. Figure 2.8 shows a general flow chart for reservation-based autonomous intersection management.

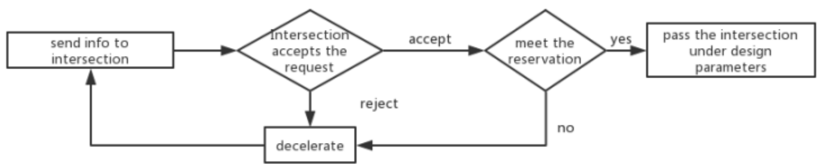


Figure 2. 8 Flow chart for reservation-based autonomous intersection management

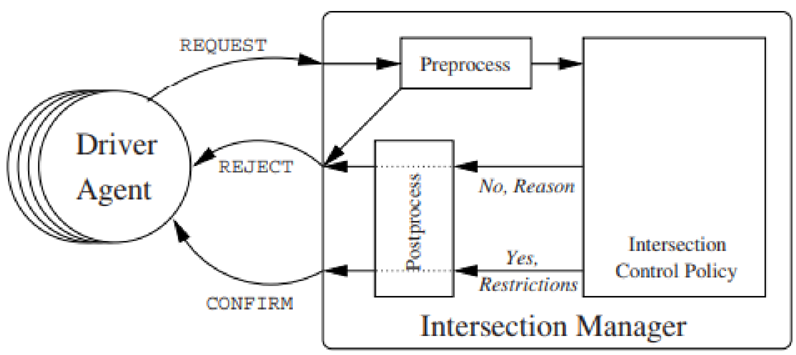


Figure 2. 9 Interaction between Driver Agent and Intersection Manager

In [24], the problem is formed as a multiagent system, and an intersection manager is placed at each intersection. The driver agent attempts to reserve a block of space-time in the intersection. According to an intersection control policy, the intersection manager decides whether to approve or reject the reservation reserve requested. Figure 2.9 shows one interaction between a driver agent and the intersection manager. Different scheduling mechanisms [25] were further proposed to improve the “first come, first served” (FCFS) policy commonly used in resource reservation. Moreover, trajectory planning-based methods [26], [27] were also proposed to achieve collision-free passing by eliminating potential trajectory overlaps and cross-collision risks. Instead of reserving the whole intersection, these two papers require VAs to reserve the collision points inside the intersection, which can further increase the utility rate at the intersection. Figure 2.10 [28] demonstrates the conflict points in a typical 4-lane 4-leg intersection. The red points represent conflict points. As shown in the figure, there are 20 conflict points for approaching vehicles from 4 different approaches.

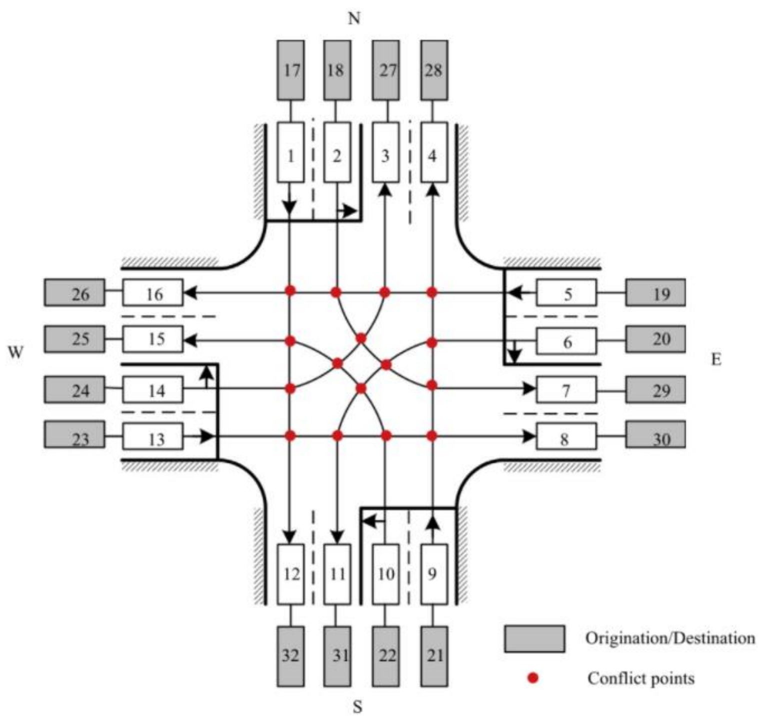


Figure 2. 10 Demonstration of conflict points in a typical 4-lane 4-leg intersection

2.4 Network Traffic Management

This section reviews some previous work in network traffic management. In the past few decades, a great amount of attention has been paid to study aggregated route choice problems. Mahmassani and Chang [38] proposed a boundedly rational user equilibrium (UE) in a transportation network where all users were satisfied with their current travel choices. An alternative approach for UE presented in [39] applied the Game Theory, where the network users "played through" all the possible eventualities before selecting their best routes. Stemmed from UE, stochastic user equilibrium (SUE) and dynamic user equilibrium (DUE) have also been wildly studied. De Cea and Fernández [40] presented an SUE assignment model for the route choice problem on the congested system. They defined an absolute capacity for each link that cannot be exceeded in practice. Huang and Lam [41] considered a simultaneous route and departure (SRD) time choice equilibrium assignment problem in the network. The main challenge for such DUE problems is to estimate the real-time traffic flow on each link. Wang et al. [42] employed a multi-agent system approach to determine the next turning direction from the route assignment perspective. The system proposed in [42] focused on the re-routing process if downstream road congestion was detected.

Compared with traditional UE formulations, the approach used in this thesis can get the time-dependent traffic volume on each link easily since the vehicle routing is formulated as a reservation-based problem, which means every CAV must reserve every link on its route for a certain amount of period before starting its trip. Reservation based method has been widely wildly applied in the autonomous intersection management problems. For example, Dresner and Stone [24] resolved CAVs' conflictions at signal-free intersections by requested approaching vehicles to reserve spatial and temporal occupancies via a first come, first served (FCFS) policy. Also, other work focused on vehicles passing through an intersection in platoons [43] also apply the reservation-based management method to reserve the conflict points inside the intersection area. Jin et al. [44 – 46] presented an intersection management system using a multi-agent system approach. The major modules in their system included a vehicle behavior planning module, a dynamic intersection time-space reservation module, and a vehicle trajectory planning module.

Chapter 3 Traffic Signal Control Using Deep Reinforcement Learning

3.1 Introduction and Motivation

Traffic signal control is a fundamental and challenging real-world problem that manages the traffic at intersection areas by adjusting signal timing and phases sequence. With the emergence of Connected and Automated Vehicles (CAVs) and Vehicle-to-Infrastructure (V2I) communication, richer real-time information is becoming available (vehicle's speed and position), which can be applied to mitigate traffic congestion. This chapter proposes a deep reinforcement learning (DRL)-based traffic signal control (TSC) at an isolated intersection, where vehicles' position and speed are processed by a convolution neural network (CNN) and fed into the RL system as inputs. The innovation point in this method is that this thesis introduces a flexible traffic light state dual-ring controller to maximize the flexibility at the intersection. Four different reward functions are designed and compared with the traditional TSC methods, including fixed-time and actuated TSC, under two traffic demands. As for the experiment, real-world intersection configurations are imported in Simulation in Urban Mobility (SUMO) and the traffic demands are generated based on real-world conditions. The result shows that our DRL model outperforms the traditional traffic signal control method in terms of average travel delay, emission ($CO_2, CO,$ and NO_X), and fuel consumption.

Traffic congestion has continued to increase worldwide over the past decade – primarily due to population growth and ongoing urbanization. The increasing congestion level forces scientists and engineers to rethink how to effectively utilize the existing traffic infrastructure to improve mobility within the limited resources, and traffic lights system is an integral part of the modern-day traffic infrastructure that controls traffic flow by adjusting signal timings and signal sequences. However, only relying on traditional TSC methods cannot perceive and react to real-time traffic patterns and need to manually change the traffic signal timings in the signal control system under certain conditions. A poorly designed signal control could cause traffic jams and accidents at intersection areas. Thus, developing intelligent TSC systems that can correctly react to real-time traffic conditions is vital to improve safety and mitigate traffic congestion.

Fortunately, the emergence of Connected and Automated Vehicles (CAVs) , and Vehicle-to-infrastructure (V2I) communications have unlocked uncountable opportunities to improve TSC in terms of mobility, efficiency, and environmental sustainability. For example, CAVs' real-time positions and speeds can be obtained under the fully CAVs environment, which are the essential resources to improve the TSC system. At the same time, with the recent successes in reinforcement learning techniques, there is an increasing interest in using RL to improve traffic signal control. RL can directly learn from the observed data without making any strong assumptions. This trial-and-error approach is also suitable for solving TSC problems. However, the current studies for RL-based TSC are lacked the flexibility of traffic light transitions and do not give the agent enough freedom to directly learn from the environment.

3.2 Problem Formulation

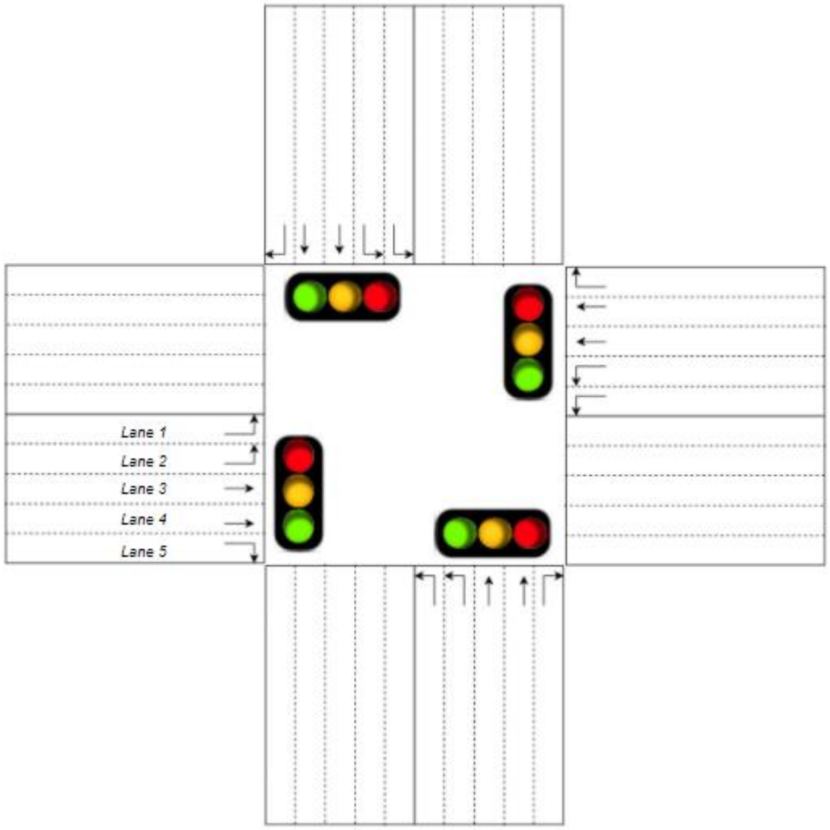


Figure 3. 1 Geometric layout of an isolated intersection.

This chapter considers a deep reinforcement learning-based traffic light control problem. Figure 3.1 shows the geometry of the intersection in the study. Each approach has five lanes, including two dedicated left turn lanes (*Lane 1 and Lane 2*), two dedicated through lanes (*Lane 3 and Lane 4*), and one dedicated right turn lane (*Lane 5*). Vehicles entering the intersection are managed by a traffic signal, including green, yellow, and red lights, which correspond to different movements from each approach. Traffic signals are grouped into several signal phases to guarantee safety at the intersection area. The set of all signal phases at the intersection is denoted by M. In some previous studies, they assumed that the phases sequence is predefined. However, fixed phases sequence may limit the RL agent to seek the optimal policy that minimizes vehicles' delay at the intersection. Thus, the flexible phases sequence is applied in the study.

Besides, the minimum and maximum green times of phase $m \in M$, namely, $t_{m,gmin}$ and $t_{m,gmax}$ are set to use as the lower bound and upper bound for each phase duration. Figure 3.2 is an illustration of signal phase diagram. In this case,

$$M = \{[1,5], [1,6], [2,5], [2,6], [3,7], [3,8], [4,7], [4,8]\}$$

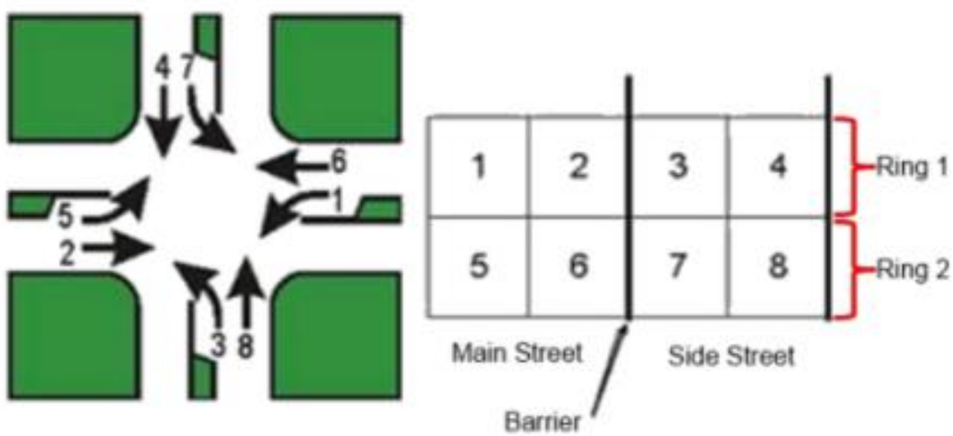


Figure 3. 2 Signal phase diagram and dual ring controller [8]

As for the dual ring controller, the two rings in Figure 3.2 are {1,2,3,4} and {5,6,7,8}, which are two sets of self-conflicting movement. Two movements from different rings are active, one from each ring, at any given time. The two rings operate independently, bounded by each movement's minimum and maximum green time. At every time step, the active movements are must from the same side of barriers (Main Street or Side Street);

for example, movement 1 and movement 6 can be activated simultaneously at the same time, while movement 1 and movement 7 cannot be activated simultaneously.

The objective of this research is to propose a deep reinforcement learning-based algorithm to control a flexible traffic light state dual ring at isolated intersections to maximize the number of vehicles passing through intersections and minimize the imbalance of the number of vehicles belong to different signal phases.

3.3 Methodology

This section models the TSC as a DRL problem and proposes a DQN agent-based algorithm to learn the optimal control policy and a knowledge-based decision-maker Algorithm to prevent unnecessary traffic signal transitions.

3.3.1 Model

This section uses the Markov decision process (MDP) framework to model the TSC problem in an RL context. The MDP is defined by a tuple $< S, A, P, R, \gamma >$, where S is the state space, A is the action space, P is the state transition dynamic model, R is the reward function and $\gamma \in [0, 1]$ is the discount factor. The traffic signal control at signal intersection is formulate as a RL problem. A RL agent will learns the control strategy independently.

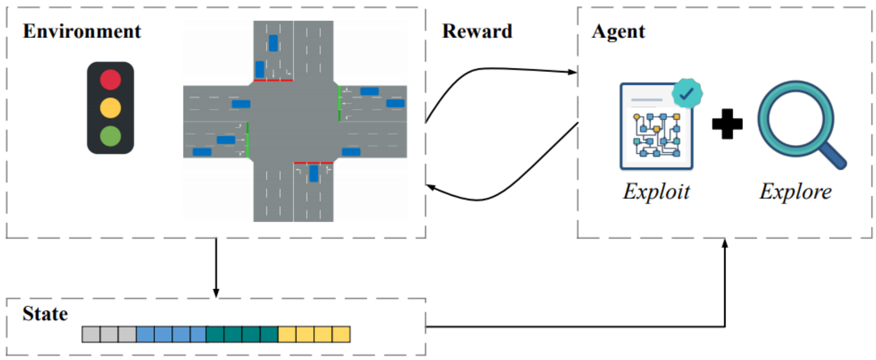


Figure 3. 3 Reinforcement learning based traffic signal control framework

At a time step $t \in T$, the agent I receives the traffic state s^t from the environment. Given the state s^t, the agent takes an action a^t based on the policy π. The traffic environment transit from state s^t into a new state s^{t+1} according to the state transition dynamics $P(s^{t+1}|s^t,\ a^t)$, and the agent receives a reward r^{t+1}. The goal of the learning-based agent I is to find an optimal policy π^* that maximizes the expected cumulative rewards $E[\sum_t^T \gamma R(s^t)]$ from the time step t onward.

a) State space S

The selection of state space is important since it determines agent decisions. For the traffic signal control problem, the states could be traffic information collected by road infrastructures such as loop sensors and cameras, including speeds, positions, and queue lengths. This study is deployed under the fully CAVs environment, and all vehicles' dynamics information can be real-time and accurate obtained via the V2X communication. This DRL model selects CAVs speed, position, and traffic light state as the state space, A new state will be generated at the beginning of each time step.

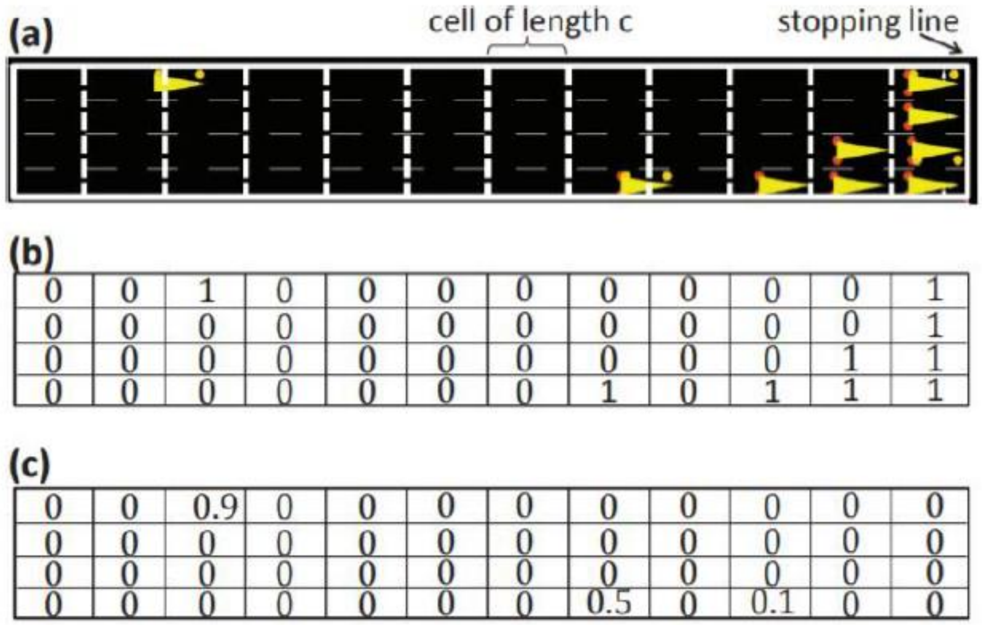


Figure 3. 4 Demonstration of traffic state conversion into state space

All the lanes joining at the intersection are discrete into a square grid with a length c. The value of c is selected to be the average length of regular vehicles that make sure no two vehicles fall into the same cell. Figure 3.4 demonstrates an example of the speed and position matrices from the eastbound edge. Each cell in Figure 3.4(a) contains a two-tuple to store the position and velocity shown in Figure 3.4(b, c). As for the position matrix, the value in each cell could be 0 and 1, depending on the corresponding cell's occupancy condition in Figure 3.4(a). The position value will be 0 if no vehicle is present and 1 if a vehicle is present in the cell. As for the velocity matrix, it stored the speed normalized by the speed limit of the vehicle present in the perspective cell. The sizes of the speed matrix and position matrices remain the same $(m \times n)$, where m is the number of lanes from every approach $(|I|)$ and $n = L/c$, where L is the length of road segments near the intersection. Besides these two matrices, the third element in the state space is the traffic light states vector. It is a $1 \times |M|$ vector, where $|M|$ is the total number of phases at the intersection. The corresponding position in the traffic light states vector of the current active phase is set to 1, and other values in the vector are set to 0.

b) Action space A

There are several common ways to define actions for traffic signal controllers [48], including 1) randomly switch to a signal phase among a set of all phases, change to the next phase based on pre-defined phase sequence, and 3) set the current phase duration. In this study, the phase sequence is not predefined, and it will always switch to the phase which can maximize the reward. The action space is defined as a $1 \times |M|$ matrix. The agent I selects an action among all the phases at each time step, and the corresponding position in the action matrix is set to 1 while other values in the matrix are set to 0. To guarantee the safety at intersection, the yellow light state with duration t_{yellow} and all-red state with duration t_{allred} are triggered when the traffic signal transit from one phase to another phase. In addition, minimum green time $t_{m,gmin}$ and maximum green time $t_{m,gmax}$ are predefined for each phase separately to supervise the behaviors of RL agents.

c) Transition dynamics *P*

The state transition *P* is a critical element in the RL problem. It shows how the environment reacts to the action taken by the RL agent. Applies to the TSC problem, the state transition shows how the traffic light transfers from one state to another. In this study, the traffic light transition is controlled by a flexible dual ring controller, as shown in Figure 3.5. This controller can provide maximum flexibility for the traffic light transitions. In most previous studies, any two non-conflict movement must be activated and inactivated together, limiting the agent to learn the optimal policy. The traffic light transitions in this study do not necessarily have a strict coupling of movement such as movement 1 & movement 6 or movement 2 and 6. Any movement can be activated with any non-conflict movements from the same side of barriers. In Figure 3.6, there are in total 33 possible traffic signal states, which are represented as circles. Green circles occur where two green movements are activated; yellow circles occur where at least one movement is yellow and red circles appear where all traffic light is red or if all but one signal phase is red. The arrows in the figure show how the traffic light is transited from one state to another state, and the blue lines are different from the traditional dual ring controller. Those transitions also do not exist in other DRL-TSC studies. The RL agent can always transfer to the most appropriate traffic state based on real world traffic conditions by applying this flexible controller.

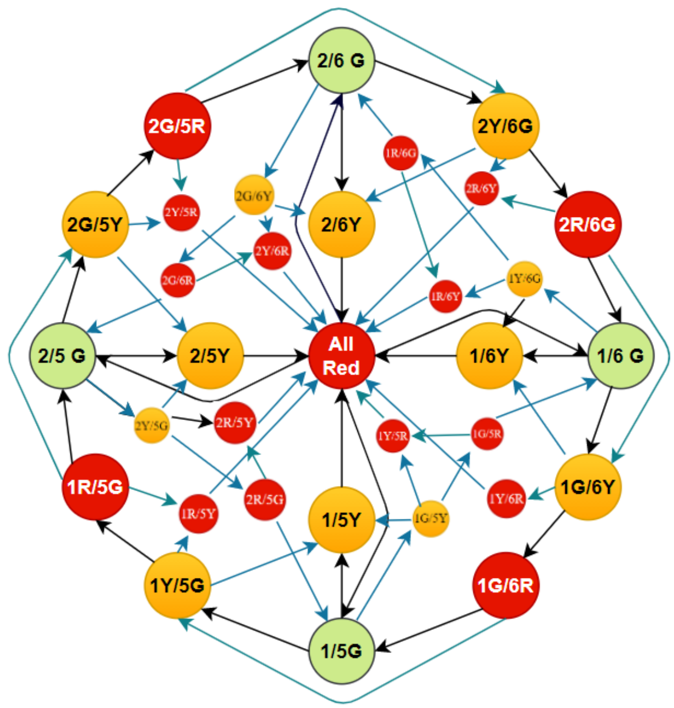


Figure 3. 5 Main Street Flexible dual ring controller, adapted from [49]

Figure 3.5 shows the dual ring controller for the Main Street. As for the Side Street, it is identical to the main street half and when the traffic light cross from one side of barrier to another side of the barrier, an all-red state must occur which is shown in Figure 3.6.

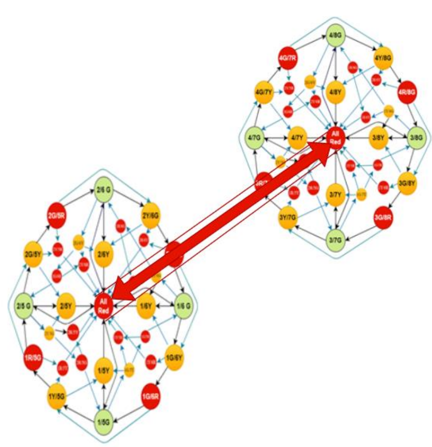


Figure 3. 6 Illustration of dual ring connectivity between Main Street and Side Street

In this research, the agents are trained on a microscopic traffic simulation platform, Simulation in Urban Mobility (SUMO) [50]. The dynamics of vehicles in SUMO are built by a car-following model that considers drivers' physical and psychological factors. Given the decision of traffic light, vehicles will adjust their trajectory according to the car-following model. With the movements of vehicles and the transitions of traffic signals, the traffic environment transits from one state into another state during every time step.

d) Reward function R

The reward function determines not only the convergence speed of RL algorithms and the intersection performance as well. The average travel delay, average waiting timing, and queue length are often considered as the objective functions [50]. Though some information may be hard to obtain in the real world, with the fully CAVs environment in the future, these individual vehicle levels information can be easily obtained. The reward function directly influences the performance of the model. To figure out which type of reward function is the most suitable for the proposed model, this study defines four different reward functions:

1) Lane level max queue length related

$$r_t^1 = \left| max_{i=lane\ id} \left\{q_t^{a,i}\right\} - max_{j=lane\ id} \left\{q_t^{b,j}\right\}\right| \tag{3.1}$$

2) Lane level average queue length related

$$r_t^2 = \left| \sum_{i=lane\ id} q_t^{a,i} /|i| - \sum_{j=lane\ id} q_t^{b,j} /|j| \right| \tag{3.2}$$

3) Waiting time related

$$r_t^3 = r_a - r_b \tag{3.3}$$

4) Combination of r_t^2 and r_t^3

$$r_t^4 = w_1 * r_t^2 + w_2 * r_t^3 \tag{3.4}$$

Where $q_t^{a,i}, q_t^{b,j}$ are the number of queued vehicles at time step t for phases a, b respectively; $r_a,\ r_b$ are the cumulated waiting time for vehicles corresponding to phases a, b.

3.3.2 Algorithms

Two algorithms are developed for the proposed DRL-TLC problem. The first is a model-free DRL algorithm that uses Q-learning to find the optimal policy. The second is a knowledge-based decision algorithm that helps the RL learn the policy and prevent the frequent change of traffic lights.

a) Deep Q-learning Traffic Light Control Algorithm

A model free DRL algorithm, called Q-learning, is applied in the study to guide traffic signal controller to select control policy. The Q-learning uses a Q-value function $Q(s, a)$ to evaluate how good an action is in each state. The Q-value function is update in Equation (3.5):

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r_i + \gamma max_{a'} Q(s', a') - Q(s, a)] \tag{3.5}$$

Where s and s' are the current and next state, a and $a'$are the current action and action for next state, r_i is the immediate reward based on which type of reward function is chosen, $max_{a'}Q(s', a')$ is the maximum possible reward for the next state, $\alpha \in [0,1]$ is the learning rate, and $\gamma \in [0,1]$ is the discount factor.

One challenge for using Q-learning to solve the DRL-TSC problem is that the traffic environment is extremely complex and highly dynamic. The traffic state changes at each time step. It is not only infeasible to store all the state-action pairs into a Q-table. It is also almost impossible to trace all the possible traffic states, even for a single intersection. Thus, the RL agent faces a non-stationary learning problem since the traffic environment of the intersection changes at every iteration. Therefore, a Deep Q-Network (DQN) is introduced as a function approximator to estimate Q-value. The architecture of DQN is shown in Figure 3.7. The network input is the observed states, including position matrix (P), speed matrix (V) and traffic light states vector (L). The first layer of convolution network has 16 filter of 4*4 with stride 2 and it applies ReLU as the activation. The second layer has 32 filter of size 2*2 with stride of 1, also allies ReLU as the activation. The third and fourth layers are fully connected layers of sizes 128 and 64 respectively. The final layer is the linear layer outputting Q values corresponding to every possible action that the agent takes.

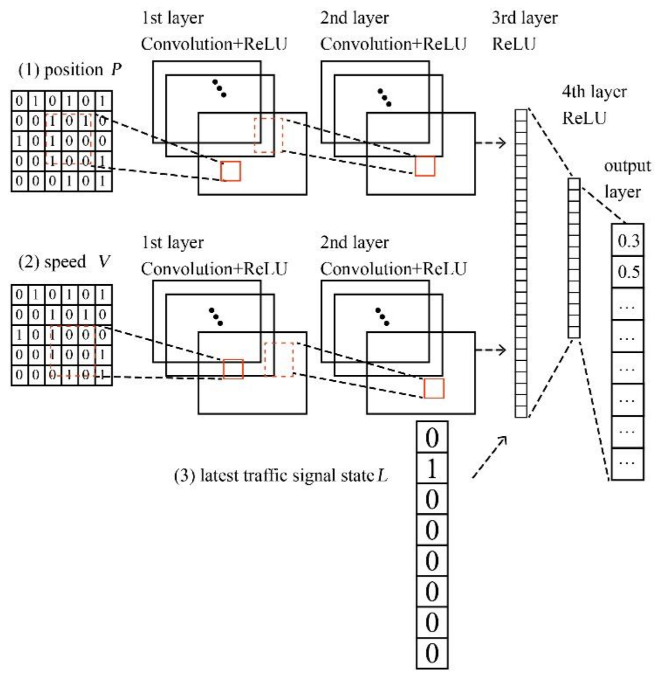


Figure 3. 7 The architecture of proposed Deep Q-Network (DQN)

As for the training of the neural network, the neural network is initialized with random weights. At the beginning of every time step, the RL agent observes the current time step state s_t, and it forms the input to the neural network and performs action a_t with the highest cumulative future reward. After performing the action, the agent receives reward r_t^i and next state s_{t+1} by the environment. Agent stores this experience (s_t, a_t, r_t^i , s_{t+1}) in memory. Oldest data is removed when memory space is full. DNN is trained by extracting training examples of type (s_t, a_t) from the memory. After collecting the training data, agent learns features θ by training the DNN network to minimize the following Mean Squared Error (MSE):

$$MSE(\theta) = \frac{1}{m}\sum_{t=1}^{m}\left\{\left(r_t^i + \gamma max_{a'}Q(s_{t+1}, a_{t+1}; \theta_{t+1})\right) - Q(s_t, a_t; \theta)\right\}^2 \tag{3.6}$$

where m is the size of the input data set. Since m is large, the computational cost to calculate MSE(θ) is large, a stochastic gradient descent algorithm RMSProp is applied with a minibatch of size of 32.

Algorithm 1: Deep Q-Network for Flexible Dual Ring Controller

---

Input: Mini-batch size B, learning rate α, discount factor γ ,exploration probability ϵ, exploration decent rate k, reward type i

Output: optimal Q-value Vector

Initialize replay memory D, random weight $\emptyset$

for *episode*=1 to E **do**

    Receive initial state s_t

    while t < T do

        if probability ϵ

            select random action a_t

        else $a_t = max_{a'}\ Q(s_t, a_t; \theta)$

        if frequent change flag = 0:

            Execute a_t

    Receive reward r_t^i, new state s_{t+1}

    Store transition $(s_t, a_t, r_t^i, s_{t+1})$ in D

    $t \leftarrow t + 1, \epsilon \leftarrow \epsilon * k$

    Sample random mini-batch of experience

    $y = r_t^i + \ \gamma max_a Q(s_{t+1}, a_{t+1}; \theta_{t+1})$

    performance a gradient descent

    update Q

---

b) Knowledge-based Decision Algorithm

The knowledge-based decision algorithm is proposed to prevent frequent traffic light transitions. Since the flexible dual ring controller is applied in the study, the state space is larger than previous work. It may cause frequent traffic light transitions, which are inefficient in traffic light management. To prevent this from happening, the RL agent has installed a knowledge-based decision algorithm. This algorithm guarantees that the traffic light will not transit to the next phase if the current phase is not fully evacuated.

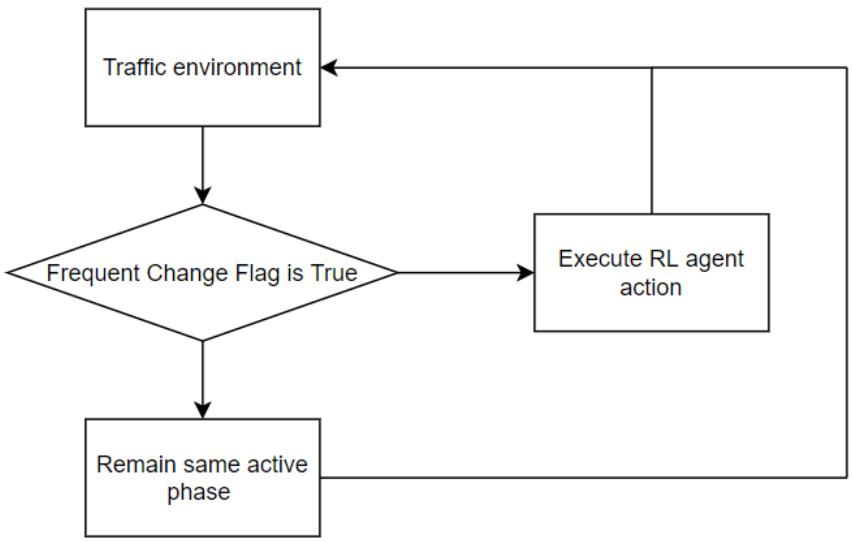


Figure 3. 8 Frequent change flag flow chart

An unnecessary traffic signal states changing indicator is introduced to supervise the agent action. The indicator is set to 1 if there are still queues in current activated phase, otherwise, indicator is set to 0. Equation (3.7) shows how the flag supervise the RLagent,

$$m = \begin{cases} a_t, & if\ f = 0 \\ m_{t-1}, & if\ f = 1 \end{cases} \tag{3.7}$$

where m is the phase executed at the current time step, a_t is the action suggested by the RL agent, and m_{t-1} is the active phase at the last time step. If f is set to true, then the action suggested by the agent will be denied and remain the same active phase; otherwise, the traffic light is transited based on the action calculated by the RL agent. Algorithm 2 summarizes the proposed Knowledge-based decision algorithm.

Algorithm 2: Knowledge-based Decision Algorithm

Input: last time step active lane level queue vehicle id list I_pre, last time step active phase m_pre
Output: Frequent Change flag f
1: Initialize Queue vehicle id list I_cur
2: **for** *lane* **in** m_pre
3: store *veh* to I_cur
4. **if** I_cur in I_pre
5: $f \leftarrow 1$ & reject a_t
6:
6: **else**
7: $f \leftarrow 0$ & execute a_t
8: I_{pre} ←get vehicle id list corresponding to a_t
9: **return** f

3.4 Simulation and Results

3.4.1 Simulation Environment

The simulations are conducted in a micro-simulation environment in SUMO. The SUMO is an advanced and flexible microscope traffic simulation software that provides a user-friendly Graphical User Interface (GUI) and an Application Programming Interface (API) - *Traci*, allowing users to design road networks and control simulation. The study uses Python scripts to code the DRL controller. The real-time traffic states information can be obtained, and traffic lights in the simulator can be controlled using *Traci.*

As for the intersection configuration, a real-world intersection scenario is imported from OpenStreetMap [51] and slight modified to make the intersection symmetrical. The intersection's final structure is shown in Figure 3.9, there are five lanes from each approach, including one combined left turn and U-turn Lane, one dedicated left turn lane, two dedicated through lanes, and one dedicated right turn lane. The minimum and maximum green time ($t_{m,gmin}$ and $t_{m,gmax}$) are set to be $10s$ and $40s$, the yellow lights duration and all-red lights state duration are set to be $5s$ and $1s$, respectively.

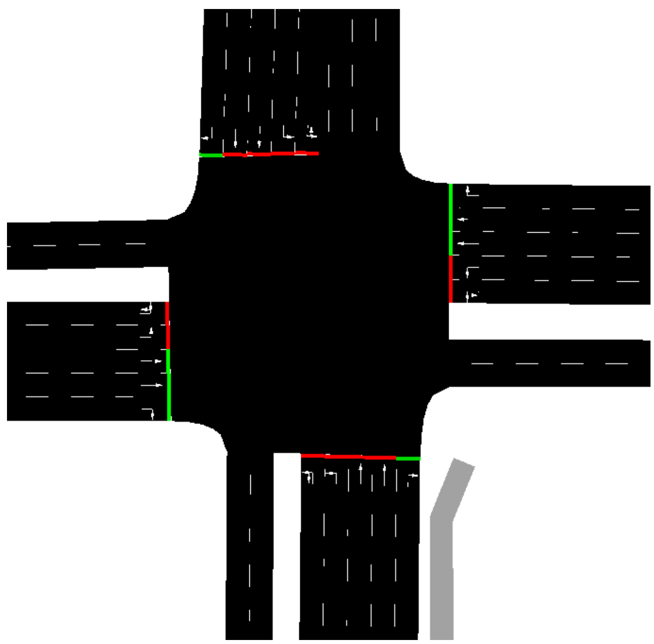


Figure 3. 9 Configuration of intersection used for simulation

3.4.2 Baseline Traffic Light Control Methods

In this study, the proposed DRL-TSC is compared with two types of traditional traffic signal control methods, fixed-time traffic light control and actuated traffic light control methods.

a) Fixed-time traffic signal control (FTTSC)

For FTTSC, the optimal signal timing plan is pre-determined based on historical traffic flow information and does not change with the varying traffic flow. For a fair comparison, the real-world traffic timing is applied at the same intersection.

b) Actuated traffic signal control (ATSC)

For ATSC, the signal controller reacts to real-time traffic flow, and each phase can be actuated. Thus, loop sensors are installed under every lane near the intersection. The loop sensor is installed 100m from the stop line. To fairly compare with proposed DRL-TSC, each phase's $t_{m,gmin}$ and $t_{m,gmax}$ are set as same in the DRL controller.

3.4.3 Traffic Demand

As for the traffic demand, two types of traffic demands are introduced to test the robustness of the proposed DRL controller. The first traffic demand is imported from the real world. The table 3.1 is the real-world traffic demand at weekday afternoon's peak hours. The traffic demands from 4:30 pm to 5:30 are selected and imported to the simulation environment. In total 3855 vehicles are passing through intersections from this time period, and the traffic demands varies every 15 minutes. As for the second type of traffic demand, the total number of vehicles remains the same but imbalanced the proportion of traffic demands every 15 minutes. Table 3.2 shows the different traffic proportions for two types of traffic demands.

Table 3. 1 Real-world Traffic Demand at Weekday Afternoon Peak Hours

PM

NS/EW Streets:	Chicago Ave			Chicago Ave			University Ave			University Ave							
	NORTHBOUND			SOUTHBOUND			EASTBOUND			WESTBOUND				UTURNS			
	NL	NT	NR	SL	ST	SR	EL	ET	ER	WL	WT	WR	TOTAL	NB	SB	EB	WB
LANES:	2	2	1	2	2	0	2	2	1	2	2	1					
4:00 PM	43	99	29	40	185	14	39	152	61	52	81	9	804	0	2	11	9
4:15 PM	56	82	36	49	180	26	53	162	63	52	74	20	853	0	3	8	10
4:30 PM	64	108	49	44	194	20	49	159	76	63	58	18	902	0	0	12	10
4:45 PM	68	111	59	53	226	14	54	206	80	65	83	7	1026	0	2	7	8
5:00 PM	62	83	48	38	191	15	52	167	69	74	96	9	904	0	1	10	7
5:15 PM	55	102	40	46	198	24	42	202	74	56	98	16	953	0	1	8	4
5:30 PM	59	95	38	32	181	19	43	175	71	45	107	15	880	0	1	7	8
5:45 PM	59	102	30	35	183	14	50	218	84	59	82	11	927	1	0	6	8
	NL	NT	NR	SL	ST	SR	EL	ET	ER	WL	WT	WR	TOTAL	NB	SB	EB	WB
TOTAL VOLUMES :	466	782	329	337	1538	146	382	1441	578	466	679	105	7249	1	10	69	64
APPROACH %'s :	29.55%	49.59%	20.86%	16.67%	76.10%	7.22%	15.91%	60.02%	24.07%	37.28%	54.32%	8.40%					
PEAK HR START TIME :	430 PM												TOTAL				
PEAK HR VOL :	249	404	196	181	809	73	197	734	299	258	335	50	3785				
PEAK HR FACTOR :	0.892			0.907			0.904			0.898			0.922				
CONTROL :	Signalized																

Table 3. 2 Traffic Proportions for Two Types of Traffic Demands

Time period	Real-world demand (RWD)	Imbalance demand (ID)
4:30 - 4:45	24.0%	35%
4:45 - 5:00	27.1%	15%
5:00 - 5:15	24.1%	30%
5:15 - 5:30	24.8%	20%

3.4.4 DRL Controller Parameters

The Deep Q-Network (DQN) architecture of the proposed controller is shown in Figure 3.7. The network input is the observed states, including position matrix (*P*), speed matrix (V) and traffic light states vector (L). The sizes of *P* and V are both 20*12, since 20 lanes approach to the intersection, each lane is discrete into 12 cells, with the length of each cell c is set to be 7m. The first layer of the convolution network has 16 filters of 4*4 with stride 2, and it applies ReLU as the activation. The second layer has 32 filters of size 2*2 with a stride of 1 and also allies ReLU as the activation. The third and fourth layers are fully connected layers of sizes 128 and 64, respectively. The final layer is the linear layer outputting Q values corresponding to every possible action that the agent takes. The outputting Q values vector has the size of 1*8 since the total number of possible phases $|M|$ is 8. Table 3.3 summarizes the parameters of RL algorithms.

Table 3. 3 Parameters of RL Model

Parameter	Value
Episodes E	300
Time step interval Δ	5s
Mini-batch Size B	32
Learning rate α	0.0002
Discount factor γ	0.95
Initial exploration rate $\in$	0.99
Minimize exploration rate $\in_{min}$	0.01
Exploration decay rate k	0.99

3.4.5 Performance Validation

In this section, the performance of proposed DRL controller is examined in terms of the average travel delay, emission (CO, CO_2, PM), and fuel consumption.

a) Average travel delay

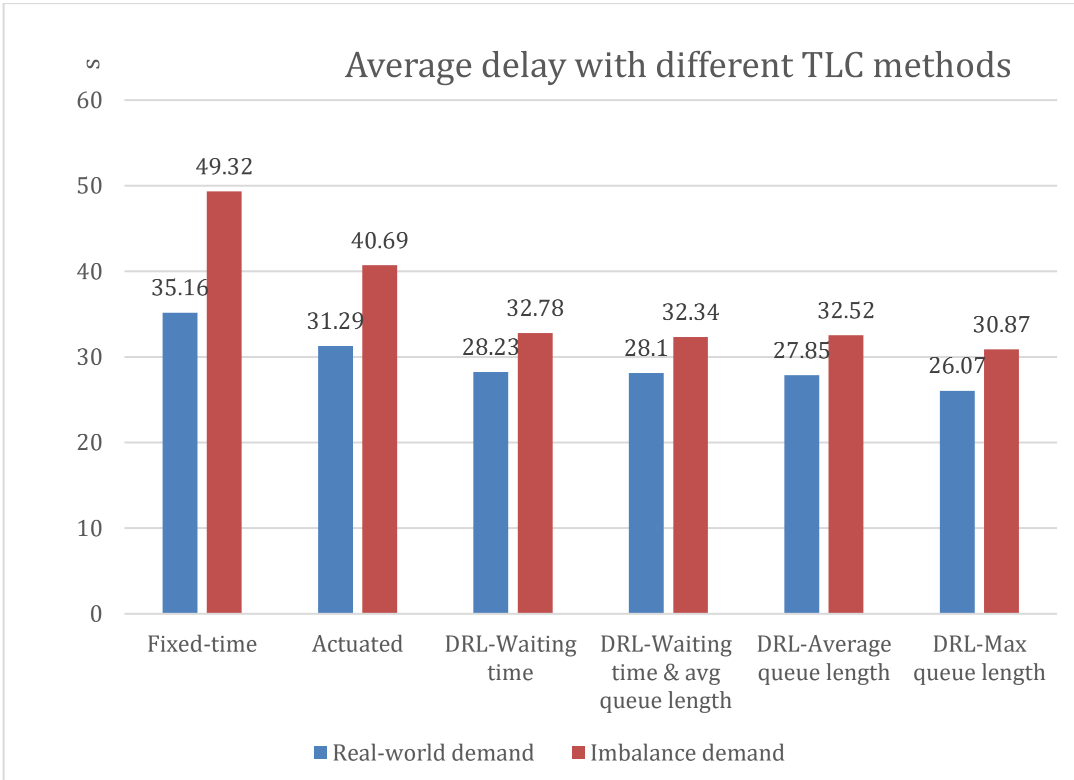


Figure 3. 10 Average travel delay with different TLC methods

Table 3. 4 Average Travel Delay and Improvement

	Fixed-time	Actuated	DRL-Waiting time	DRL-Waiting time & avg queue length	DRL – Average queue length	DRL – Max queue length
Real-world Demand	35.16 (-)	31.29 (11.0%)	28.23 (19.7%)	28.10 (20.0%)	27.85 (20.7%)	26.07 (23.8%)
Imbalance Demand	47.32 (-)	40.69 (14.0%)	32.78 (30.7%)	32.34 (31.5%)	32.52 (31.3%)	30.87 (34.7%)

Figure 3.10 and Table 3.4 present the average delay per vehicle for the six TLC methods and percentage improvement compared with FTTSC. For the FTTSC and ATSC, the average delay is the same for each simulation because the number of vehicles generated at each time step is fixed. For the DRL-based TLC control methods, all of them over-performance when compared with the traditional TLC methods (FTTSC and ATSC).

Among all DRL-based models, the one applies max queue length as the reward function is performed best. The average delay per vehicle under two traffic demands are 26.07s and 33.87s, improved 23.80% and 34.70 % respectively. The DRL-based models are performed more stable under the imbalance traffic demand. For the FTTSC, the average travel delay increased 40.2% when the traffic demand changed from real-world demand to imbalance demand, while the DRL – Max queue length method only increased 18.4%.

b) Emission and Fuel Consumption

In this part, the emission and fuel consumption related results are evaluated, including CO_2, CO, NO_X and fuel consumption. Table 3.5 presents Emission and fuel consumption-related results for different TSC methods under real-world traffic demand. Similar to the average travel delay, the DRL-based TSC control models over performance when compared with the traditional TSC method (FTTSC and ATSC). Among all DRL-based methods, the one applies max queue length as the reward function is still performed the best, the CO_2, CO, NO_X and fuel consumption are 14.7%, 18.5%, 13.5%, and 12.6% better than FTTSC, respectively.

Table 3. 5 Emission and Fuel Consumption Related Result for Different TLC Methods under Real-world Traffic Demand.

TLC Method	$CO2$ (g/mile)	CO (g/mile)	NO_X (mg/mile)	Fuel consumption(mile/gallon)
Fixed-time	417.7 (-)	2.86 (-)	478.5 (-)	18.2 (-)
Actuated	375.7 (10.1%)	2.54 (11.2%)	443.6 (7.3%)	19.2 (5.5%)
DRL – Waiting time	356.1 (14.7%)	2.37 (17.1%)	427.0 (10.7%)	20.0 (9.9%)
DRL – Waiting time & average queue length	362.5 (13.2%)	2.46 (14.0%)	438.7 (8.3%)	20.0 (9.9%)
DRL – average queue length	358.4 (14.2%)	2.35 (17.8%)	421.8 (11.8%)	20.2 (11.0%)
DRL – max queue length	354.1 (14.7%)	2.27 (18.5%)	414.0 (13.5%)	20.5 (12.6%)

Chapter 4 Signal-free Intersection Management based on First Come First Served (FCFS) Policy

4.1 Introduction and Motivation

Connected and Automated Vehicles (CAVs) can significantly improve the operation efficiency of traffic in the urban network. Specifically, the coordination among CAVs can mitigate potential conflict and reduce unnecessary stop-and-go at intersections, thus increasing the throughput of the entire network. As a result, understanding the system impacts is of unprecedented importance due to the introduction of CAVs and cooperative traffic management in the urban environment. However, it is difficult to test and verify such scenarios and strategies in the real world.

In this chapter, a signal-free intersection management strategy based on a first come, first served (FCFS) policy is developed. The goal of the management strategy is to decide the passing sequence of CAVs at each intersection if there are potential conflicts along the respective routes. A centralized autonomous intersection management strategy for all the CAVs travel inside the network is proposed, which guarantees safety at intersection areas and improves efficiency. The proposed management strategy is first tested in a numerical network environment coded by python and then tested in a miniature urban scenario in the real world. The result shows that the proposed signal-free intersection management based on a first come, first served (FCFS) policy can guarantee safety at intersection areas and improve travel efficiency.

4.2 Problem Formulation

A group of small-scale CAVs are built to travel in a miniature urban environment in this research to address the problem. As shown in Figure 4.1, the gride-shape scenario represents a typical urban traffic network with one-lane tracks and twelve intersections. During the test, four vehicles (differentiated by different colors, i.e., red, green, blue, and yellow) can recognize traffic signs with the associated color at the intersections. Through Vehicle-to-vehicle (V2V) communications, the signal-free intersection management algorithm collects the information about the states of all the vehicles. It controls CAVs access priorities by a centralized reservation-based management strategy.

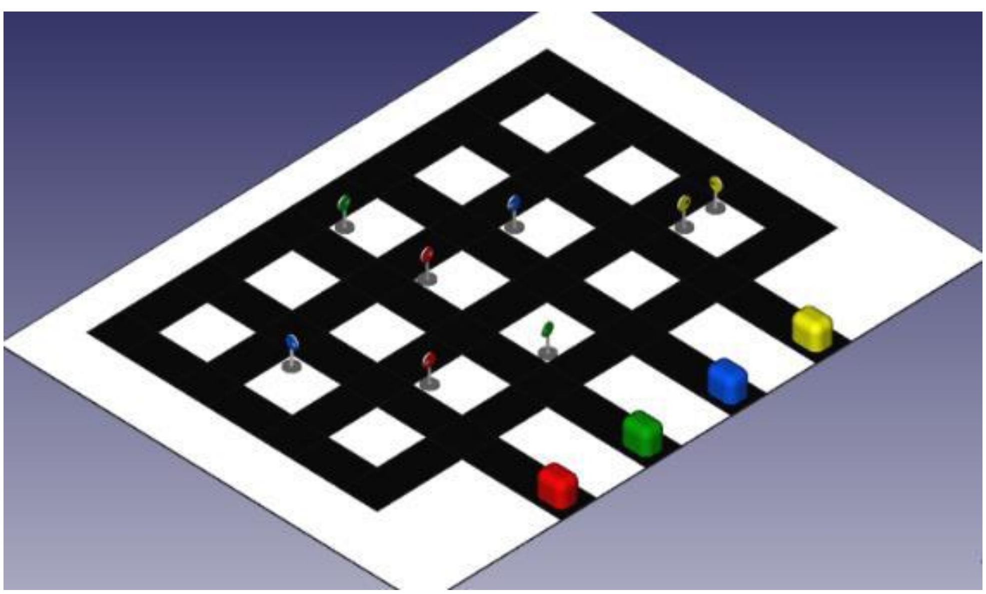


Figure 4. 1 Illustration of artificial urban environment

4.3 Methodology

4.3.1 Reservation Policy

This section introduces a centralized reservation-based signal-free intersection management strategy. The role of this strategy is to decide the passing sequence of CAVs at each intersection if there are potential conflicts along respective routes. An intersection manager is introduced to coordinate the passing sequence at intersections for CAVs inside the network.

The reservation-based method has been widely applied in the autonomous intersection management problems. For example, Dresner and Stone [24] resolved CAV's conflictions at signal-free intersections by reserving spatial and temporal occupancies via a first come, first serve (FCFS) to approaching vehicles. Also, other work focused on vehicles passing through an intersection in platoons [43] also apply the reservation-based management method to reserve the conflict points inside the intersection area. Apply the reservation-based idea to this specific case, only one CAV can reserve a particular link and intersection at each time step. Since the lanes in the designed scenario are one-way lanes, it is hard to solve the conflict when two CAVs with opposite head directions drive into the same one-way road. An example of the reservation policy is shown in Figure 4.2. In this example, the four solid circles represent four CAVs traveling in the network, and other hollow circles are the lanes and intersections they need to reserve before entering the coming intersection. At the time step shown in Figure 4.2, the next movements for those four vehicles are: left turn (Blue and Green) and through (Red and Yellow). In general, the reservation policy is defined as each CAV must successfully reserve both the coming intersection and lane before entering the heading intersection area.

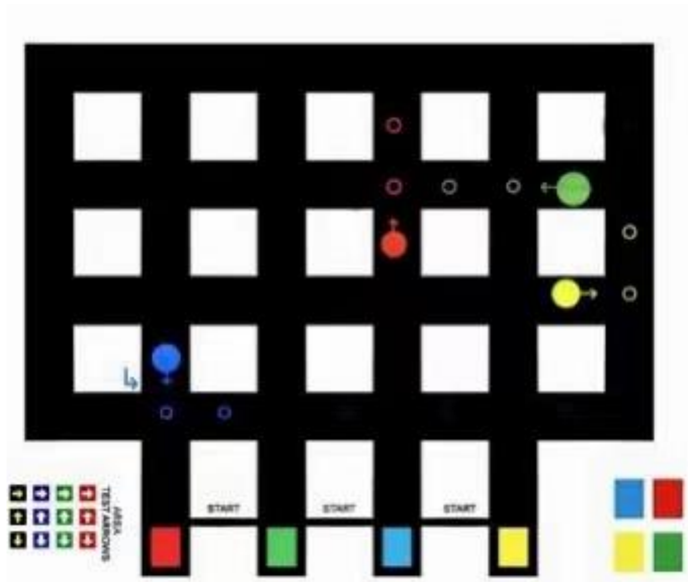


Figure 4. 2 An example of reservation policy for proposed management strategy

4.3.2 Intersection Manager

The primary responsibility of the intersection manager (IM) is to monitor the state of each CAV and decide the passing sequence of CAVs at intersections if there are potential conflicts along the respective routes. A centralized management strategy is installed in IM and a reservation matrix is introduced to store real-time reservation information. Table 4.1 shows the reservation matrix corresponding to the traffic condition presented in Figure 4.2. It is a matrix with size $m \times n$, where m and n are the width and length of the given network. If an intersection or lane is not reserved by any CAV, the corresponding position is 0; otherwise, it is set by 1, and different colors represent four CAVs reservation information, respectively.

Table 4. 1 Example of Reservation Matrix

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0

The sequence diagram of the proposed agent-based signal-free intersection management is shown in Figure 4.3. There are three steps to determine the passing sequence of CAVs at potential conflict intersections:

1) When CAV is reaching an intersection, it must send its dynamic traffic data, including vehicle position and heading direction to the IM to request permission to pass the intersection.
2) Once the IM receives the data send by CAV, it looks up the reservation matrix to check whether the intersection and lane the CAV requests to reserve are occupied or already reserved by other CAVs in the network.
3) If the intersection and lane CAV requests to reserve are not occupied and reserved by others, the IM sends the passing permission back to CAV, and CAV can pass

through the coming intersection without stop. At the same time, IM updates the reserve matrix to upload the reservation information. Otherwise, the IM rejects the request, and CAV must stop before the stop line and go back to step 1).

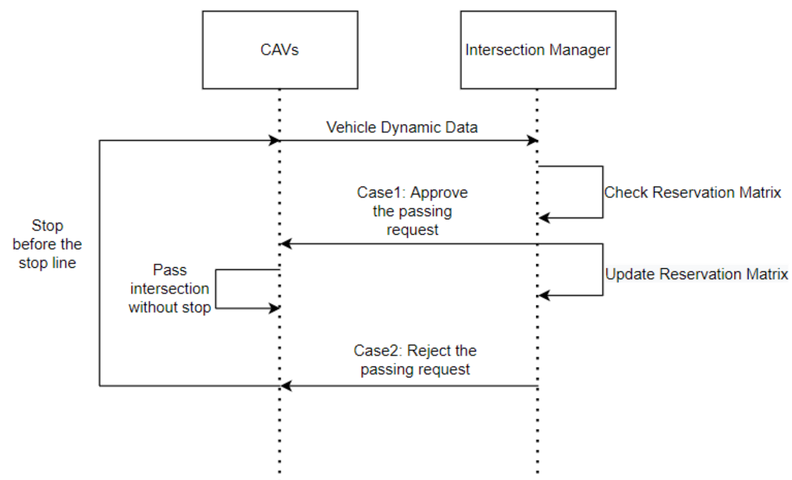
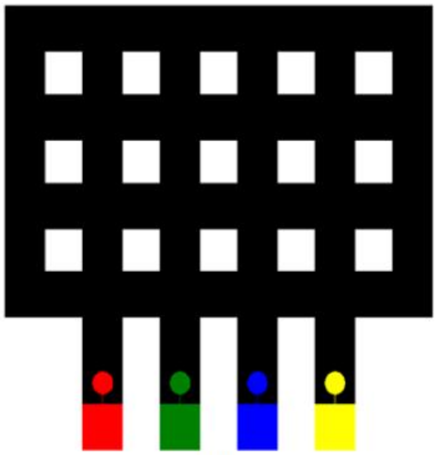


Figure 4. 3 Sequential diagram for the proposed agent-based signal-free intersection management

4.4 Simulation and Real-world Application

To verify the proposed signal-free intersection management strategy, two test scenarios are created, one is for numerical simulation in python, and another is a miniture urban environment in real world with the same network configuration.

Figure 4.4(a) is a screenshot of the numerical simulation environment; it is a typical urban traffic network with one-lane tracks and twelve intersections. Four solid circles with different colors represent four CAVs. The route for each CAV traveling inside the network is predefined and this study assumes all CAVs are equipped with V2V communications devices. Thus, the traffic management algorithm collects the states of all CAVs and controls CAVs' access priorities by sending the passing permissions.


(a)

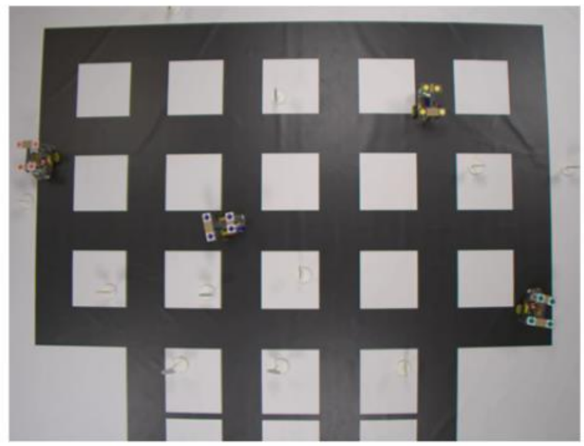

(b)

Figure 4. 4 Test Scenarios in both numerical simulation environment and real world

Figure 4.5 shows a typical example of the proposed autonomous intersection management strategy. In Figure 4.5(a), the red CAV desires to through at the coming intersection, and the blue CAV desires to turn right at the same intersection. Here a potential conflict happens since both CAVs request to reserve the same intersection in an overlapped time window. In this case, the intersection manager rejects the request from the red CAV and gives the passing permission to the blue CAV, based on the First Come First Serve (FCFS) policy. Thus, in Figure 4.5(b), the blue CAV passes the intersection without stop while the red CAV stops before the intersection and requests permission repeatedly. In Figure 4.5(c), after the blue CAV successfully passes the intersection, the intersection manager gives permission to the red CAV and rejects the request sent from the yellow CAV. Finally, all three CAVs (red, blue, and yellow) passing the intersection smoothly and efficiently, as shown in Figure 4.5(d).

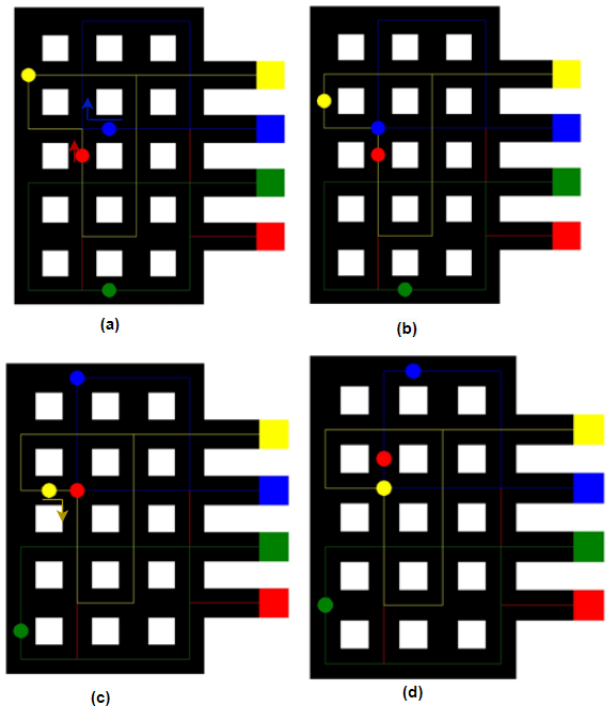


Figure 4. 5 A typical example for signal-free intersection management

After verifying the proposed signal-free reservation-based intersection management in the simulation environment, a miniature urban environment is built with the same configuration, shown in Figure 4.4(b). Also, four miniature CAVs are set up based on the Raspberry Pi 3.0 [52] system, and the system architecture for CAV model is shown in Figure 4.6, the proposed autonomous intersection management is served as the intersection management module in the network management part. The role of the module is to decide the passing sequences of miniature CAV at each intersection if there is potential conflict along respective routes. The module inputs are:

- Position (from link level positioning module)
- Heading direction (from link level positioning module)
- Traffic sign direction (from traffic sign recognition module)

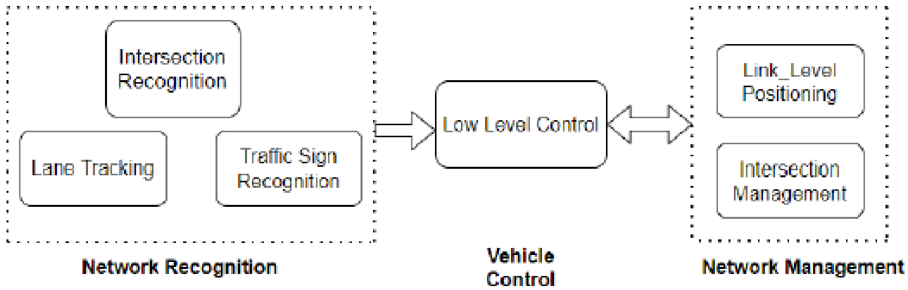


Figure 4. 6 Architecture for miniature CAV model

As for the result, similar to the numerical simulation, all CAVs can pass through every intersection smoothly and efficiently. In addition, the proposed autonomous intersection management strategy served as a key module in a cooperation traffic management system which represented the University of California, Riverside to participate in the JRC AUTOTRAC Competition [53] and rewarded 2nd place among all participate teams.

Chapter 5 Reservation–based Network Traffic Management with a Multi-agent System (MAS) Approach

5.1 Introduction

Due to the increase in population and traffic demand, traffic congestion has become a significant concern in urban areas. With the emergence of Connected and Automated Vehicles (CAVs) and Vehicle-to-infrastructure (V2I) communications, richer real-time information is becoming available, and higher definition control (at the individual vehicle level) can be applied to mitigating traffic congestion. This chapter proposes an innovative network traffic management (NTM) framework for CAVs using a multi-agent system (MAS) approach. Three agents are defined as network management agent, vehicle agents, and link agents. A link-level reservation-based strategy and optimal route searching algorithms are developed to route individual CAV traversing a given network to minimize its arrival time while balancing the flow rate of each link. A numerical example is presented to evaluate the system performance under different scenarios. The results show that the system can reduce travel time in the range of 8 - 12%, compared with the state-of-the-practice strategy. The system can also balance link utilization across the network, which is another key feature due to reservation.

5.1.1 Motivation

Rising traffic congestion is an inescapable condition in large and growing metropolitan areas across the world. Valuable time is wasted by vehicles stuck in traffic. A 2017 report from Texas A&M Transportation Institute [2] showed that the annual travel delay cost by congestion for each commuter in Los Angeles is 119 hours, and 47.9% of the travel delay (caused by congestion) occurred during peak hours on the freeway.

Fortunately, the emergence of Connected and Automated Vehicles (CAVs) has unlocked uncountable opportunities to improve transportation system management and vehicle operations in terms of safety, mobility, efficiency, and environmental sustainability [2]. From the network traffic management perspective, real-time exchange of vehicle information (e.g., origin, destination, and entry time) and link-level traffic conditions (such as time-dependent link volumes) enable more dedicated routing control at the individual vehicle level, which improves the utilization of network capacity for better mobility performance.

5.1.2 Background Information

Traffic network management is another research topic in this thesis. With the growing population, the need for transportation facilities is increasing day by day. As the available land area for new roadway infrastructure has become limited, it is needed proper transportation planning so that limited resources can be efficiently utilized. However, the heart of transportation planning is the prediction of future travel demand. With the emergence of CAVs and V2X communication, richer real-time information is becoming available, and higher definition control (at the individual vehicle level) can be applied to solve traffic network management problems. The selection of routes (alternative called paths) between origins and destinations in a transportation network is more concerned in the traffic network management problem or route assignment problem. The number of travelers on each route or link of the network (a route is a chain of links between an origin and destination) needs to be identified to determine the optimal route. Here are the basic notations in a general network traffic management problem:

G: network topology.

i: subscript for origin node, $i \in I$;

j: subscript for destination node, $j \in J$;

n: node in the network, $n \in N$;

a: subscript for a link in the network, $a \in A$;

k: subscript for a path in the network, $k \in k_{ij}, for\ i \in I\ and\ j \in J$;

τ: departure time interval, $\tau = 1, \dots, T$;

t: current time interval, $t = 1, \dots, T$;

Δ: length of a time interval

a_n^+: subscript for links leaving from node $n, a_n^+ \in A$;

a_n^-: subscript for links entering to node n, $a_n^- \in A$;

$\delta_{ijk}^{\tau ta}$: time-dependent link-path indicator, equal to 1 if vehicles going from i to j assigned to path k at time τ are on link a in period t;

T_{ijk}^{τ}: time-dependent path travel time for vehicles going from i to j that are assigned to path k at time τ;

$d_{ijk}^{\tau ta}$: number of vehicles going from i to j assigned to path k in period τ that are on link a at time t

$m_{ijk}^{\tau ta}$: number of vehicles going from i to j assigned to path k in period τ which exit link a in period t

r_{ij}^{τ}: number of vehicles departure from i to j in period τ:

r_{ijk}^{τ}: number of vehicles departure from i to j in period τ assigned to path k;

x^{ta}: number of vehicles on link a at the beginning of period t;

d^{ta}: number of vehicles which enter link a in period t;

m^{ta}: number of vehicles which exiting link a in period t;

I_n^t: number of vehicles generated at node n in period t;

O_n^t: number of vehicles exiting the network through node n in period t;

5.2 Problem Description and System Architecture

In this chapter, a multi-agent-based network management system is proposed. Given a connectivity-enabled roadway network, respective origin/destination, and entry time of each CAV, the goal is to find the optimal route and release time (into the network) for each vehicle to minimize its arrival time while balancing the link-level utilization (i.e., avoiding oversaturation along with a certain link) across the entire network. This section describes how formulate the problem in a multi-agent system (MAS) framework.

5.2.1 Problem Description

To deploy the proposed MAS, a directed network is denoted by $G = (N, A, W)$, where $N = \{n_1, n_2, \dots, n_i\}$ is the set of nodes, $A \subseteq N \times N$ is the set of directed arcs or links, and W is the weight matrix representing the travel time on each link. Two subsets are defined as the origin/destination node sets, I and J. All vehicles in this system must enter and exit the network from an origin/destination node belong to I and J. Note that the travel time on each link depends on the time-dependent link volume. Therefore, the total travel time for the route r is the sum of the weight on each visited link. In general, vehicles can utilize every link in the network but can only use the same link once.

5.2.2 System Architecture

This section proposes a network traffic management system for CAVs using a multi-agent system (MAS), where vehicles and road segments are defined as individual agents. Then, the behaviors of different agents are coordinated to effectively manage the entire system. The block diagram in Figure 5.1 outlines the function for each type of agent and the relationship between different types of agents. More specifically, the proposed network traffic management system consists of three types of agents: vehicle agent (VA), link agent (LA), and network management agent (NMA). VA is responsible for calculating its route based on the information/guidance from NMA. Each road segment in the network has an associated LA to record the traffic conditions (e.g., the number of vehicles on the segment, estimated travel time) and to keep a reservation table for those vehicles to enter it in the future. The network management agent acts as a coordinator between VA and LA, whose main functions are: 1) to receive information from VA and determine the set of links (i.e., network partition) that should provide information for routing support based on VA's origin and destination; 2) to request traffic information and reservation table from LAs of interest and send them to target VA, and 3) to receive VA's travel route request and provide permission or guidance on the route request. Section 5.3 presents the description of each agent's function in more detail.

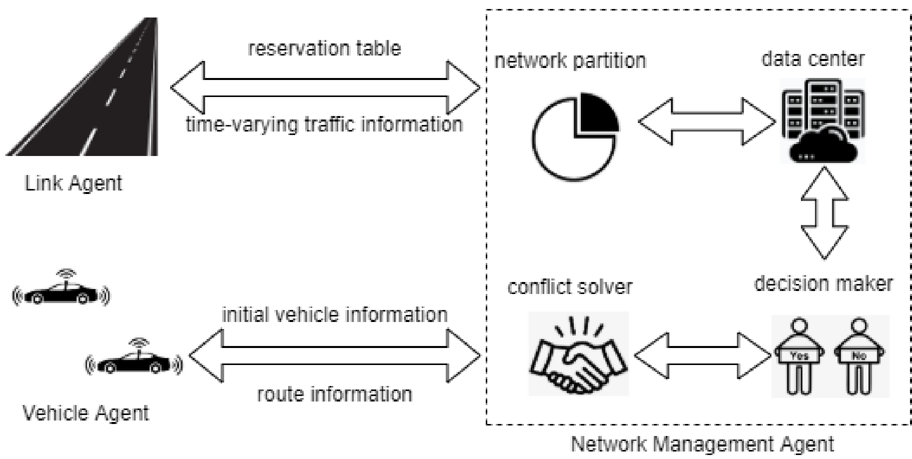


Figure 5. 1 Network management multi-agent system architecture.

The sequence diagram of the proposed system is shown in Figure 5.2. There are seven steps to determine a route for each CAV entering the network:

1. Once a VA enters the network (or its previous request gets rejected), it sends its origin/destination and entry time information to the NMA.
2. The NMA partitions the network and requests time dependent traffic information from involved LAs.
3. LAs send the link information (including estimated future travel time based on the

reservation) to the NMA upon request, and the NMA forwards this to the target VA. In the meantime, the NMA may request VA to provide more than one route as candidates based on the entry traffic conditions. If only one VA enters the partitioned network at a particular time step, then VA(s) may only provide one route (e.g., with earliest arrival time). If multiple vehicles enter at the same time step, then VAs need to calculate the first *k* shortest routes in terms of travel time.

4. Based on the information provided by the NMA, VA applies a *k-th* shortest path algorithm to generate the desired route/route set and sends the reservation request to the NMA for approval. Note that the request will include each involved link index and the respective entry/exit time.
5. If one or multiple non-conflicting route(s) for each involved VA can be identified, the NMA will select the set of routes with minimal total delay (for all involved VAs). Otherwise, the NMA will make decision(s) to reject some VAs' requests.
6. Based on the decision, the NMA notifies the involved LAs and VAs
7. LAs update their reservation lists, while the VA starts its trip by following the reservation or request again in the next time step.

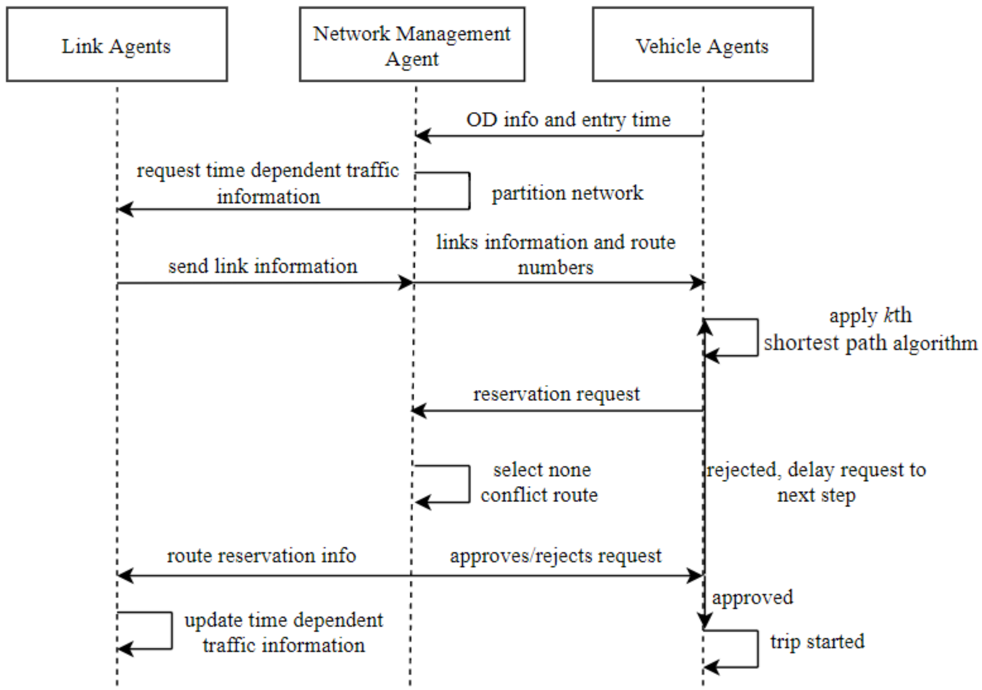


Figure 5. 2 Sequential diagram for the proposed MAS

5.2.3 Assumptions

To demonstrate the properties of the proposed MAS, several assumptions are made in the management mechanism:

- All VAs are CAVs with perfect communication and control performance.
- VAs can only start their trips with approvals by the NMA.
- VAs can and must follow the reservation (in terms of links and entry/exit time) exactly.
- Conflicts of VAs from different directions at the intersection (if any) are not considered in this study

5.3 Description of Different Agents

This section describes the functionality of each type of agent in the proposed system.

5.3.1 Link Agent

Primary functions of a link agent (LA) include:

1. Monitoring its prevailing traffic condition.
2. Updating its reservation list.
3. Providing its travel time profile (over time) based on the up-to-date reservations.

An example of the reservation list and profile for a LA is shown in Table 5.1 and Figure 5.3, representing time-dependent link volume due to reservation. In this example, the free flow travel time of this link i is set to 5 seconds. Vehicle A reserves the link from 0 – 5s; vehicle B and C both reserves the link from 2 – 7s; vehicle D reserves the link from 6s to 11s. Every LA in the network keeps a reservation profile similar to the one shown in Table 5-1. The reservation profile keeps updating as new requests come in. In addition, since the travel speed on each link is dependent on the time-dependent traffic volume, we convert the reservation list to the time-dependent link volume based on the reservation information, as shown in Figure 5.3.

Table 5. 1 Example of Link Reservation List

Vehicle ID	Reserve period (s)
A	0 - 5
B	2 - 7
C	2 - 7
D	6 - 11
…	…

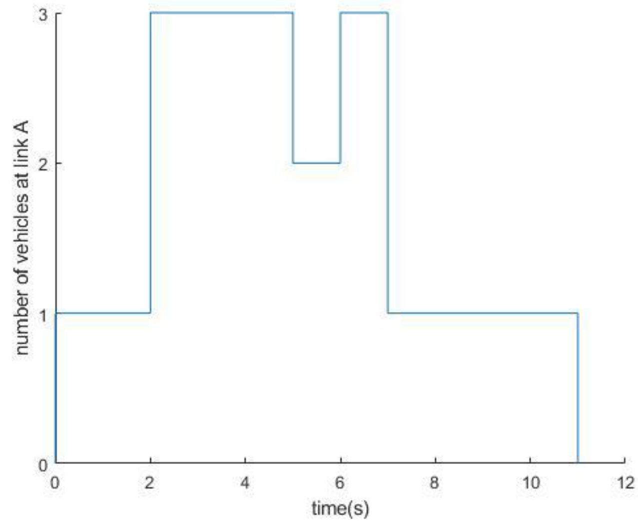


Figure 5. 3 Time dependent link volume based on reservation

Another information in the reservation profile is the link weight (or travel time) at each time step. In the proposed system, a weight in W is a time dependent variable and determined by the traffic volume. To obtain the weight or travel time, a triangular fundamental diagram [47] is applied, as shown in Figure 5.4, which represents the relationship between density and traffic flow for the i-th LA. More specifically, the link capacity q_i^{cap}, link length L_i, critical density k_i^{cap}, jam density k_i^{jam}, and link free flow speed v_i^f are predefined for all the links in the network. To estimate the link travel time of a VA entering the i-th LA at time t_0, Algorithm 3 is proposed, considering the speed fluctuations due to the change of downstream (with respect to the VA) traffic density. At time t, VA's instantaneous travel speed is determined by the downstream traffic density k_i^t and calculated in Eq. (5.1).

$$v_i^t = \begin{cases} v_i^f & if\ k_i^t \leq k_i^{cap} \\ \frac{q_i^{cap} * (k_i^{jam} - k_i^t)}{(k_i^{cap} - k_i^{jam}) * k_i^t} & if\ k_i^t > k_i^{cap} \end{cases} \tag{5.1}$$

Algorithm 3: Time Dependent Weight (Travel Time) Calculating

```
Initialize link parameters, W, L_t (traveled distance)
Input t_0 (time entered link i)
 v_i^{t_0} = equation (1)
t = t_0
while L_t < L_i:
        v_i^t = equation (1)
        if v_i^t < v_i^{t_0}:
                L_t += v_i^{t_0} * time step solution
        else:
                L_t += v_i^t * time step solution
        t += time step solution
        W += time step solution
return W
```

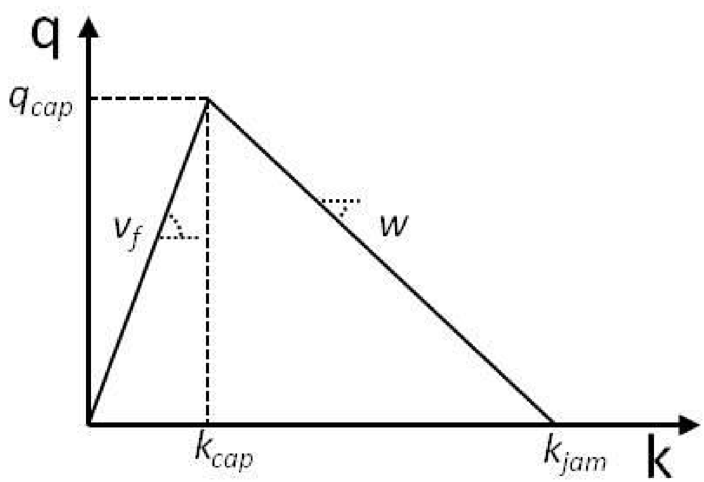


Figure 5. 4 Fundamental diagram for link inside the network

5.3.2 Network Management Agent

The network management agent (NMA) has three main functions. The first function is to serve as the data/information exchange media for the entire system. Both LAs and VAs are communicated via NMA, as shown in Figure 5.1. The second function is to partition the network upon receiving (multiple) VAs' requests (along with their origins and destinations). This function mitigates potential conflicts in routing among VAs and helps reduce the computational load of each VA. In addition, network partition may enhance the system's scalability. The third function is to solve reservation conflicts for VAs that enter the network simultaneously. In this study, reservation conflict may only occur when multiple CAVs enter the network at the same time and they both request to reserve the same link in the same time window. Figure5.5 shows the flow chart of the NMA's decision-making process in the proposed multi-agent system.

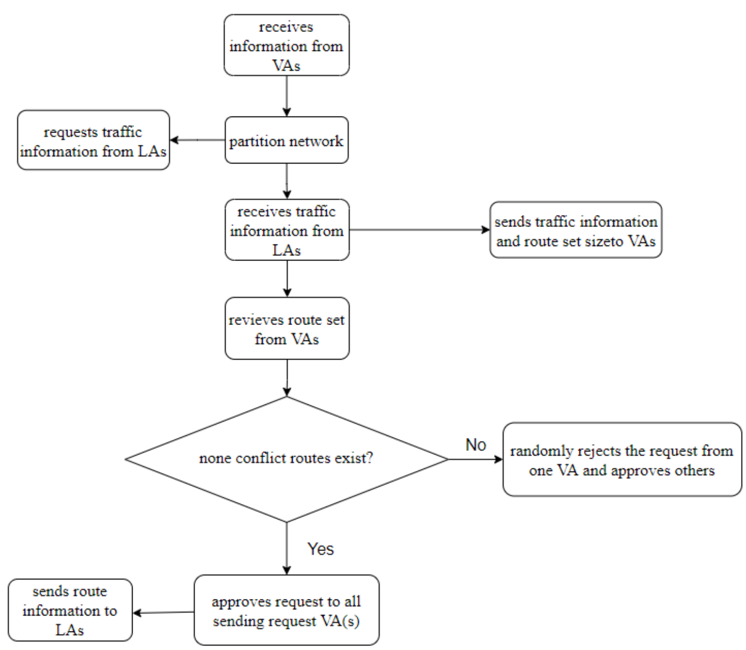


Figure 5. 5 Network management agent flow chart

5.3.3 Link Agent

The main function of a VA is to calculate the optimal route based on different objective functions and the up-to-date LA reservation profiles from the NMA. A Depth First Search (DFS)-based optimal route searching algorithm is developed for each VA to find the optimal route. The basic idea is to find all the possible routes in the partitioned map given by the NMA and sort them out by travel time. Because this algorithm (see Algorithm 4 below) explores each branch as far as possible before backtracking. Therefore, travel time on each link can calculated by checking its reservation information in a forward manner.

Algorithm 4: DFS-based k-th Optimal Route Searching

```
Initialized visited record
Function DFS (u (start node), d (end node), t (time entry network), K)
if u == d:
        APPEND cur_path to path_list
        SORT path_list by weight_calcution (t, cur_path)
        RESIZE path_list to K
        Return path_list
SET u as visited
for v in adjacency point of u:
        APPEND v to cur_path
        CALL DFS
        Return u visited
```

Figure 5.6 shows the flow chart of a VA in the proposed multi-agent system. Once a VA enters the network, it sends its origin/ destination information and entry time to the NMA. The NMA responds to the VA with the set of LAs and their reservation lists as well as a value of K. Then, VA applies the DFS k-th Optimal Route Selection Algorithm to find the optimal route/route set, depending on the value of K. Next, VA sends a request to the NMA for initiating the trip with the candidate route/route set. If the NMA approves the request with the suggested route, VA will start the trip by following the instruction. Otherwise, VA needs to wait at the start node till next time step and resend the request as well as re-calculate the optimal route/route set.

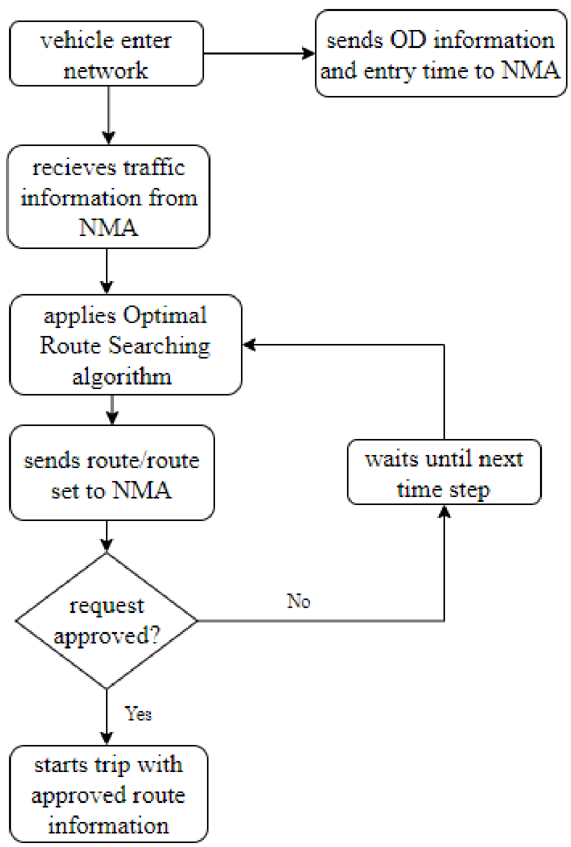


Figure 5. 6 Vehicle agent flow chart

. 5.4 Simulation and Results

5.4.1 Simulation Setup

To evaluate the performance of the proposed network management system, a 3 by 3 network (Figure 5.7) along with different traffic conditions and management strategies have been coded in python on an agent basis microscopic numerical environment.

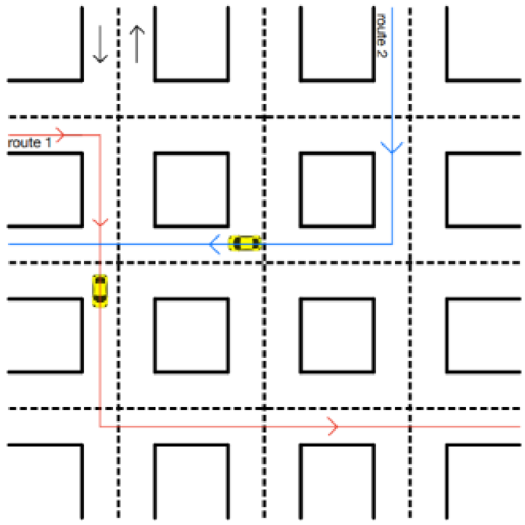


Figure 5. 7 Network illustration

The simulation setup in this paper is described as follows:

- Simulation time step solution: 1s
- Link length L: 500m
- Free flow speed V_f: 17.9 m/s (40 mph)
- Link capacity q_{cap}: 2000 veh/hr/ln
- Critical density k_{cap}: 0.03 veh/m (50 veh/mi/ln)
- Jam density k_{jam}: 0.14 veh/m (229 veh/mi/ln)

Note, for simplicity, we set the same parameters for all links inside the network.

5.4.2 Management Strategies

Three types of management strategies are deployed in this paper, whose goals are all to minimize the arrival time.

1. Optimal route using traffic condition upon entry (ORE)

   In this strategy, each VA can only access traffic conditions at the time when it enters the network. Then, VA calculates the optimal route with the earliest arrival time based on current traffic conditions. This strategy is considered as the baseline since it is similar to the state of the practice.

2. Optimal route using reservation information (ORT)

   The second strategy is to find the optimal route with the earliest arrival time, based on LAs' reservation information (Figure 5.3) rather than LAs' current traffic conditions. With LA's reservation information, a VA can determine its route based on future traffic conditions. Note that both ORE and ORT strategies require VAs to start the trip right away upon arriving at the network.

3. Earliest arrival with reservation information and flexible departure time window (ORTD)

   The last strategy also considers LAs' reservation information, but it allows a VA to not only make a reservation on its route but also specify its departure time to achieve its earliest arrival (i.e., the VA may postpone its leaving time at the origin node). To find the route with the earliest arrival time for VA, a candidate departure time window $[\mathrm{t_0}, \mathrm{t_0} + \mathrm{T}]$ is casted:

$$\mathrm{T} = \mathrm{t_{ORT}}(\mathrm{t_0}) - \mathrm{t_f} \tag{5.2}$$

where $\mathrm{t_0}$ is the current time; $\mathrm{t_{ORT}}(\mathrm{t_0})$ is the VA's earliest arrival time if it starts the trip at $\mathrm{t_0}$ with the reservation information up to $\mathrm{t_0}$; $\mathrm{t_f}$ is VA's earliest arrival time if the vehicle travels through the network at free flow speed $\mathrm{V_f}$. It turns out that the best departure time (with the earliest arrival) needs to fall within the time window specified in Eq. (5.2). Otherwise, the VA should start the trip at time $\mathrm{t_0}$. Please note that the calculation and decision are made at time $\mathrm{t_0}$.

5.4.3 Route Set Size (k)

As aforementioned, to mitigate potential route conflict(s), each VA will provide k-th shortest routes to the NMA for decision making. In this section, the impact of the route set size k on the network mobility performance is examined. The k's value is varied from 1 to 5 and the average trip delay (compared with free-flow speed) is calculated is the measurement. Table 5.2 shows the results from the ORT strategy, where each value represents 10 simulation runs with different seeds. As illustrated in the table, the average trip delay decreases as the value of k increases from 1 to 3. However, no further degradation in mobility is observed when the value of k keeps increasing till 5 in this study case. It is also expected that the computational efforts increase as the value of k grows. Therefore, the value of route set size k is set to be 3 in following analyses.

Table 5. 2 Sample Landscape Table

Value of k	Average Trip Delay(s)
1	3.01
2	2.98
3	2.95
4	2.95
5	2.95

5.4.4 Mobility Analysis

The system mobility performance is evaluated in this section by calculating the average trip delay (compared with the free flow speed). Totally 1000 vehicles are spawned into the network with two inflow rates, representing two different traffic conditions. For a lighter traffic condition, 2 vehicles are generated per time step; for a heavier traffic condition, 3 vehicles are generated per time step. For each VA, the origin and destination are generated randomly. Figure 5.8 shows the average link volume/maximum link volume under light/ heavy traffic conditions over time by applying ORT, where the dotted line represents the link capacity. As shown in Figure 5.8, under the light traffic condition, the maximum link volume across the network does not exceed the link capacity for most of the time, while congestion (when the maximum link volume is greater than the capacity) occurs frequently under the heavy traffic condition.

Table 5. 3 Average Trip Delay under Different Strategies and Traffic Conditions

Traffic condition	Management Strategy	Average Trip Delay(s)	Improvement (%)
Light	ORE	0.23	-
	ORT	0.21	8.33%
	ORTD	0.20	13.02%
Heavy	ORE	3.45	-
	ORT	3.04	11.67%
	ORTD	2.92	15.31%

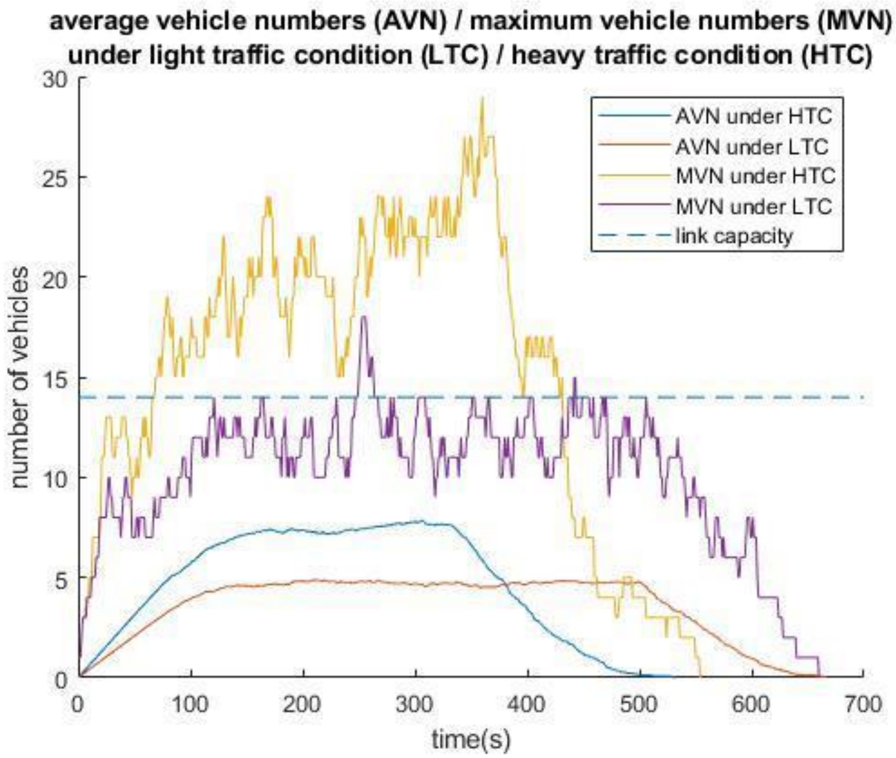


Figure 5. 8 Key statistics over time for different traffic conditions

Results of average trip delays are shown in Table 5.3, and 10 simulations run with different random seeds for each scenario. Under the light traffic condition, most vehicles travel at the free flow speed and the average trip delay is around 0.2s. Compared with ORE (baseline), average trip delays for ORT and ORTD are reduced by 8.33% and 13.02%, respectively. Under the heavy traffic condition, links in the network become more congested and the average trip delay with ORE strategy is 3.45s. The system performs significantly better with the ORT and ORTD strategy, where the average trip delays are shortened by 11.67% and 15.31%, respectively.

5.4.5 Network Usage Rate Analysis

This section analyzes the network usage rate under different management strategies and check if reservation-based strategies (i.e., ORT and ORTD) would help balance link utilization across the network. The results are analyzed under the heavy traffic condition. Figure 5.9 presents the standard deviation of the volumes of all links in the network over time with different management strategies. It can be observed that vehicles under the ORE and ORTD strategies can utilize the entire network resources more efficiently, compared with the ORE strategy. This also explains the improvement in system mobility with the proposed strategies.

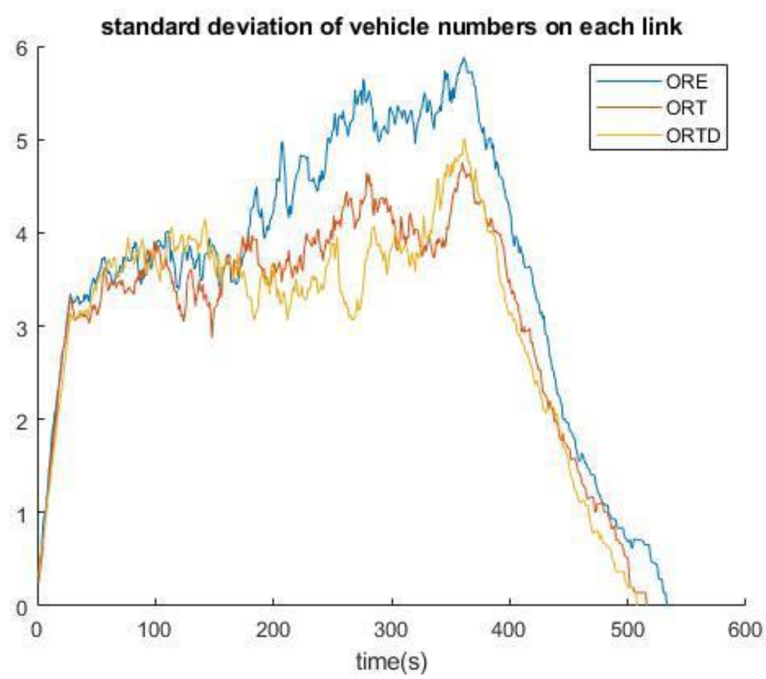


Figure 5. 9 Standard deviation of vehicle number on each links over time by applying different management strategies

Chapter 6 Conclusions and Future Work

6.1 Conclusions

This thesis was focused on the area of the transportation system and applied the latest advances in the field of Intelligent Transportation Systems (ITS) to the mitigation of the environmental and energy issues associated with the movement of goods and people. The motivation of this thesis is the increasing traffic congestion and the emerging autonomous vehicles (AVs), vehicle-to-everything (V2X) communication, and advanced machine learning algorithms. Traffic management problems were studied in the thesis, which can be divided into intersection management and network management problems.

As for the intersection management problems, this thesis first studied the signalized intersection management problem using deep reinforcement learning, where vehicles' position and speed are processed by a convolution neural network (CNN) and fed into the reinforcement learning system as inputs. And a Deep Q-Network (DQN) was introduced to estimate the Q value. The most innovative point in this study was the 65 states flexible traffic light state dual-ring controller. It can maximize the flexibility at the intersection. In addition, four different types of reward functions were designed and compared with traditional TLC methods, including fixed time and actuated control, under two different traffic demands. As for the experiment, the real-world intersection was imported in SUMO and generated the traffic demand based on the real-world conditions. The results showed that proposed DRL model over-performance traditional traffic signal control method in terms of average travel delay, emission ($CO, CO_2,\ NO_X$) and fuel consumption by 23.8%, 14.7%, 18.5%, 13.5%, 12.6%, perceptively. Among all the reward functions, the one using max queue length had the best performance mainly due to this reward function designed to better balance the number of vehicles from different approaches. Then, the thesis moved from signalized intersection management to a signal-free intersection management problem and proposed a centralized reservation-based management strategy to decide the passing sequence of CAVs at each intersection if there are potential conflicts along the respective routes, which guarantees the safety at intersection areas and improves the efficiency. To validate the strategy, the proposed strategy was first tested in a numerical network environment coded by python and then tested in a miniature urban scenario in the real world. The result showed that the proposed signal-free intersection management based on a first come, first served (FCFS) policy can not only guaranteed safety at intersection areas and improved travel efficiency.

As for the network traffic management problem, a scalable multi-agent system (MAS) framework had been proposed for network traffic management, where the network management agent (NMA), link agents (LAs), and vehicle agents (VAs) would cooperate with each other, and their functionalities were clearly defined in a scalable manner. Also, a reservation-based strategy had been developed to help the system find and secure the route for each vehicle agent (VA) based on the up-to-date network traffic conditions. A simulation study illustrated promising results of the proposed framework and strategy in improving system mobility and balancing link usage across the entire network. The results showed that the proposed system could reduce average travel delay in the range of 8 - 12%, compared with the state-of-the-practice strategy.

6.2 Future Directions

This section lists possible future directions for three topics studied in this thesis. As for the signalized traffic signal control using reinforcement learning, the future directions are:

- To apply propose DRL model to network scenarios to control multiple intersections.
- To further improve the reward functions.
- To change the DQN Architecture.

As for signal-free intersection management, one of the future works can be to obtain numerical results and to compare with other signal-free intersection management strategies, e.g., all-way stop.

As for the reservation-based network traffic management with a multi-agent system (MAS) approach, the future directions are:

- To compare proposed strategies with traditional network management methods (UE and SO)
- To improve the management strategy to deal with the situation that vehicle agents cannot follow the reserved time slots correctly
- To test proposed the management strategy in a real-world network environment.

Reference

[1]. https://www.cert.ucr.edu/transportation-systems-vehicle-infrastructure-interaction

[2]. 2017 Urban mobility report by Texas A&M Transportation Institute.

[3]. Zhang, K., & Batterman, S. (2013). Air pollution and health risks due to vehicle traffic. The Science of the total environment, 450-451, 307-316.

[4]. Society of Automotive Engineers (SAE): https://www.sae.org/automated-unmanned-vehicles/

[5]. Mahmood, Adnan & Zhang, Wei Emma & Sheng, Quan. (2019). Software-Defined Heterogeneous Vehicular Networking: The Architectural Design and Open Challenges. Future Internet. 11. 70. 10.3390/fi11030070.

[6]. Haghighat, A.K., Ravichandra-Mouli, V., Chakraborty, P. et al. Applications of Deep Learning in Intelligent Transportation Systems. J. Big Data Anal. Transp. 2, 115–145 (2020). https://doi.org/10.1007/s42421-020-00020-1

[7]. Choi, Eun-Ha. Crash Factors in Intersection-related Crashes: An Onscene Perspective. NO. HS-811 366. 2010.

[8]. C. Tang, X. Wei, M. Hao, C. Zhu, R. Wang and W. Chen, "Traffic Signal Phase Scheduling Based on Device-to-Device Communication," in *IEEE Access*, vol. 6, pp. 47636-47645, 2018, doi: 10.1109/ACCESS.2018.2867553.

[9]. Peter Koonce et al. 2008. Traffic signal timing manual. Technical Report. United States. Federal Highway Administration.

[10]. Roger P Roess, Elena S Prassas, and William R McShane. 2004. Traffic Engineering. Pearson.

[11]. John DC Little, Mark D Kelson, and Nathan H Gartner. 1981. MAXBAND: A Versatile Program for Setting Signals on Arteries and Triangular Networks. (1981).

[12]. P. M. Murray-tuite, "A Comparison of Transportation Network Resilience under Simulated System Optimum and User Equilibrium Conditions," *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 1398-1405, doi: 10.1109/WSC.2006.323240.

[13]. Beckmann, M.J., McGuire, C.B. and Winsten, C.B. Studies in the Economics of Transportation, Yale University Press, New Haven, Connecticut; published earlier as Rand-RM 1488, May12, 1955

[14]. Sheffi, Y. 1985. Urban transportation networks: Equilibrium analysis with mathematical programming methods. New Jersey: Prentice-Hall, Inc. pp. 55-60

[15]. https://en.wikipedia.org/wiki/Bellman_equation

[16]. Qing He, K. Larry Head, Jun Ding, Multi-modal Traffic Signal Control with Priority, Signal Actuation and Coordination, Transportation Research Part C: Emerging Technologies, Volume 46, 2014, Pages 65-82, ISSN 0968-090X, https://doi.org/10.1016/j.trc.2014.05.001.

[17]. K. Pandit, D. Ghosal, H. M. Zhang and C. Chuah, "Adaptive Traffic Signal Control with Vehicular Ad hoc Networks," in *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459-1471, May 2013, doi: 10.1109/TVT.2013.2241460.

[18]. Mohammad Khayatian, Mohammadreza Mehrabian, Aviral Shrivastava, "RIM: Robust Intersection Management for Connected Autonomous Vehicles", Real-Time Systems Symposium (RTSS) 2018 IEEE, pp. 35-44, 2018.

[19]. Seung-Bae Cools, Carlos Gershenson, and Bart DâĂŹHooghe. 2013. Self-organizing traffic lights: A realistic simulation. In Advances in Applied Self-organizing Systems. Springer, 45–55.

[20]. Carlos Gershenson. 2004. Self-organizing Traffic Lights. arXiv preprint (2004).

[21]. Pravin Varaiya. 2013. The Max-pressure Controller for Arbitrary Networks of Signalized Intersections. In Advances in Dynamic Network Modeling in Complex Transportation Systems. Springer, 27–66.

[22]. Pravin Varaiya, Max Pressure Control of a Network of Signalized Intersections, Transportation Research Part C: Emerging Technologies, Volume 36, 2013, Pages 177-195, ISSN 0968-090X, https://doi.org/10.1016/j.trc.2013.08.014.

[23]. K. Sjoberg, P. Andres, T. Buburuzan, and A. Brakemeier, “Cooperative Intelligent Transport Systems in Europe: Current Deployment Status and Outlook,” IEEE Veh. Technol. Mag., vol. 12, no. 2, pp. 89–97, 2017.

[24]. K. Dresner and P. Stone, “A multiagent approach to autonomous intersection management,” *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, 2008.

[25]. L. Li and F. Y. Wang, "Cooperative Driving at Blind Crossings using Intervehicle Communication," IEEE Trans. Veh. Technol., vol. 55, no. 6, pp. 1712–1724, 2006.

[26]. J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1, pp. 81–90,2012.

[27]. M. A. S. Kamal, J. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 3, pp. 1136–1147, 2015.

[28]. Feng Zhu, Satish V. Ukkusuri, A Linear Programming Formulation for Autonomous Intersection Control within a Dynamic Traffic Assignment and Connected Vehicle Environment, Transportation Research Part C: Emerging Technologies, Volume 55, 2015, Pages 363-378, ISSN 0968-090X, https://doi.org/10.1016/j.trc.2015.01.006.

[29]. T. L. Thorpe, "Vehicle Traffic Light Control using SARAS," in Online]. Available: citeseer. ist. psu. edu/thorpe97vehicle. html. Citeseer, 1997.

[30]. B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," Journal of Transportation Engineering, vol. 129, no. 3, pp. 278–285, 2003.

[31]. L. Li, Y. Lv, and F.-Y. Wang, "Traffic Signal Timing via Deep Reinforcement Learning," IEEE/CAA Journal of Automatica Sinica, vol. 3, no. 3, pp. 247–254, 2016.

[32]. S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," IET Intelligent Transport Systems, vol. 11, no. 7, pp. 417–423, 2017.

[33]. E. Van der Pol and F. A. Oliehoek, "Coordinated Deep Reinforcement Learners for Traffic Light Control," Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016), 2016.

[34]. M. Aslani, S. Seipel, M. S. Mesgari, and M. Wiering, "Traffic Signal Optimization Through Discrete and Continuous Reinforcement Learning with Robustness Analysis in Downtown Tehran," Advanced Engineering Informatics, vol. 38, pp. 639–655, 2018.

[35]. Y. Gong, M. Abdel-Aty, Q. Cai, and M. S. Rahman, "Decentralized Network Level Adaptive Signal Control by Multi-agent Deep Reinforcement Learning," Transportation Research Interdisciplinary Perspectives, vol. 1,
p. 100020, 2019.

[36]. X. Wang, L. Ke, Z. Qiao, and X. Chai, “Large-scale Traffic Signal Control using a Novel Multiagent Reinforcement Learning,” IEEE transactions on cybernetics, vol. 51, no. 1, pp. 174–187, 2020.

[37]. H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, “Cooperative Deep Q-learning with Q-value Transfer for Multi-intersection Signal Control,” IEEE Access, vol. 7, pp. 40 797–40 809, 2019.

[38]. H. S. Mahmassani and G.-L. Chang, “On Boundedly Rational User Equilibrium in Transportation Systems,” Transportation Science, vol. 21, no. 2, pp. 89–99, 1987.

[39]. M. G. H. Bell and C. Cassir, “Risk-averse User Equilibrium Traffic Assignment: an Application of Game Theory,” Transportation Research Part B: Methodological, vol. 36, no. 8, pp. 671–681, 2002

[40]. J. de Cea and E. Fernández, “Transit Assignment for Congested Public Transport Systems: An Equilibrium Model,” Transportation Science, vol. 27, no. 2, pp. 133–147, 1993.

[41]. H.-J. Huang and W. H. K. Lam, “Modeling and Solving the Dynamic User Equilibrium Route and Departure Time Choice Problem in Network with Queues,” Transportation Research Part B: Methodological, vol. 36, no. 3, pp. 253–273, 2002.

[42]. S. Wang, S. Djahel, and J. McManis, “A Multi-agent-based Vehicles Re-routing System for Unexpected Traffic Congestion Avoidance,” 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014.

[43]. I. H. Zohdy, R. K. Kamalanathsharma, and H. Rakha, “Intersection Management for Autonomous Vehicles Using iCACC,” 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012.

[44]. Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, “Advanced Intersection Management for Connected Vehicles using a Multi-agent systems Approach,” 2012 IEEE Intelligent Vehicles Symposium, 2012.

[45]. Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, “Multi-agent Intersection Management for Connected Vehicles using an Optimal Scheduling Approach,” 2012 International Conference on Connected Vehicles and Expo (ICCVE), 2012.

[46]. Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, “Platoon-based Multi-agent Intersection Management for Connected Vehicle,” 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013.

[47]. C.F. Daganzo and N. Geroliminis, “An analytical approximation for the

macroscopic fundamental diagram of urban traffic," Transportation Research Part B: Methodological, vol. 42, no. 9, pp. 771–781, 2008

[48]. Krajzewicz D, Erdmann J, Behrisch M, et al. Recent development and applications of SUMO-Simulation of Urban MObility[J]. International journal on advances in systems and measurements, 2012, 5(3&4).

[49]. D. Kari, G. Wu and M. J. Barth, "Development of an agent-based online adaptive signal control strategy using connected vehicle technology," 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 1802-1807, doi: 10.1109/ITSC.2014.6957954.

[50]. Wei H, Zheng G, Gayah V, et al. A survey on traffic signal control methods[J]. arXiv preprint arXiv:1904.08117, 2019.

[51]. https://www.openstreetmap.org/#map=15/42.5378/-96.3833

[52]. https://www.raspberrypi.org/]

[53]. https://ec.europa.eu/jrc/en/event/other-event/autotrac-2020-final-event

[54]. https://www.dot.nv.gov/mobility/avcv

[55]. July 5, 2018 / Deep learning, Machine learning, Reinforcement learning, Popular post / by Błażej Osiński and Konrad Budek

[56]. https://www.synopsys.com/automotive/autonomous-driving-levels.html