The Hong Kong Polytechnic University

Department of Computing

Adaptive Traffic Light Control in Wireless Sensor Network-based Intelligent Transportation System

by

ZHOU BINBIN

A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Philosophy

December 2010

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________	(Signature)
Binbin Zhou	(Name of Student)

Abstract

Intelligent Transportation System (ITS) refers to a system that integrates advanced communications, computing and electronics technologies into transportation infrastructure and vehicles, to improve safety and efficiency and to reduce traveling time and fuel consumption. The conventional surveillance methods used in ITS to detect real-time traffic data, e.g. video image processing and inductive loops detection, have several shortcomings, such as limited coverage and high costs of implementation and maintenance. Wireless sensor networks (WSNs) offer the potential of providng real-time traffic data without these drawbacks. Hence, in the past decade, WSNs have been applied to ITS to improve the performance of ITS.

Controlling traffic lights plays a key role in ITS. An optimal traffic light control approach can increase traffic throughput and reduce waiting time. In this thesis, we investigate how to design methods and algorithms for adaptive traffic light control in a WSN-based ITS. We review the related work on collecting real-time traffic data and on controlling traffic lights, including fixed-time control, actuated control, and adaptive control. We propose models and schemes for adaptive traffic light control for both isolated intersections and multiple intersections. The proposed approaches take advantage of real-time traffic information collected by WSNs to achieve high system throughput, low waiting time and few stops for the vehicles.

First, we describe an adaptive traffic light control scheme proposed for an isolated intersection. This scheme can adjust both the sequence and length of traffic lights in accordance with real-time traffic loads. It takes into consideration a number of factors such as traffic volume, waiting time, vehicle density, and others, to determine the sequence and the optimal length of green lights. Simulation results demonstrate that our approach results in much higher throughput and lower average waiting time as compared with the optimal fixed-time control approach and an actuated control approach.

We then propose an adaptive traffic light control scheme for multiple intersections. In this case, we need to also consider controlling the traffic lights for multiple adjacent intersections in a distributed way. Our proposed scheme can collect real-time traffic data, and adjust both the sequences and lengths of the green lights of intersections cooperatively. Real-time traffic data, e.g. traffic volumes, waiting time, number of stops, vehicle densities, are taken into account to determine the sequence of green lights in each intersection. The optimal lengths of the green lights can be calculated based on the information about local traffic volumes and the remaining green light durations of neighboring intersections. Simulation results likewise show that our scheme produces much higher throughput, lower average waiting time and fewer average stops, compared with the optimal fixed-time control approach, an actuated control approach, and an adaptive control approach.

We have implemented the proposed schemes on our testbed for Intelligent Services with Wireless Sensor Networks, iSensNet to evaluate and demonstrate the performance. Our experimental results show that our approaches can deal with different traffic conditions in an effective manner.

Keywords: wireless sensor network, intelligent transportation system, adaptive traffic light control, real-time traffic data.

Publication

Journal Paper

1. **Binbin Zhou**, Jiannong Cao, and Hejun Wu, “Adaptive Traffic Light Control in WSN-based ITS”, Submitted to *IEEE Transaction on Intelligent Transportation System*.

Conference Papers

1. **Binbin Zhou**, Jiannong Cao, Xiaoqin Zeng, and Hejun Wu, “Adaptive Traffic Light Control in Wireless Sensor Network-based Intelligent Transportation System”, *IEEE 72nd Vehicular Technology Conference (VTC2010-Fall), Ottawa, Canada*, September 2010.

2. **Binbin Zhou**, Jiannong Cao, and Hejun Wu, “Adaptive Traffic Light Control of Multiple Intersections in WSN-based ITS”, *IEEE 73rd Vehicular Technology Conference (VTC2011-Spring), Budapest, Hungary*, May 2011.

Book Chapter

1. Bai Li, Longxiang Gao, and **Binbin Zhou**, “The Simulation on Wireless Sensor Networks” chapter in book “Wireless Sensor Networks”, accepted by *Nova Science Publishers, Inc. New York*, January 2010.

Acknowledgement

First, I would like to thank my supervisor, Professor Jiannong Cao, for his rigorous supervision of my research. I thank him for his guidance, patience and encouragement during my M.Phil. study. He unremittingly trained me to be a good researcher. He taught me how to find research issues and how to solve problems. Time and again, he showed me how to express my ideas and write research papers. His knowledge, vision, passion, professional spirit and attitude towards the research deeply inspired me. Without his help and support, this body of work would not have been possible. What I have learned and experienced during the time I spent in his research group will always help and encourage me in the future.

Second, I would like to thank another excellent and distinguished person, Professor Xiaoqin Zeng. I would like to express my heartfelt gratitude to him for his insightful and illuminative suggestions on my research. His acuminous insight and guidance is invaluable to the accomplishment of this thesis.

Third, I thank Dr. Hejun Wu for his great help and wise advices during my M.Phil. study. He made a lot of contributions to this research work. Also, I would like to convey my thanks to Dr. Xuefeng Liu, Yang Liu, Jingjing Li, and Yi Lai for their kind help in my research and project works. Furthermore, I would like to thank all my teachers from whom I have learned so much during my long journey in academia. They are Dr. Wei Lou, Dr. Bin Xiao at the Hong Kong Polytechnic University, and many others. In addition, I also wish to thank Junchao Ma, Tao

Zhang, Xuan Liu, Jie Zhou, Miao Xiong, who have rendered their help to my research work in one way or another and shared with me the pleasure of study at the Hong Kong Polytechnic University.

Last but not least, I thank my parents for their continuous love, support, trust, and encouragement for me through the entire period of my study. Without them, none of this would have been possible.

Contents

List of Figures

List of Tables

List of Abbreviations

ITS	intelligent transportation system
WSN	wireless sensor network
ATMS	advanced traffic management systems
ATIS	advanced traveler information systems
APTS	advanced public transportation systems
ARTS	advanced rural transportation systems
AVCSS	advanced vehicle control and safety system
CVO	commercial vehicle operation
AADT	average annual daily traffic
VMT	vehicle miles traveled
RSSI	received signal strength indication
FTC	fixed-time traffic control
ATC	actuated traffic control
AL	approaching lane
LL	leaving lane
USN	upstream sensor node
ISN	intersection sensor node
LC	local controller
AFLC	adaptive fuzzy logic control
iSensNet	intelligent services with wireless sensor network
DSN	detection sensor node
VN	vehicle node
RN	roadside unit

Chapter 1

Introduction

Traffic congestion is a huge problem nowadays, due to the rapid increase in the demand for transportation and limited resources provided by traffic infrastructures [66] [86]. This results in longer vehicle travel time, increased energy consumption, growing environmental pollution, reduced traffic safety, and a decrease in the efficiency of the transportation infrastructure [66] [85]. Hence, controlling traffic has become a very important issue under a growing pressure to relieve traffic congestion. Traffic control is an important component of Intelligent Transportation System (ITS). ITS refers to a system that integrates advanced communications, information, and electronic technologies into transportation infrastructure and vehicles, to relieve traffic congestion, improve safety, and reduce transportation times and fuel consumption.

Controlling traffic lights plays a key role in increasing traffic throughput and reducing delay. When scheduling traffic lights, current traffic conditions should be considered as they can significantly affect the control scheme. Hence, collecting real-time traffic data is a very important issue. Conventional methods of controlling traffic lights have limitations. These include limited coverage due to sensors' fixed-location installations and the cable-based communication methods used to transmit the detected traffic information, which increases the cost of implementation and maintenance [22] [8]. Based on these drawbacks, it is necessary to search for another way to monitor traffic conditions. With the continuing development of Wireless Sensor Networks (WSNs), which use wireless sensor nodes for surveillance and communication, the possibility of overcoming these drawbacks is increasing. ITS can detect traffic information dynamically and then transfer traffic data through wireless technologies at a low cost. Thus, we choose to integrate WSN to ITS to obtain the real-time traffic data to design an algorithm to control traffic lights.

In this chapter, we first give an introduction to the intelligent transportation system, including different types of subsystems, the applications, and the technologies used. We then describe wireless sensor network technology, including the definitions, sensor nodes and applications. After this, we discuss the traffic light control, including the basic concepts, different types of control strategies, their objectives, and their shortcomings. We then point out the most challenging issues in the design of adaptive traffic light control approaches for a WSN-based ITS. At the end of this chapter, we summarize our contributions in this study and describe the organization of the thesis.

1.1 Intelligent Transportation System

As mentioned above, ITS applies advanced technologies on sensing, detecting, communication, and control into transportation systems for the purposes of improving transportation safety, decreasing transportation time, and reducing power consumption.

Generally, ITS can be classified into several subsystems according to their functions: Travel Management and Traveler Information Systems (TMTIS), Advanced Vehicle Control and Safety System (AVCSS), Commercial Vehicle Operation (CVO).

TMTIS is responsible for providing real-time traffic information, and for managing and controlling traffic conditions and traffic lights. It includes four subsystems: Advanced Traffic Management Systems (ATMS), Advanced Traveler Information Systems (ATIS), Advanced Public Transportation Systems (APTS), and Advanced Rural Transportation Systems (ARTS).

ATMS is responsible for detecting current traffic conditions, transmitting such traffic data to a control center by advanced communication technologies, and then designing traffic control schemes based on this information. ATMS already has some applications, such as traffic light control, incident management, electronic tolls, high occupancy vehicle control, and so forth [1]. ATIS is responsible for providing real-time traffic data to road users everywhere by advanced communication technologies, so that the users can make decisions about trips or routes based on this information. In addition, ATIS already has some applications, such as GPS, internet connections, message signs, and so on [6]. APTS is responsible for improving the service quality of public transportation. It applies the technology of ATMS and ATIS to public transportation to increase system efficiency. The applications include automatic vehicle monitoring, E-tickets, and so on [6]. ARTS is responsible for providing useful information to drivers by applying the technology of ATMS and ATIS in rural transportation, with the purpose of improving service quality and increasing efficiency.

AVCSS is responsible for providing useful information to road users to help them reduce the possibility of accidents and improve traffic safety through using advanced technologies. Applications of AVCSS include collision avoidance systems, automatic highway systems, driving assistance, and so forth [6].

CVO is responsible for applying the technologies of ATMS, ATIS and AVCSS to commercial vehicles such as trucks and taxes to improve their efficiency and traffic safety. Examples of applications are the automatic monitoring of vehicles and electronic payment [6].

Within the above systems, ATMS is the most important one for relieving traffic congestion. It applies advanced technologies for surveillance, real-time collection and then transmission of the traffic data to a management center by some advanced communication technologies. To alleviate congestion, the management center then designs traffic light control approaches based on all of the detected information.

Currently, there are some conventional surveillance methods to collect real-time traffic data, such as video image processing and inductive loops detection. Using these methods, traffic volumes, vehicle speeds, vehicle classifications, occupancies, and presence can be detected. However, these collection methods have some shortcomings, such as their limited coverage and high cost of implementation and maintenance [8]. Meanwhile, wireless sensor networks can offer the potential to provide real-time traffic data overcoming these drawbacks. Hence, we study how to apply WSNs to ITS to improve its performance in relieving traffic congestion.

1.2 Wireless Sensor Networks

With the development of technologies in wireless communication, digital electronics, and low power sensing, WSNs have emerged in the past decade as a promising technology for surveillance and data collection. WSNs are composed of spatially distributed small sensor nodes that use wireless communication to communicate among them when sensing the physical world. [12] [13] [20] [25] [68].

A sensor node consists of four components: sensors, a processor, a radio, and an energy source, so that it is able to sense the surrounding environment, perform some processing, and communicate with other connected nodes [2] [25]. Sensor nodes can detect conditions such as temperature, sound, pressure, motion, and so forth. The cost of sensor nodes is variable, ranging from hundreds of dollars to a few pennies, depending on the size of the network and the requirements of individual sensor nodes [48].

Because of the flexibility in deployment and various functions, WSN has numerous potential applications. These typically include environmental monitoring, industrial monitoring, machine health monitoring, and tracking or controlling [3] [72] [48] [80].

Under the guarantee that all of the traffic data in the whole network range can be measured, using sensor nodes can overcome the shortcomings mentioned above [22]. Therefore, we apply WSNs in ITS to collect real-time traffic data and enhance traffic safety.

In a WSN-based ITS, there are several types of communications, such as vehicle-to-vehicle, vehicle-to-infrastructure, or infrastructure-to-vehicle. Traffic conditions can be detected and the information transmitted to vehicles through vehicle-to-vehicle or infrastructure-to-vehicle communications [47].

1.3 Traffic Light Control

The term traffic light control, also called traffic signal control, refers to a strategy to schedule the traffic lights to ensure that traffic can move as smoothly and safely as possible. Different control strategies have different performances. The performance criteria include vehicle throughput, waiting time, and the number of vehicle stops.

An optimal control strategy can increase the utilization of infrastructure, improve traffic safety, and reduce energy consumption. In contrast, a suboptimal control strategy would result in considerable delay due to frequent changes of traffic lights and inaccuracies of scheduling. When designing the strategy to control traffic lights, a number of apparent difficulties should be considered [66]. They include the increasing size of the problem for a large traffic network, the limited coverage of traffic detection, and many unpredictable disturbances which are difficult to measure, such as traffic incidents and illegal parking.

The combination of these difficulties makes it harder to design a effective traffic light control strategy to achieve an optimal real-time schedule. This is especially the case in multiple intersections scenarios, since the coordination between adjacent intersections also should be taken into account. Traffic conditions from neighboring intersections would have a dynamic impact on the scheduling of local traffic lights since these traffic flows should become part of the local traffic flow after a certain period.

When designing a traffic light control strategy, there are several factors that should be kept in mind, cycle time, split, phase sequence and offset, as these factors may affect current traffic conditions.

Cycle time is the total duration of all the series of traffic light combinations at an intersection. A longer cycle time would increase the capacity of the intersection due to the decrement of the fraction of the constant lost time. On the other hand, a longer cycle time would increase vehicle delay in under-saturated intersections because the waiting time during the red phase would grow accordingly [66]. A phase, also called a stage, is part of a whole cycle of traffic light combinations, during which one set of traffic flows has the right of way [66]. Constant lost time means the necessary time between the phases to avoid interference between the antagonistic traffic flows of two conflicting phases.

Split is the proportion of the cycle time that is assigned as the green phase for a set of traffic movements. It is a common element to be optimized according to current traffic conditions.

Phase sequence means to select which phase as the following phase to switch into; it is also a common element to be optimized in the traffic light optimization problem.

Offset is the time difference between the green phases of adjacent intersections for a smooth flow of traffic at multiple intersections, which is also defined as a *green wave*. Definitely, the offset adjustment should take the possible existence of vehicle queues into account.

The above four factors are common optimization criteria in the traffic light control problem. Taking these issues of optimization into consideration, there are currently many traffic light control strategies that vary in complexity. We can categorize them in two aspects [75] [11], as shown in Table 1.1.

Table 1.1: Type of Traffic Light Control Strategy

Control Logic	Control Scope		
	Isolated Intersection	Arterial Coordination	Network Control
Fixed-time	✓	✓	✓
Actuated	✓	✓	✓
Adaptive	✓	✓	✓

From the different structures of traffic scenarios, the control strategies can be classified into isolated intersection control, arterial coordination control, and network control.

Isolated intersection control means to control the traffic lights at a single intersection, taking into consideration local traffic conditions, such as the traffic volume, waiting time, occupancy, vehicle speed, and so on. An intersection consists of a number of directions, and each direction has one or more lanes for vehicles to move in. The common available movements include going forward, turning right, and turning left, although some of them are not considered in some control strategies. Based on all of the permitted movements, the number of phases can be determined, which constitutes all possible optimization elements together with cycle time and split. The offset is not taken into account in the control of a single intersection. The common performance measurement includes maximum intersection throughput and the minimum average delay for vehicles.

Arterial coordination control means that this control strategy is implemented in multiple adjacent intersections in the same direction on the road. The possible available movements and number of phases are similar to traffic light control in an isolated intersection. Since this type of control considers coordination between two adjacent intersections, the offset adjustment cannot usually be ignored. The common measurement of performance includes maximum intersection throughput, minimum average delay for vehicles, and minimum number of vehicle stops. Here, the optimization on the number of stops is aimed at trying to increase the possibility of forming a green wave for traffic flow.

Network control means that this control strategy is used in a traffic network consisting of multiple intercrossing roads. It is an extension of arterial coordination control because of the intercrossing lanes. The possible available movements, number of phases, optimization elements, and performance measurement are similar to traffic light control in arterial coordination control.

On the other hand, from different control logic, traffic light control strategies can be grouped into three categories: fixed-time control, actuated control, and adaptive control.

Fixed-time control is currently the most commonly implemented control strategy in the real world. A large amount of historical traffic data needs to be collected and recorded, and then used to derive some traffic patterns based on different geographic locations of intersections, different time periods (e.g., peak hours), and different weather conditions (e.g., rain). Based on these traffic patterns, fixed values of cycle time, and split and phase sequences can be determined using some optimization methods.

Hence, fixed-time control cannot deal well with a situation in which there are fluctuations in traffic flow, whereby the volume of traffic may change significantly and dynamically from time to time throughout the day. It is only applicable in stable traffic conditions. Under such conditions, it can achieve a better performance than the other two types of control methods due to the small computation time.

Actuated control is a traffic response control that can handle real-time traffic conditions. It uses advanced surveillance technologies to collect real-time traffic data, and then uses the information to design a control strategy to schedule the traffic lights.

The most common optimization factors are cycle time, split, and offset, especially the split adjustment. Usually, actuated control makes control decisions about the duration of green lights under the constraint of a minimum green duration and a maximum green duration. Actuated control can in fact be treated as an intermediate solution between fixed-time control and adaptive control [9].

Obviously, actuated control methods perform much better than fixed-time control, especially in fluctuating traffic conditions. However, actuated control does not optimize the phase sequence, which is also a key factor in solving some cases. Take, for instance, one certain case in which some phases are rather busy with traffic and some phases have fewer or even no vehicles. In actuated control, each phase has a minimum length for green lights such that a number of green lights will be wasted. Furthermore, actuated control does not take traffic conditions in other lanes into account. This leads to less effective performance, especially in heavy traffic conditions. In addition, unsolved problems, such as *the-early-return-to-green* [88], may result in unnecessary stops for vehicles.

Adaptive control, similar to actuated control, is also a strategy for responding to demands from traffic in real time. It is an improvement to actuated control, since it can not only deal with the cycle time, split, offset (optional and only applicable in multiple intersections control) optimization, but also handle the phase sequence problem, thereby totally avoiding the certain cases which would occur in actuated control. However, it is very difficult to achieve a truly adaptive traffic light control because of the dynamic, discrete, and unpredictable characteristics of a traffic network.

1.4 Contributions of the Thesis

We study how to design adaptive traffic light control approaches in a WSN-based ITS to relieve traffic congestion, meanwhile achieving high performance in terms of throughput, average delay, and average number of stops. In this section, we first briefly discuss the limitations of previous works on traffic light control. We then describe the adaptive traffic light control approaches proposed by us for isolated intersections and multiple intersections, respectively. These approaches can overcome the limitations of previous solutions and achieve higher performance.

Although a large number of traffic light control approaches have been proposed for isolated intersections, most of them [67], [17], [14], [81], [63], [64] do not deal with the light sequence adjustment when scheduling traffic lights. Light sequence adjustment can reduce average delay and improve throughput, especially in traffic fluctuation conditions. Most traffic light control approaches use fixed sequences of traffic lights with optimization on the lengths of the lights. Furthermore, they usually take minimum average waiting time and the number of stopped vehicles as objectives, while failing to consider throughput. In addition, many of the approaches [77], [79], [10], [23], [14], [83], [84] employ artificial intelligence techniques, such as neural network, learning and genetic algorithm, to optimize the decision making of the traffic light control. Due to the number of iterations, more computation time is incurred. In addition, many existing works pay little attention to the characteristics of traffic flow, especially when dealing with the discontinuous traffic flow. They do not consider solutions for special traffic circumstances, such as ambulances, fire engines, or traffic accidents.

The existing adaptive traffic light control approaches proposed for multiple intersections are yet not truly adaptive due to the difficulties with coordination control between adjacent intersections, combined with shortcomings in existing work on isolated intersections. In the past decade, a number of well known online traffic light control optimization systems have been developed, such as SCOOT [45] [71], [31], SCAT [55], [73], DYPIC [70], OPAC [38], RHODES [58], UTOPIA [32], [33], and PRODYN [44]. Of these, SCOOT and SCAT are the two that are most often implemented in the real world to improve traffic scheduling efficiency and alleviate traffic congestion. However, SCOOT is limited to fixed green light sequences of each intersection, rather than dynamically adjusting the green lights sequences. SCAT provides a number of pre-defined control plans covering various traffic conditions, but it is not able to optimize the traffic optimization parameters values online.

Taking advantage of real-time traffic data that is detected and transmitted using WSNs, the proposed approaches can dynamically control traffic lights so that the green lights sequences and durations can be adapted to a dynamically changing traffic environment. At the same time, they can achieve more attractive performance in terms of network throughput, average waiting time, and average number of stops compared with previous works. This thesis makes the following specific contributions.

We propose an adaptive traffic light control scheme applied to an isolated intersection, with a wireless sensor network to detect and transmit real-time traffic data, so that the proposed approach outperforms the previous solutions in terms of throughput and average delay. Our approach has the following advantages. First, we define twelve green light configurations in Fig. 3.2, named as *case*, under the constraints of a given intersection model and subject to traffic safety rules. Second, hunger level is defined to guarantee a fairness for each case, which refers to a ranking expressed as a number that represents how many times a green light has governed in the twelve cases. Third, using the real-time traffic data collected, we can identify discontinuous traffic flow. In this circumstance, a *blank*, which means a subinterval that is not occupied by any vehicle, is taken into account for decision making. Fourth, the special circumstances can also be detected through the different types of vehicles, such as ambulances, fire engines, or some vehicles with special priority. We also take them into consideration when making control decisions. We conduct simulations to evaluate the performance compared with previous solutions. Our extensive simulation results demonstrate that our scheme produces much higher throughput and lower average waiting time for vehicles, compared with the optimal fixed-time traffic light control approach and an actuated traffic light control approach [81].

We propose an adaptive traffic light control scheme applied to a traffic network consisting of multiple intercrossing intersections, with a wireless sensor network that detects and provides real-time traffic data, such as traffic volume, waiting time, number of stops, and characteristics of traffic flow. This work is extended from the adaptive traffic light control for a single intersection. The proposed scheme can outperform the previous approaches in terms of throughput, average delay, and average number of stops because it has the following advantages. Similar to the single intersection control, we use 12 cases as the candidate to compete for the green lights under the constraint of maximum waiting time and the upper bound of the hunger level. The blank circumstance and special circumstances have also been considered. In each intersection, the case with the greatest value would obtain green lights. Subsequently, the optimal green light length can be calculated from the local traffic data and traffic conditions of neighboring intersections. We conduct simulations to evaluate our scheme's performance compared with previous solutions. Our extensive simulation results demonstrate that our scheme produces much higher throughput, lower average waiting time for vehicles, and fewer average number of stops, compared with the optimal fixed-time control approach, an actuated control approach, and an adaptive fuzzy control approach [52].

Last, but not least, we apply the proposed approaches into our WSN-based ITS platform iSensNet, and define several traffic scenarios to evaluate the performance. A demonstration [5] with different types of traffic conditions shows that our approach is effective and can be practical in our platform.

1.5 Outline of the Thesis

The remainder of this report is organized as follows. In Chapter 2, we briefly discuss the previous works on traffic light control. In Chapter 3, we present an adaptive traffic light control scheme applied to an isolated intersection. An adaptive traffic light control scheme applied to multiple intersections is described in Chapter 4. We implement our proposed approaches to our WSN-based ITS project iSensNet in Chapter 5. In Chapter 6, we present our conclusions and discuss future works.

Chapter 2

Background and Literature Review

In the past decade, the optimization of traffic light controls in intelligent transportation system environments has attracted a great deal of attention. In this chapter, we review existing works according to the problems addressed. Since our research is focused on adaptive traffic light control, which needs to use traffic data when making scheduling decisions, we first briefly describe works on obtaining traffic data in Section 2.1. Then, in Section 2.2, we provide a detailed survey of existing approaches to controlling traffic lights in both an isolated intersection and multiple intersections.

Section 2.2.1 introduces the existing approaches for an isolated intersection, and Section 2.2.2 discusses the previous approaches for multiple intersections.

2.1 Collecting Real-time Traffic Data

The term traffic data, also called traffic flow data, means relative data on the flow of vehicles moving on a road. It includes speed, density, traffic volume, and so forth. In the problem of controlling traffic lights, the traffic lights controller schedules the lights so that, by outputting different signals, the traffic flow in different roads alternates in crossing the intersection. To achieve effective scheduling, the scheduling should be based on current traffic information. If the traffic light controller does not have current traffic data, it would not realize the traffic conditions and the demand for green lights, and therefore be unable to schedule the lights properly. Hence, traffic data is a vital issue in the problem of controlling traffic lights.

With the growing development of ITS, real-time traffic data is increasingly required. In the past, manual counts were the most popular method. Several observers use some equipment, such as a mechanical counting board and electronic counting board, to count the number of passing vehicles, their corresponding types, and even the number of pedestrians. In addition, real-time traffic data estimation methods have also come into wide use in the past decade. In such methods, a large amount of historical traffic data is collected and used to estimate current traffic conditions [8] [51], such as in the Average Annual Daily Traffic method (AADT) and the Vehicle Miles Traveled method (VMT). Using these methods, it is possible to obtain an approximate idea of current traffic conditions to alleviate traffic congestion.

In the past decade, with the rapid development of technologies, a number of collection methods have been developed to obtain the real-time traffic information [1], [69] [8], [51], [49], [50], [4], such as inductive loops detection, video recognition detection, microwave radar, ultrasonic, and so forth.

Some of these collection methods locate sensors in the road or along the side of the road to detect traffic conditions and calculate real-time traffic information, and then transmit such data to a control center through cables. We call them wire-based collection methods, which include inductive loops detection, video recognition detection, and so forth. Others use wireless technologies to detect traffic information. These are called wireless-based collection methods, and include passive infrared, active infrared, microwave radar, ultrasonic, passive acoustics, and so forth.

Hence, we divide the real-time traffic data collection methods into two categories and give a briefly description of these two types of collection methods in section 2.1.1 and section 2.1.2, respectively.

2.1.1 Wire-based Methods

Wire-based methods represent those collection methods that transmit the detected traffic data through wired technologies, including inductive loops detection, video recognition detection, microwave radar. A brief description of these typical wire-based collection methods is given below.

Inductive loops detection is the most conventional method used in ITS. The loops can be placed in the roadbed [1] or in roadways [51] to create a magnetic field for detecting. When the vehicles pass these loops, the magnetic field is measured to calculate real-time traffic information, such as the number of vehicles, vehicle speed, vehicle length, vehicle weight, even the distance between two adjacent vehicles. This detection method is insensitive to bad weather conditions, such as rain, fog, snow, and so forth. It also can provide better accuracy in counting data compared with other techniques [8]. However, there exist some drawbacks to this detection method. The sensing coverage is limited because of the fixed-location installation. In addition, a pavement equipped with loops is much more easily damaged than a common pavement. Furthermore, it is expensive to implement and maintain these loops, as they are weak and cannot sustain heavy vehicles [8], [51]. Moreover, the accuracy of detection is reduced when a large variety of vehicles is involved.

Video recognition detection also is a popular method used worldwide. Rather than installing sensors directly in the roadbed, video cameras are usually installed on poles to keep watch on road traffic conditions [1]. Initially, the in-built processor should input the location of these cameras so that it can process the images that are recorded. When the vehicles pass by these cameras, the video that is recorded is processed to analyze whether there have been any changes to the image. The number of vehicles, vehicle types, vehicle speeds, lane occupancies, and so forth, are then calculated. This detection method can monitor multiple lanes simultaneously. In addition, it is easy to extend the areas of detection by adding video cameras. However, there are some drawbacks to this method. It is sensitive to different weather conditions, especially rain and fog [51]; and it is also easily affected by the day/night transition. In addition, street lights are required when detecting traffic data in the evening.

Microwave radar detection uses radar-sensing technologies to detect traffic conditions using the sensors installed over the roadway. These microwave radar sensors transmit energy to the area of detection through an antenna. When vehicles pass by, the transmitted energy is reflected back to determine such traffic data as the number of vehicles, vehicle speeds, and vehicle classifications. This method of detection is insensitive to different weather conditions. Furthermore, multiple lanes can be detected simultaneously without any interference. However, there are also drawbacks to this detection method. For example, the continuous wave Doppler radar, which is one of the most popular sensors used in this method, cannot detect stopped vehicles.

2.1.2 Wireless-based Methods

Wireless-based methods represent methods of collection that use some sensor nodes to detect real-time traffic data, which is then transmitted through wireless technologies. They include passive infrared detection, active infrared detection, ultrasonic detection, passive acoustic detection. Below, we briefly introduce these typical methods of collection.

Passive infrared detection uses a passive infrared sensor (PIR sensor) to detect traffic conditions by measuring the infrared (IR) energy radiating from the area of detection. The PIR sensors are usually installed over or adjacent to the roadway. The number of vehicles, the presence of vehicles, vehicle speeds, vehicle types, and occupancies can be detected through this method. However, there are some drawbacks to this detection method. The sensing coverage is limited because of the fixed-location installation. Furthermore, to the sensors have difficulty detecting vehicles during poor weather conditions, such as rain, fog, and snow.

Active infrared detection, which uses a laser radar, illuminates the areas of detection, transmits two beams [8], and then determines the real-time traffic data based on the IR energy that is received. These sensors are usually installed over the roadway. The number of vehicles, presence of vehicles, vehicle speeds, vehicle lengths, and queue lengths can be detected through this method. This detection method can detect multiple lanes simultaneously without any interference. However, this detection method has some drawbacks. It has difficulty detecting vehicles under conditions of low visibility, such as fog. Furthermore, it is expensive to install and maintain.

In ultrasonic detection, sound waves are transmitted to detect real-time traffic data by measuring the time it takes for the signs to return. Ultrasonic sensors are installed over the roadway. Traffic volumes, the presence of vehicles, and occupancy data can be calculated by this method. This method of detection can monitor multiple lanes simultaneously. In addition, some vehicles that exceed height limits can be detected. However, there are some drawbacks to this detection method. When detecting vehicles, it is easily affected by extreme temperatures or bad weather conditions, such as rain, fog, and snow.

Similar to the ultrasonic detection method, passive acoustic detection also involves the emitting of sound waves to detect real-time traffic data by means of measuring the time delay of returning signals. These acoustic sensors are installed along the roadside. Traffic volumes, the presence of vehicles, vehicle speeds, vehicle classifications, and occupancy data can be calculated by this method. This method of detection can monitor multiple lanes simultaneously. In addition, it is insensitive to some weather conditions, such as the rain. However, this detection method has some drawbacks. It is sensitive to extreme temperatures or bad weather conditions when detecting vehicles, such as low temperatures and snow.

Table 2.1 shows the type of traffic data detected by the different collection

Table 2.1: Types of Traffic Data by Detection Method

Sensor Type	Traffic Data					Comm. Tech.
	Traffic Volume	Speed	Presence	Occu-pancy	Classi-fication	
Inductive Loops	Y	Y	Y	Y	Y	W
Video Image	Y	Y	Y	Y	Y	W
Microwave Radar	Y	Y	Y	Y	Y	W
Passive Infrared	Y	Y	Y	Y	Y	WL
Active Infrared	Y	Y	N	N	Y	WL
Ultrasonic	Y	N	Y	N	N	WL
Passive Acoustic	Y	Y	Y	Y	Y	WL

methods [69], [8], [51], [49], [50]. Y means available, N means not available. Some data can be available when some specified requirements are met, especially speed detection and classification detection data.

Table 2.2 presents a summary of environmental factors that affect the performance of different detection methods [25]. Y means can be affected; N means cannot be affected. In both tables, W means wire, while WL means wireless.

2.2 Intelligent Traffic Light Control

As mentioned before, actuated coordination control and adaptive control are both traffic-responsive strategies that design an optimization algorithm as a response to current traffic demands in order to improve operational efficiency and enhance traffic safety. Regarding adaptive traffic light control, it is also worth referring

Table 2.2: Environmental Factors that Affect the Performance of Detection Methods

Sensor Type	Environmental Factors							Comm. Tech.
	Rain	Fog	Snow	Temperature	Wind	Lighting	High traffic flow	
Inductive Loops	N	N	N	Y	N	N	N	W
Video Image	Y	Y	Y	Y	Y	Y	N	W
Microwave Radar	N	N	N	N	N	N	Y	W
Passive Infrared	Y	Y	Y	N	N	N	N	WL
Active Infrared	Y	Y	Y	N	N	N	N	WL
Ultrasonic	Y	Y	Y	N	N	N	N	WL
Passive Acoustic	N	N	N	Y	N	N	Y	WL

to some actuated coordination approaches because of the issue of optimization: the length of the green lights. Furthermore, a number of existing works have not identified these two types of control; rather, they use the term adaptive control to represent both actuated coordination and adaptive control. Not much work has been done on true adaptive traffic light control. Hence, we review the existing studies on actuated coordination and adaptive traffic light control together, under a common term: intelligent traffic light control.

2.2.1 Intelligent Traffic Light Control in an Isolated Intersection

Intelligent traffic light control in a single intersection refers to the application of an actuated or adaptive traffic light control method in one intersection to control the local traffic flow in a dynamic traffic environment. The common objectives include maximum intersection throughput, minimum average delay, and a minimum number of stopped vehicles. A number of studies have focused on this topic with the aim of optimizing the sequence of green lights (*phase sequence*), or the length of the green lights (*phase length*), or both. Many researchers have tried to determine whether several techniques can be applied in this kind of problem, and have produced studies on how to apply them, such as Fuzzy Logic Control, Rules, Neural Network, Learning, Genetic Algorithm, and so forth. We categorize these techniques based on their logic. For example, fuzzy logic control can be grouped together with rules, and neural network can accompany learning methods. We then review existing approaches to intelligent traffic light control in a single intersection.

(1). Intelligent Traffic Light Control in an isolated intersection Using Fuzzy Logic and Rules

In the past decade, there have been several studies applying fuzzy logic to traffic light control to optimize traffic conditions in an isolated intersection [91], [63], [81], [67], [60]. A common method of implementation is to first use some fuzzy membership functions to fuzzify the traffic input, such as the vehicle arrivals in each lane and the length of the queue in each lane, to cater to the fuzzy rules format. After that, reasonable and corresponding fuzzy rules are designed to determine the fuzzy output results, based on the researchers' own knowledge and experience. The fuzzy output is then defuzzified to crisp values as the output decisions.

In this procedure, membership function should be a significant factor in the performance of applications of fuzzy logic. Different fuzzy membership functions have different performances. The most widely used membership is Mamdani's MaxMin inference method. In the following section, unless otherwise specified, the membership function used in existing work refers to this Maxmin method.

Fuzzy logic was first introduced by Zadeh [89] [90], and Pappis and Mamdani [67] were the pioneers in applying it to traffic control. They considered an unsaturated isolated intersection with a simple one-way traffic control without permission to turn. The inputted traffic data were random vehicle arrivals in each lane. Their paper only focused on optimizing the current length of the extension of green light in response to current traffic conditions, which can be regarded as actuated control. *Degree of confidence* was defined to measure the respective suitability of several possible durations of extension for the current green light, and the duration of the extension with the highest degree of confidence would be selected.

Another example of an application, proposed by Bisset and Kelsey [17] [14] simulated a two-phase intersection with going through, right turn, and left turn movements. The objective was to minimize the average waiting time of vehicles. In these two papers, the length of the current cycle time, the traffic volume in both lanes with a green light and lanes with a red light are treated as traffic input. Appropriate fuzzy rules were then designed to produce the output. The approaches proposed in these two papers perform much better than conventional approaches, especially in heavy traffic.

A two-stage fuzzy logic algorithm is presented in [81]. A fuzzy logic controller was designed for a four-direction isolated intersection with going through and left turn movements. The inputted traffic data, detected by intrusive sensors placed upstream in each direction, are traffic volumes and the queue lengths of each lane. These data are used to determine whether to extend or terminate the current green light. In the first stage, the detected traffic data are used to estimate the relative traffic intensities in the conflict lanes. In the second stage, these intensities can be used to produce the decision on whether the current green light should be extended or terminated. The performance metrics contain average vehicle delay and the percentage of stopped vehicles, and are compared with an actuated algorithm. The simulation results show that the proposed algorithm can achieve a lower average delay time for vehicles, while the percentage of stopped vehicles remains unchanged.

Using fuzzy logic control to control the traffic lights in an isolated intersection has many advantages. One obvious benefit is that minimal computation resources are needed compared with other approaches to adaptive traffic light control [88], so that the computation time requirements are likewise minimal. Moreover, the current state of the system can be better represented since one fuzzy input can be fuzzified by membership functions to a different fuzzy set with a corresponding degree of fuzziness. However, similar to rule-based and knowledge-based approaches, the design of the fuzzy rules relies on expert knowledge, so that it is difficult to obtain the optimal rules.

(2). Intelligent Traffic Light Control in an isolated intersection Using a Neural Network and Learning

Both the applications of neural network and reinforcement learning use some learning methods to train input data to obtain optimal output. However, many differences between them still exist. A neural network can be treated as some kind of supervised learning method, which must have a set of training data and expected output values. The weight of the neural network should be optimized to reduce the difference between the output value and the expected value as much as possible. On the other hand, reinforcement learning can be treated as some kind of machine learning [59]. In reinforcement learning, different actions of an agent will have a different influence on the surrounding environment. An action with a positive impact on the environment will be awarded with a positive reward, while an action resulting in a negative impact will be awarded with a negative reward [88].

Some existing works have applied reinforcement learning and neural networks to control the traffic lights in an isolated intersection, such as SARSA (State Action Reward State Action) [77] [79], and Q-Learning [10].

The first known attempt to apply reinforcement learning to traffic light control problems was that by Thorpe [77] [79]. A single intersection with 4×4 traffic lights was considered. There are two phases, a north-south permission and an east-west permission. Thorpe used a neural network to predict the Q-values for each possible decision, based on the total waiting time of all vehicles and the time since the lights last changed. This method is capable of dealing with a huge number of states, where the learning time may be quite large. Compared with a fixed control and a rule-based control in realistic simulations with varying speeds, this method presents near optimal performance.

Abdulhai *et al.* proposed a truly adaptive traffic light control strategy using Q-Learning to a four-direction and two-lane intersection in [10]. Only the going forward movement was permitted to vehicles. The traffic data, vehicle arrivals, are defined as individual poison processes and the average arrival rates are pre-defined. The objective is to minimize the average delay for vehicles. Based on the only existing two phases, they considered a fixed cycle time, so that they only needed to decide whether to switch the green light at each time. The only optimization is the duration of the green light. The queue lengths of each lane and the elapsed time since the last change in phase are treated as the state variables, and the delay accumulated from two phases is treated as the reward. A technique called Cerebellar Model Articulation Controller (CMAC) is proposed to store and generalize the value function of the learned action. Compared with a fixed-time control, this method can achieve a much lower average delay of more than 50% in variable traffic conditions.

(3). Intelligent Traffic Light Control in an isolated intersection Using a Genetic Algorithm

Several existing studies have applied a genetic algorithm in controlling the traffic lights of an isolated intersection, such as [23].

The genetic algorithm, first introduced by Goldberg [39], has been adopted in problems of optimizing the control of traffic lights of single intersections in several studies in the past decade [23].

Chen and Shi applied a real-coded genetic algorithm (RGA) to an isolated two-way intersection with multiple lanes in [23]. Going through, right turns, and left turns are permitted, and form four phases. This paper designed a traffic flow model and then used RGA to optimize the green times and cycle time in order to minimize the throughput.

(4). Intelligent Traffic Light Control in an isolated intersection Using Hybrid Techniques

There are also several studies on applying hybrid techniques to controlling traffic lights in an isolated intersection, such as a combination of fuzzy logic and learning [16], fuzzy logic and a genetic algorithm [84] [24] [53], and fuzzy logic, a genetic algorithm, and learning [83].

Bingham considered an isolated intersection with two one-way streets in [16]. The inputted traffic data are the total traffic volume in the lane with the green light and in the other lanes.

The author proposed a Generalized Approximate Reasoning-based Intelligent Control (GARIC) developed by Berenji and Khedkar [14], which can essentially be treated as an Actor Critic Reinforcement Learning (ACRL) method [76]. GARIC includes two components: an action selection network (ASN) and an action evaluation network (AEN).

Based on the current traffic environment, ASN, which was designed in the form of a fuzzy logic controller, proposes several fuzzy rules to generate continuous action candidates to represent the possible duration of the extension of the current green light. Then, AEN, which was designed in form of a fully connected feed-forward neural network, computes the value of each state. The candidate action with the greatest value can be selected as the decision. If it can achieve a positive impact on the environment, this action would gain a corresponding reward. Meanwhile, a TD error was defined to update the parameters of the fuzzy membership functions of ASN and the weights of AEN. The results of the simulation showed that in conditions of heavy traffic, the proposed method can outperform the original fuzzy logic control.

Wei *et al*. [83] [84] proposed a traffic light control method based on fuzzy logic and neuro-fuzzy and multi-objective genetic algorithms (MOGA) for a four-direction isolated intersection with going through and left turn movements. The optimization criteria are the length of the green light and the sequence of green lights, and the performance metrics include vehicle delay and the percentage of stopped vehicles. A fuzzy logic controller was designed to determine whether to extend or terminate the current green phase and select the sequences of phases. A method based on neuro-fuzziness is used to predict traffic parameters in the fuzzy logic controller. Several optimizing sets of parameters for the fuzzy logic controller can be searched through (MOGA). Through simulation, this proposed method can achieve a better performance than a traffic-actuated control.

2.2.2 Intelligent Traffic Light Control in Multiple Intersections

Intelligent traffic light control in multiple intersections refers to the application of an actuated or adaptive traffic light control method in multiple interconnected intersections to control the flow of traffic in a dynamic traffic environment. It includes arterial coordination control and traffic network control. Different from traffic light control in an isolated intersection, the application in multiple intersections needs to take into account the influence from neighboring intersections. As mentioned before, different from an isolated intersection application, an additional optimization parameter is the offset adjustment, which refers to the relationship between adjacent intersections. The common performance measurement includes the maximum intersection throughput, the minimum average delay for vehicles, and the minimum number of vehicle stops. Here, the optimization on a number of stops is an attempt to increase the possibility of forming a green wave for traffic flow.

As mentioned before, a number of well known traffic light control systems have been developed, such as SCOOT [45] [71], [31], SCAT [55], [73], DYPIC [70], OPAC [38], RHODES [58], UTOPIA [32], [33], and PRODYN [44]. We will give a brief introduction to the two most widely deployed systems, SCOOT and SCAT. Man studies have focused on this topic, using some techniques to optimize the green light sequence (*phase sequence*), or green light length (*phase length*), or both. Similarly, we group these studies by the techniques they applied, such as Fuzzy Logic Control, Neural Network, Genetic Algorithm, Petri-Net, and so forth. We when review existing approaches to intelligent traffic light control in multiple intersections.

(A). SCOOT

The Split Cycle, Offset Optimization Technique (SCOOT) is a centralized traffic response system for coordinating traffic lights in urban areas as an automatic response to fluctuations in the flow of traffic. SCOOT, which was first developed by Hunt *et al.* [45] [46], can essentially be treated as a TRANSYT-7F traffic model with an optimization algorithm for online application [19] [18] [15]. With the development of advanced technologies, some new features have been added, extending to several versions. SCOOT has been widely applied in over 150 cities in the United Kingdom and elsewhere.

SCOOT uses embedded sensors to detect real-time traffic data, and also predicts the vehicle arrival pattern, vehicle delay, and vehicle stops. Based on this information, the system makes small changes to the optimization parameters, such as cycle length, split, and offset. The intersections involved are grouped in several subareas, and the traffic lights in each subarea must use a common cycle length. Under the constraints of a minimum and a maximum cycle length, the cycle length adjustment keeps a maximum degree of saturation of 90% at the most saturated intersection. Regarding split optimization, the optimizer makes a decision on whether to extend or terminate the current green phase a few seconds before the phase changes. Then, the offset optimizer assesses whether the performance can be improved if the offset is altered 4 seconds earlier or later. The split and offset alteration candidates that improve the performance of the intersection are then selected and implemented immediately. If the changes did not seem beneficial, they are submitted to the local controllers.

(B). SCATS

The Sydney Coordinated Adaptive Traffic System (SCATS) is another widely used system that can provide intelligent traffic plans to schedule traffic lights.

The result is a substantial reduction in vehicle delay, particularly in peak periods. SCATS [55] [73] was developed by Australian researchers. It can be considered an optimization control method that is between the first generation control of UTCS and the second generation control of UTCS [9]. SCAT consists of three types of controllers: a central controller, regional controllers, and local controllers.

The local controller, which is installed at each intersection, is responsible for collecting traffic data from traffic detectors, processing the collected data, making assessments of the performance of the detectors, and putting the traffic control decisions into operation.

The regional controller, which maintains autonomous control of several local controllers in its area, is the heart of SCATS. It is responsible for analyzing the information that has been preprocessed by the local controller, and implementing the corresponding signals.

The central controller, which focuses on monitoring the entire system, is responsible for providing system management support, data backups, fault analysis, and system inventory facilities [88].

Both SCOOT and SCATS use detected real-time traffic data to make control decisions in a whole traffic network, with the purpose of reducing delay, decreasing stops, and so on. They have some common features, such as cycle length, phase duration, phase sequence, and an offset that is fixed within short time periods and updated every few minutes to avoid disturbing normal operations. However, one critical difference between the two systems is that, unlike SCOOT, SCATS does not have a traffic model or a traffic light control plan optimizer [88]. SCAT pre-specifies several traffic light control plans, and selects the best control decision based on real-time traffic conditions [9].

(1). Intelligent Traffic Light Control in Multiple Intersections Using Fuzzy Logic and Rules

There are several studies on applying fuzzy logic or rule-based approaches to controlling traffic lights in multiple intersections [61] [26] [27] [52] [28], [92] [82].

Nakatsuyama *et al.* considered two one-way adjacent intersections on an arterial road in [61]. The authors applied fuzzy logic to model the control and developed corresponding reasonable fuzzy control rules, to determine whether to extend or terminate the current green lights for the downstream intersection based on the upstream traffic conditions.

Chiu and Chand [26], [27] pioneered the application of fuzzy logic to multiple intersections in a network. Their papers considered an intersection with two-way streets where only the going through movement was permitted. Fuzzy rules were used to adjust cycle time, phase split, and offset parameters independently, based on local traffic conditions. Adjustments to the cycle length and splits were made based on the degree of saturation of each direction of each intersection. These adjustments can be tested through simulations to significantly reduce the average delay. The offset is adjusted by the fuzzy sets, which were designed to determine the degrees of saturation in order to coordinate the lights in the adjacent intersection in such a way as to minimize stops in the dominant direction.

Lee *et al.* [52] proposed a traffic fuzzy controller for a set of eight-phase intersections that permit going forward and left turn movements. The controller installed two sensors at each lane to detect real-time traffic data; and then, based on the calculated information, to determine both the phase sequence and phase length by an optimization method. The controller consists of three modules with an independent fuzzy rules base: the next phase module, the stop module, and the decision module. The next phase module assesses the degree of urgency of each phase, and then chooses the most urgent phase as the next phase candidate. The stop module evaluates the degree to which the green phase can stop. The decision module makes a decision on whether to switch to the next phase based on the next phase module and the stop module. The controller can periodically make such a decision. The simulation results show that the proposed fuzzy controller can gain obtain better performance by reducing the average delay, particularly in conditions of fluctuating and heavy traffic.

There are also some studies on rule-based and knowledge-based intelligent traffic light control systems [54] [65] [34]. Owen and Stallard [65] described a rule-based scheduling algorithm to control the lights of a traffic network in a distributed way, called the Generalized Adaptive Signal Control Algorithm Project (GASCAP). GASCAP consists of three key parts: a queue estimation model, a set of rules for controlling uncongested traffic, and a fixed time control algorithm for congested traffic. The queue estimation algorithm uses an upstream detector to calculate the arrival of vehicles, and then to estimate the volume of traffic approaching an intersection and the traffic volume in the queue. Regarding the uncongested traffic control, which can be treated as the major difference between GASCAP and other traffic control optimization approaches, GASCAP has five sets of rules. Based on the estimated data, each of these sets can calculate priority values related to green light demand, coordinated progression, saturation urgency, spillback case, and minimum green light durations, respectively. The simulation results indicated that GASCAP significantly reduced the delay and increased the throughput for each of these networks.

(2). Intelligent Traffic Light Control in Multiple Intersections Using a Neural Network and Learning

In the past, some studies, such as [77] [78], applied reinforcement learning and a neural network to control the traffic lights in multiple intersections.

Thorpe [77], [78] conducted one of the pioneering studies on controlling traffic lights using reinforcement learning. The authors considered a simple traffic network with 16 one-lane four-direction intersections. Going forward and right turn movements were permitted, resulting in the existence of only two phases. SARSA (State Action Reward State Action) [76] was used to represent the current state of traffic and the train intersection controller. The performance of four different representations of the current state of traffic was analyzed using two reinforcement methods. In simulations, the proposed method performed better at minimizing total traffic travel time, individual vehicle travel times, and vehicle wait times compared with a fixed-time traffic light control.

(3). Intelligent Traffic Light Control in Multiple Intersections Using a Genetic Algorithm

Several studies, such as [36] [40] [41], have applied a genetic algorithm to control traffic lights in multiple intersections.

The first attempt to apply a GA to control traffic lights was by Foy *et al.* in [36]. That study considered a traffic network with four intersections with the purpose of minimizing delay. The flow of traffic was assumed to be constant. The cycle time, split, and offset were three optimization criteria. The authors designed a traffic light timing strategy based on the parallel, global, and robust search characteristics of GAs. The results of the simulation showed that the proposed method could have a better performance in reducing delay.

Hadi and Wallace [40] used genetic algorithms in combination with the TRAN SYT-7F [7] to optimize all four basic elements in designing a traffic light control strategy: phase sequences, cycle length, split, and offset. The main purpose of this GA application is to optimize the phase sequences, while the other three elements can be optimized by TRANSYT-7F. The authors proposed two implementations. In the first method, the GA application and the TRANSYT-7F optimization were implemented concurrently in order to obtain an optimal solution. In the second method, the GA was used to optimize cycle length, phase sequences, and offsets; and then TRANSTY-7F was used to adjust the resulting green light duration. The simulation results showed that both methods could optimize the phase sequence and phase length. However, the first method produced a better performance, while also requiring a longer computation time.

(4). Intelligent Traffic Light Control in Multiple Intersections Using Hybrid Techniques

There have also been several studies on applying hybrid techniques to controlling traffic lights in multiple intersections, such as a combination of fuzzy logic, learning, and an evolution algorithm [30] [29] [74], and fuzzy logic, a neural network, and SPSA [75].

Choy *et al*. [30], [29], [74] proposed a distributed, cooperative approach to managing the real-time traffic in an arterial network by using a hybrid multi-agent system involving an effective traffic light control strategy. The multi-agent system architecture is designed in a hierarchical way, consisting of three layers listed from lowest to highest, intersection controller agents (ICA), zone controller agents (ZCA), and a regional controller agent (RCA). ICA is responsible for controlling an isolated intersection in the traffic network; ZCA is responsible for controlling several ICAs; and RCA is responsible for controlling all of the ZCAs.

The large-scale traffic light control problem is divided into various sub-problems, each of which can be solved by the respective individual intelligence agents from different layers. Each individual agent has a fuzzy-neural decision making module (FNDM), which can make control decision by mediating lower-level agents and their respective higher-level agents. FNDM consists of five layers to represent: in between the layers, the fuzzification, implication, consequent, and defuzzification processes. The fuzzy input data includes occupancy, traffic flow and changing rate of the traffic flow, and intersection cooperative factors recommended by corresponding ICAs. All of these are detected in real-time by inductive loops. The Gaussian membership function is used to fuzzify and transfer them to the third layer to fire the fuzzy rules. In the fourth layer, the fuzzy output is presented, and then defuzzified as the traffic light control strategy in the fifth layer.

Furthermore, this system can achieve online adjustment in response to a dynamic traffic environment. This multistage online learning process includes three steps: reinforcement learning, weight adjustment, and fuzzy relations adjustment. First, the reinforcement from the environment would be back-propagated to the RCA and then to all of the lower-level intelligent agents. Second, based on this information, each agent proceeds to dynamically adjust the learning rate of each neuron and subsequently to adjust the weights of the neurons in the FNDM. Finally, the reinforcement is used to dynamically adjust the fitness value of each neuron in the FNDM. If the fitness values are smaller than some predefined values, the fuzzy relations will be updated by the evolutionary algorithm.

The performance evaluation shows that the proposed multi-agent architecture can reduce the total average delay by 40% and the total number of vehicle stops by 50%.

Srinivasan *et al.* [75] also proposed another multi-agent system approach to the real-time traffic light control problem in an urban traffic network. The first multi-agent system mechanism is similar to the work in [30] [29] [74]. The second multi-agent system was designed by integrating the simultaneous perturbation stochastic approximation theorem (SPSA) in fuzzy neural networks (FNN). It uses SPSA to update the weight of each neuron. The results of the simulation show that the proposed multi-agent system can reduce the average delay of each vehicle by 78% and the average number of vehicle stops by 85%, as well as provide a significant amount of improvement when the complexity of the simulation scenario increases.

2.3 Summary

In this chapter, we first reviewed the existing approaches to collecting real-time traffic data from two categories of method: wire-based and wireless-based. We then discussed the advantages and disadvantages of all of the existing collection methods. Next, we reviewed the existing actuated coordination and adaptive traffic light control approaches, as applied in an isolated intersection and multiple intersections, respectively. In each area of application, we also discussed the different techniques used in the problem of controlling traffic lights. Furthermore, we referred to the drawbacks to the existing works and showed that accuracy and real-time should be the most important factors in measuring the performance of the existing approaches.

Chapter 3

ATLCII: Adaptive Traffic Light Control for an Isolated Intersection

In this chapter, we investigate the problem of adaptive traffic light control in an isolated intersection using real-time traffic information collected by a wireless sensor network (WSN). Existing studies have mainly focused on determining the length of the green lights in a fixed sequence of traffic lights. We propose an adaptive traffic light control scheme that adjusts both the sequence and length of the traffic lights in accordance with the real-time traffic that is detected. Our scheme considers a number of traffic factors such as traffic volume, waiting time, vehicle density, and others, to determine the green light sequence and the optimal green light length. The results of the simulation demonstrate that our scheme produces much higher throughput and a lower average waiting time for vehicles compared with the optimal fixed-time control approach and an actuated control approach.

This chapter is organized as follows. In section 3.1, we briefly introduce the work. In section 3.2, we model the problem and define some notations. In section 3.3, we propose an adaptive traffic control scheme to detect traffic conditions, and then determine the sequence and length of the green lights. In section 3.4, we evaluate the performance of our scheme through simulations. In section 3.5, we summarize this chapter.

3.1 Overview

Intelligent transportation system is an automatic road traffic management system that can manage road traffic with the goal of improving traffic safety, optimizing the speed of the flow of traffic, and minimizing the energy consumption of vehicles running on the roads. Traffic light control system play a key role in ITS due to their high performance in relieving traffic congestion.

Most current traffic light control systems use one of three control approaches: fixed-time, actuated, or adaptive. In each case the overriding goal is the same, to maximize safety, speed, and energy efficiency or minimize waiting time and the number of vehicle stops. This is not a simple problem in a dynamically changing traffic environment in which each traffic light system must take into account a wide range of variables, such as the type of intersection (whether single-lane or multiple-lane), traffic volumes, time of day, the effects on other roads, and the involvement of pedestrian traffic.

The two main traffic control systems currently in use around the world are

SCOOT [71] [31] and SCAT [73]. In addition, various computational intelligence approaches have been proposed for the design and implementation of adaptive light control systems, such as the Genetic Algorithm [23], Fuzzy Logic Control [67] [62] [57], Neural Network [83] [16], Queuing Network [42] [43], and so on.

Most existing studies [71] [31] [73] [23] [67] [62] [57] [83] [16] [42] [43] use a fixed sequence for controlling traffic lights, and take the minimum average waiting time and the number of vehicle stops as objectives. However, they fail to consider the throughput, and pay little attention to the characteristics of traffic flow and special traffic circumstances, such as ambulances, fire engines, or traffic accidents. Therefore, it is desirable to dynamically control traffic lights so that the sequence of green lights is adaptive to a dynamically changing traffic environment, with the objectives of maximizing intersection throughput and minimizing the average waiting time.

We propose an adaptive traffic light control scheme to use the traffic information that is detected to determine the sequence and length of traffic lights. The scheme contains three steps: vehicle detection, green light sequence determination and light length determination. We compare the performance of our scheme with a fixed-time control approach and an actuated control approach. The results of the simulation show that our scheme can achieve a higher intersection throughput and a lower average waiting time for vehicles.

3.2 Problem Formulation and Notations

The problem of controlling traffic lights is how to respond adaptively to a dynamically changing traffic environment to improve efficiency of control, under the constraint of guaranteeing fairness for each lane. Here, efficiency includes the intersection throughput (the number of vehicles passing through the intersection), and the average waiting time for vehicles.

To model this problem, we will consider (see Fig. 3.1) a sensor-equipped intersection with four directions (north, south, east, and west), each of which has two lanes, one for going forward and the other for turning left. Each lane is controlled with a traffic light that offers two signals, red for stop and green for go. A total of sixteen sensor nodes are placed on the eight lanes to detect the flow of traffic. Each lane has two sensor nodes: one is installed at the intersection and the other at a given distance, called the $SensorDistance$, from the intersection.

Subject to traffic safety rules, there exist a maximum of twelve different possible cases of green lights (see the Appendix), depicted in Fig. 3.2. Therefore, in the face of a dynamically changing traffic environment, the problem is transformed to the decision on which case should obtain a green light next and how long it should last for.

To formulate the problem, we use the following notations, and assume that all vehicles run at a constant speed $speed$ and that all of the vehicles are of the same

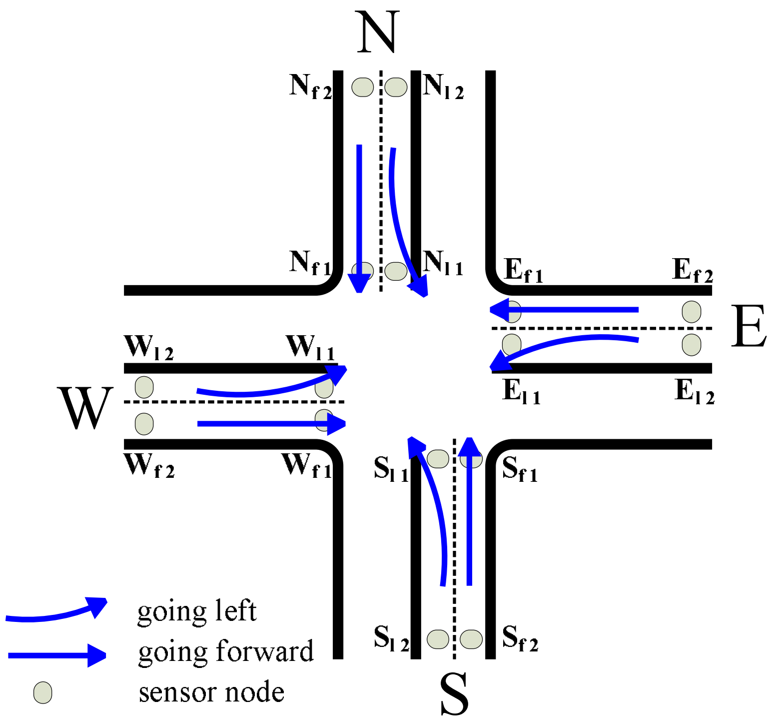

Figure 3.1: Isolated Intersection Model

type:

I = {north, south, east, west}, which represents the four directions of the intersection.

J = {forward, left}, which represents the two movements permitted.

R = {1,2,3,...,8}, which represents the eight lanes of the intersection.

C = {1,2,3,...,12}, which represents the twelve configurations of green lights.

TP: total throughput.

$AVGWT$: average waiting time.

T: total time period.

$DP(k,t)$: number of vehicles passing through the intersection at case k at time t, k

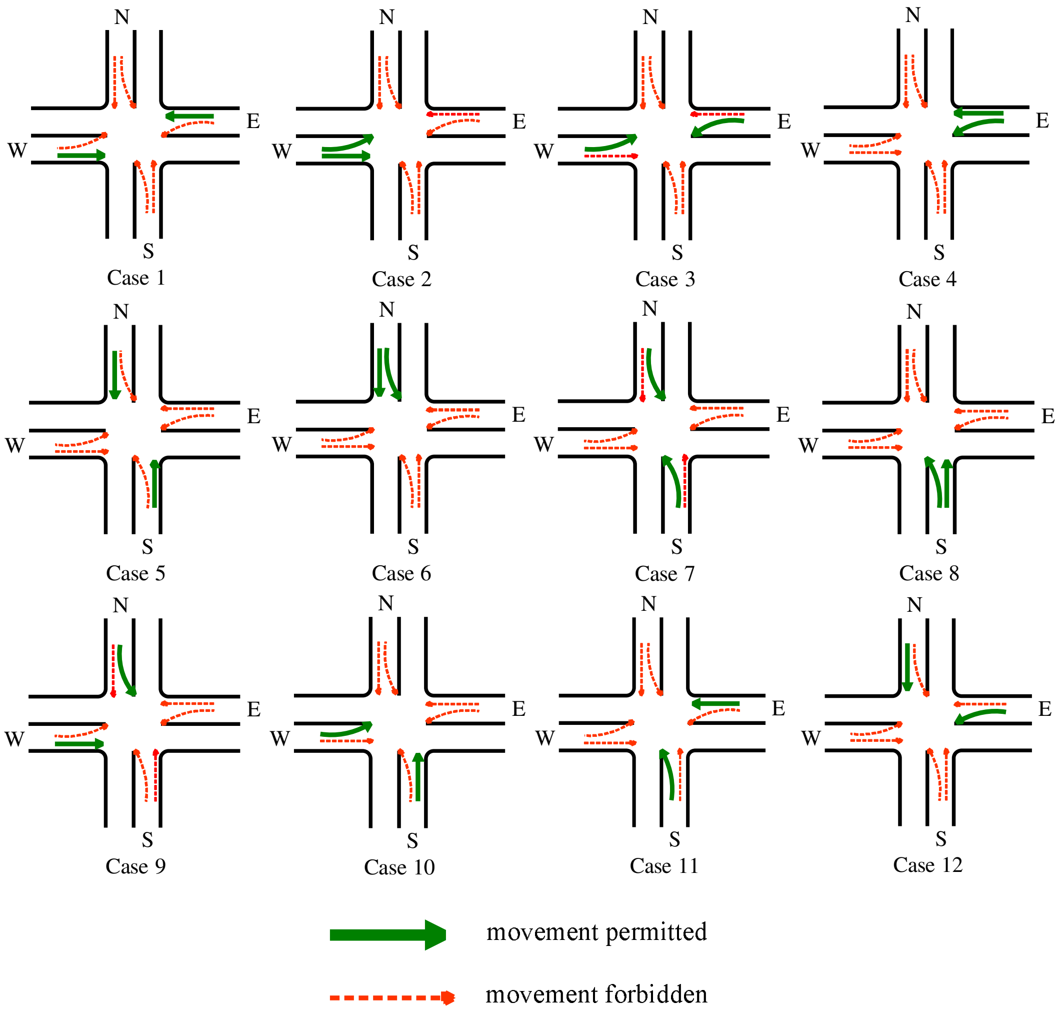

Figure 3.2: Twelve possible configurations of green lights

$\in C$.

$WT(k,t)$: sum of the waiting time of vehicles at case k at time t, $k \in C$.

$RM(k,t)$: number of vehicles at case k at time t, $k \in C$.

X_{y1}: sensors installed at the intersection in lane y at direction X, $X \in I$, $y \in J$.

X_{y2}: sensors installed with distance $SensorDistance$ from the intersection in lane y at direction X, $X \in I$, $y \in J$.

Efficiency:

$$TP = \frac{\sum_{t=1}^{T} \sum_{k \in C} DP(k,t)}{T} \tag{3.2.1}$$

$$AVGWT = \frac{\sum_{t=1}^{T} \sum_{k \in C} WT(k,t)}{\sum_{t=1}^{T} \sum_{k \in C} RM(k,t)} \tag{3.2.2}$$

Equation 3.2.1 calculates the number of vehicles passing through the intersection within a unit of time (TP), Equation 3.2.2 calculates the average waiting time for vehicles ($AVGWT$) during time period T.

In order to maintain fairness for each case, we introduce a maximum waiting time T_{max} and the *hunger level*. The former is to guarantee that each vehicle in a lane with a red signal will not wait too long. The latter is to reflect the times for the green lights of the case. It is to avoid the circumstance in which one case has fewer vehicles and has not gotten a green light for a long time while the other eleven cases have gotten green light in turn again and again. Hence, if the hunger level of one case is high, this means that this case did not have a green light for a rather long time, thus its green light demand is high; while a low hunger level means that the green light demand is low.

3.3 The ATLCII Scheme

In this section, we propose an adaptive traffic light control scheme based on the above established model. The scheme contains three steps: real-time traffic detection, green light sequence determination, and light length determination. Real-time traffic detection involves detecting and calculating traffic information in a real-time manner. Green light sequence determination involves using traffic information to determine the next green light for the case with the largest demand. Light length determination refers to determining how long the green light will last for.

At the beginning, we first set a control cycle $T_{control}$, which is defined as an upper bound of light length. This value of $T_{control}$ is based on expert knowledge.

3.3.1 Real-time Traffic Detection

The first step is to detect the arrival and departure rate of vehicles in each lane, and then collect relevant data, with sensor nodes installed in each lane of the intersection, as illustrated in Fig. 3.1. Sensor nodes detect the number of vehicles in each lane and each vehicle's ID and type. X_{y1} is responsible for detecting vehicles at the intersection; X_{y2} is responsible for detecting vehicles from the intersection with the distance $SensorDistance$ mentioned. $SensorDistance$ is equal to $T_{control} \times speed$ so that X_{y1} will get the information on the vehicles that will reach the intersection after $T_{control}$ time in advance through the communication between X_{y1} and X_{y2}.

Using these detected data, the arrival rate and departure rate in each lane can be determined in real-time. In a lane with a green light, both the arrival and departure rates are calculated in real-time. In a lane with a red light, the departure rate is zero and the arrival rate reflects how many vehicles are waiting in the lane.

Because each vehicle has a length $L_{vehicle}$, we divide lane length L_{lane} into m intervals with the same length $L_{interval}$ equal to $\frac{L_{lane}}{m}$, shown as $D_1, D_2, ..., D_m$. D_i is demonstrated as interval $[d_{i-1}, d_i]$; d_i is defined as the distance to the intersection, which is equal to $i \times L_{interval}$. $RM(D_i, t)$, $AR(D_i, t)$, $DP(D_i, t)$ are defined as the number of vehicles in, arriving in and departing from D_i at time t, respectively. The arrival rate in D_i at time t is equal to the departure rate in D_{i+1} at time $t - 1$. $RM(D_i, t)$ can then be calculated(in equation 3.3.1 and equation 3.3.2). After that, $G(D_i)$ can be determined (in equation 3.3.3), which is defined as the density of the traffic flow in interval D_i. The density of the traffic flow in the lane $VDDF(D_1, D_2, ..., D_m)$ can be demonstrated in equation 3.3.4.

$$AR(D_i, t) = DP(D_{i+1}, t - 1)\} \tag{3.3.1}$$

$$RM(D_i, t) = max\{RM(D_i, t - 1) + AR(D_i, t) - DP(D_i, t), 0\} \tag{3.3.2}$$

$$G(D_i) = \frac{RM(D_i, t)}{L_{interval}} \tag{3.3.3}$$

$$VDDF(D_1, D_2, ..., D_m) = f(G(D_1), G(D_2), ..., G(D_m)) \tag{3.3.4}$$

This is a nonlinear function, an example of a random function is shown in Fig. 3.3. Different intervals have different traffic flow densities, which means a different number of vehicles. At some intervals, there exists a sub-interval without any vehicle, and its length is larger than $L_{vehicle}$. Here, we define this sub-

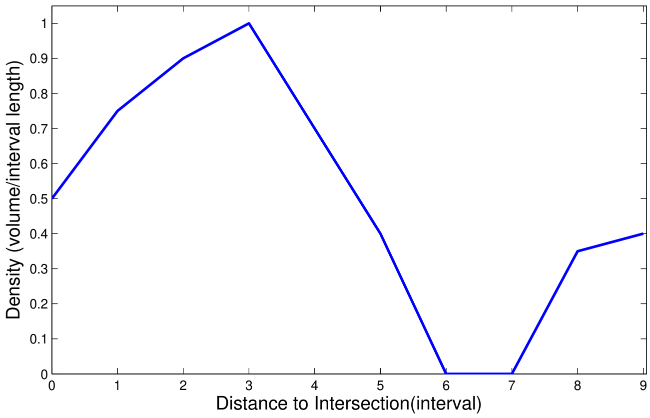

Figure 3.3: Vehicle Distribution Density Function

interval as a *blank*. In order to accurately check blanks, $L_{interval}$ should be equal to $2.5 \times L_{vehicle}$. Then, if there exists a $G(D_i)$, whose value is lower than 0.4 and higher than 0.2, we can decide that there is a blank in D_i and that the length of the blank $L(blank)$ is equal to $L_{vehicle}$. If there exists a $G(i)$, whose value is lower than 0.2, we can decide that there is a blank in $G(D_i)$ and that the $L(blank)$ is equal to $2 \times L_{vehicle}$.

What needs to be considered with blanks is dealing with the problem that arises when a blank reaches the intersection and the current green light is for its lane, which leads to a waste of a green light. This means that, within the a period of time equal to the length of this blank $L(blank)$, the number of vehicles passing through the intersection is not as large as supposed, so that there is an increase in the total waiting time of vehicles in other lanes. Therefore, we try to release the blank by making the blank reach the intersection with the red light for that certain lane.

3.3.2 Green Light Sequence Determination

The second step is to make a decision to determine the sequence of green lights, using real-time traffic data. In order to make this decision, we define $GLD(k,t)$ to indicate case k's green light demand at time t, so that the case with the most urgent demand should get the next green light. Since our objectives are to increase the throughput and decrease the average waiting time, the number of vehicles detected in each lane, their corresponding waiting time, and the blank circumstance are influential factors. To guarantee that each case will not wait too long, it is also necessary to take the hunger level into account in determining the sequence of green lights. Furthermore, special circumstances and the effect from adjacent intersections can also play a role. Equation 3.3.5 demonstrates all of the factors of $GLD(k,t)$.

$$\begin{aligned} GLD(k,t) &= a_1 \times TV(k,t) + a_2 \times WT(k,t) + a_3 \times HL(k,t) + a_4 \times BC(k,t) \\ &\quad + a_5 \times SC(k,t) + a_6 \times Neibor(k,t) \end{aligned} \tag{3.3.5}$$

Here, $TV(k,t)$, $WT(k,t)$, $HL(k,t)$, $BC(k,t)$, $SC(k,t)$, $Neibor(k,t)$ are defined as the weight of the traffic volume, average waiting time, hunger level, blank circumstance, special circumstance and influence from neighboring intersections of case k at time t, respectively, and a_i is defined as the coefficient of these parameters to demonstrate their priorities, $i = 1,2,3,4,5,6$. In our problem, since the distance between two intersections is longer than $SensorDistance$, $Neibor(k,t)$

can be ignored in this problem. Therefore, we discuss the five main factors sequentially as shown in Fig. 3.4.

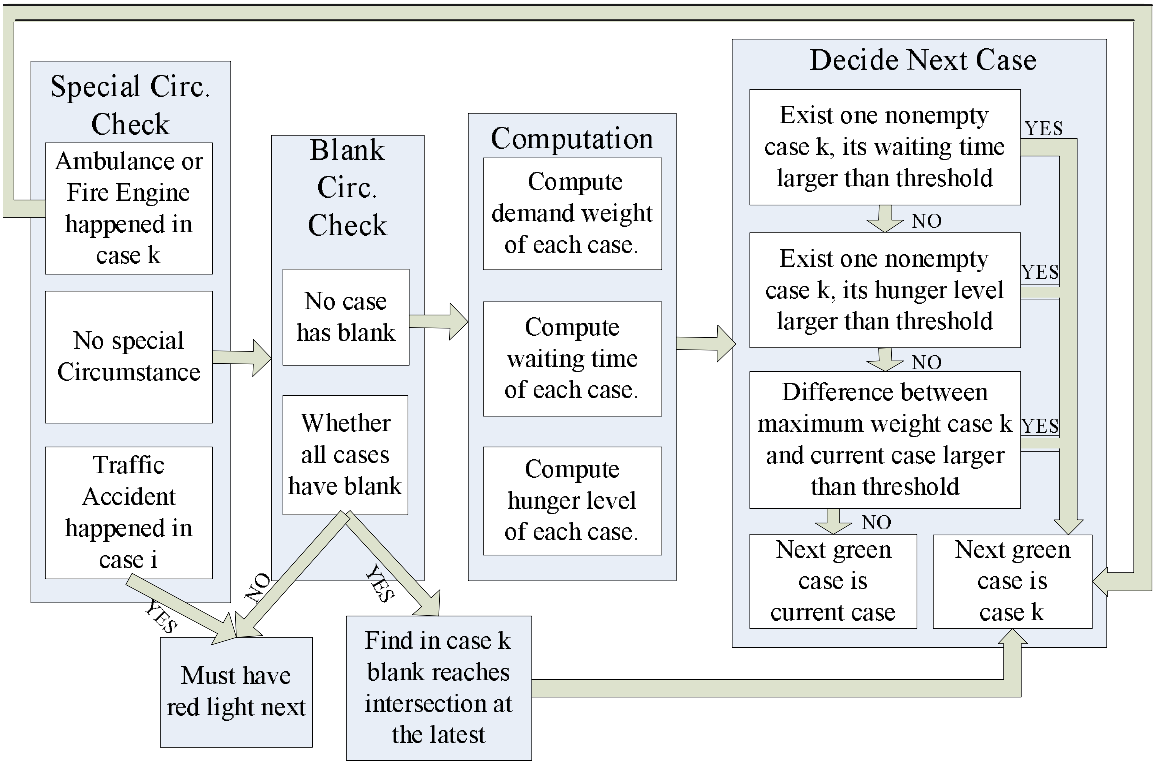

Figure 3.4: Green Light Sequence Determination

1) Traffic Volume

After the $VDDF(d, RM(t))$ calculation, we can calculate the weight of the traffic volume of each case. To calculate $TV(k,t)$, we first need to obtain $TraVol(i,t)$, which is defined as the total number of vehicles in lane i, from time t to following $T_{control}$ time. $FV(i,t)$ is defined as the number of vehicles that would reach the intersection at time t in lane i, $i \in R$. Equation 3.3.6 shows $TraVol(i,t)$ in lane i with the green light at time t, and equation 3.3.7 shows $TraVol(i,t)$ in lane i with the red light at time t. Thus, traffic volume in case k can be obtained (in equation 3.3.8), and u,v are two lanes of case k. Then, the traffic volume weight can be calculated (in equation 3.3.9). A higher TV has more influence in decision-making.

$$TraVol(i,t) = RM(i,t) + \sum_{j=1}^{T_{control}} (FV(i,t+j) - DP(i,t+j)) + \Sigma L(blank) \quad (3.3.6)$$

$$TraVol(i,t) = RM(i,t) + \sum_{j=1}^{T_{control}} FV(i,t+j) \quad (3.3.7)$$

$$TraVol(k,t) = TraVol(u,t) + TraVol(v,t) \quad (3.3.8)$$

$$TV(k,t) = \frac{TraVol(k,t)}{\sum_{k \in C} TraVol(k,t)} \quad (3.3.9)$$

2) Waiting time

To calculate $WT(k,t)$, we need to obtain $AVGT_{wait}(i,t)$ first, which is defined as the average waiting time in lane i, from time t to following $T_{control}$ time. Equation 3.3.10 shows $AVGT_{wait}(i,t)$ in lane i with the green light at time t, and equation 3.3.11 shows $AVGT_{wait}(i,t)$ in lane i with the red light at time t. Thus, the average waiting time in case k can be obtained (in equation 3.3.12), and u,v are two lanes which of case k. Then, the weight of the average waiting time can be calculated (in equation 3.3.13). A longer WT has more influence in decision-making.

$$AVGT_{wait}(i,t) = 0 \quad (3.3.10)$$

$$AVGT_{wait}(i,t) = \frac{RM(i,t) \times T_{control} + \sum_{j=1}^{T_{control}} FV(i,t+j) \times (T_{control} - j)}{TraVol(i,t)} \quad (3.3.11)$$

$$AVGT_{wait}(k,t) = \frac{(AVGT_{wait}(u,t) + AVGT_{wait}(v,t))}{2} \quad (3.3.12)$$

$$WT(k,t) = \frac{AVGT_{wait}(k,t)}{\sum\limits_{k \in C} AVGT_{wait}(k,t)} \tag{3.3.13}$$

3) Hunger Level

The hunger level $HL(k,t)$ is defined to guarantee fairness. It can be determined by the number of times case k has a green light, which is represented by $N(k,t)$, $k \in C$, in equation 3.3.14. The more times the case previously got green lights, the lower its current hunger level; the fewer times the case previously got green lights, the higher its current hunger level.

$$HL(k,t) = 1 - \frac{N(k,t)}{\sum\limits_{k \in C} N(k,t)} \tag{3.3.14}$$

4) Blank Circumstance

Blanks play an important role in calculating $GLD(k,t)$. We try to minimize the frequency of the circumstance in which there is a blank at the intersection with the green light for a certain lane. In order to increase the throughput and decrease the average waiting time, we calculate how many blanks there are in each lane, and the length of each blank.

Within a $T(blank)$ time, if a sensor node cannot detect a vehicle passing through, we decide there is a blank of length $L(blank)$. $T(blank)$ should be larger than $\frac{L_{vehicle}}{speed}$, and $L(blank) = T(blank) \times speed$.

In the detection of blanks, there are three possible circumstances: where every case has a blank, or some cases have a blank, or none of them has a blank. Different circumstances have different solutions. When every case has at least one blank, we would like to give a green light with high priority to the case in which the first detected blank has the farthest distance to the intersection. In this way, a green light would be provided to let more vehicles leave. When some cases have a blank, we would decide to give a red light for these cases next. When none of them has a blank, we treat them with the same level of priority. How to determine blank length has been mentioned before.

5) Special Circumstance

Special circumstance refers to some situations where a green or red light must be activated urgently. For example, a green light must urgently be given for the lanes having ambulances or fire engines; a red light should be given for the lanes in which a traffic accident has occurred. Hence, we define $SC(k,t)$ to demonstrate these green light demands; $SC(k,t)$ is a signum function (equation 3.3.15) with only three values, 1, 0, and 1.

$$SC(k,t) = \begin{cases} 1 & \text{if high green light priority vehicle detected} \\ -1 & \text{if high red light priority vehicle detected} \\ 0 & \text{otherwise} \end{cases} \tag{3.3.15}$$

Let us suppose an urgent circumstance in which an emergency vehicle requires a specific light. If a vehicle that needs a high green light priority, such as an ambulance or fire engine, is detected by node X_{y2}, $SC(k,t)$ is equal to 1, and a green light will be provided for certain lanes next. If a vehicle that needs a high red light priority, such as that which caused a traffic accident, is detected by node X_{y2}, $SC(k,t)$ is equal to 1, to demonstrate that this circumstance has very little demand for a green light. Here, traffic accident is defined as a circumstance in which one or more vehicles do not move within a period time. We use received signal strength indication (RSSI) to detect traffic accidents. After being detected by X_{y2}, with the vehicle running, the signal strength X_{y2} received should become weaker and weaker. If the difference in the strength of the signal received before and after $T_{control}$ time is within a small range, we can decide that a traffic accident happened, and then give a red light for the case until the accident is solved. If no special circumstance occurred, $SC(k,t)$ is equal to 0.

6) Coefficient Determination

Finally, we need to determine the coefficient of each factor, which is treated as a priority. Priority for a green or red light should be assigned to these factors. Different priorities are given to different factors, sorted from high to low as special circumstance, blank circumstance, hunger level, traffic volume, and waiting time, as shown in Table 3.1. Based on the value of GLD, the case with the largest value can get a green light next.

3.3.3 Light Length Determination

The third step is to determine the length of the green light - that is, how long the green light should last for. G_{next} is defined as the length of the next green light. It

Table 3.1: Green Light Sequence Determination in an Isolated Intersection

Green Light Sequence Determination in an Isolated Intersection
Input: VDDF(d,RM(t)), the case i holding green lights.
Output: decision on which case should obtain green lights.
begin
Check special circumstance.
if there exists a case k with green light priority **then**
Assign green lights to case k next.
else
if there exists a case j with red light priority **then**
Assign green lights to case j next.
else Check blank
if all cases have a blank **then**
Find the case k with the farthest blank.
Assign green lights to case k next.
else
if at least one case has a blank **then**
Assign red lights to these cases next.
else Compute $TV(k,t)$, $WT(k,t)$, $HL(k,t)$.
if there exists a case k with $HL(k,t) \geq threshold$ **then**
Assign green lights to case k next.
else
if maximum $WT \geq WT(i,t)$ **then**
Assign green lights to the case with maximum WT next.
else
if maximum $TV \geq TV(i,t)$ **then**
Assign green lights to the case with maximum TV next.
else Assign green lights to current case i next.
end

Table 3.2: Light Length Determination in an Isolated Intersection

Light Length Determination in an Isolated Intersection
Input: next case k
Output: green duration of next case
begin
1. calculate green duration of next case $G_{nextcase}$
2. **if** $G_{nextcase} > T_{control}$
3. **then** $G_{nextcase} = T_{control}$
4. **end**
end

is equal to the time for vehicles in lanes having the next green light to go through the intersection (in equation 3.3.16), in which i, j are two lanes of the case with the next green light. If the value of G_{next} is larger than $T_{control}$, G_{next} should be equal to $T_{control}$, as shown in Table 3.2. Then, after G_{next} time, we would calculate the current traffic environment and again determine the sequence and length of the green lights.

$$G_{next} = \frac{max\{TraVol(i,t), TraVol(j,t)\}}{speed} \tag{3.3.16}$$

After determining the sequence and length of the green lights, the traffic lights will change color accordingly.

3.4 Performance Evaluation

To evaluate our scheme's performance, we conduct simulations by matlab using the proposed adaptive traffic light control scheme, comparing it with a fixed-time traffic control (FTC) and an actuated traffic control (ATC), which are based on the same random arrival rate of each lane, T, $T_{control}$, $speed$, $L_{vehicle}$ and L_{lane}.

The FTC used is the optimal fixed-time control under the given traffic parameters that can be achieved through simulations. The ATC used is from [81] because the traffic structure is the same as four directions with two permitted movements, going forward and left.

We define *volume-to-capacity* to indicate the degree to which traffic conditions are busy. Here, *volume* is calculated based on a random arrival rate in each lane. And *capacity* is defined as the number of vehicles that can be in the lane at the same time, which would be equal to $\frac{L_{lane}}{L_{vehicle}}$. We set different ranges to constrain the arrival rate to come out different *volume-to-capacity*. To describe the general traffic situation, *volume-to-capacity* here is defined as the average value of the total approaching lanes, and would be $\frac{TraVol}{capacity}$. We consider one hour as T's value, to evaluate the approaches' performance in different *volume-to-capacity*. We choose 30km/h as the car speed which is the most common speed limit in inner cities in real world.

The performance metrics include throughput-to-volume and average waiting time as a response to our objectives. The intersection throughput values were calculated using the total number of vehicles departing the intersection per unit of time, and were expressed as the number of vehicles per second. Throughput-to-volume is defined as the percentage of passing vehicles in total traffic volume. Average waiting time was calculated using the total waiting time of vehicles per vehicle. The average waiting time is defined as the average degree of waiting time of the total traffic volume.

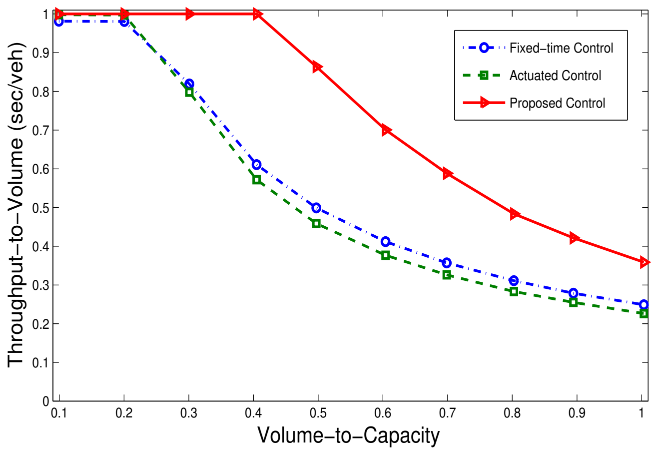

Figure 3.5: Throughput-to-volume comparisons between fixed-time control, actuated control, and the proposed adaptive approach

Fig. 3.5 presents the throughput-to-volume comparisons between an FTC, an ATC, and our proposed adaptive scheme. From the figure, we can observe that our scheme can achieve the best throughput. When the *volume-to-capacity* is lower than 0.2, all of the three methods can achieve such a good performance that they come extremely close to a 100% throughput. Among the three, the difference is very small. When the *volume-to-capacity* is in intervals of [0.2, 0.4], FTC and ATC begin to perform worse while our scheme can still obtain a throughput performance of almost 100%. The difference between our approach and the other two approaches grows larger. In particular, when the *volume-to-capacity* is 0.4, our approach achieves twice as much in through put as the other two approaches. When the *volume-to-capacity* is [0.4, 1], there are more vehicles in the traffic environment, and our scheme cannot gain the throughput ever. The throughput-to-volume obtained by our scheme begins to decrease, when the other two methods continuing to get lower throughput. The throughput of our scheme remains at almost 1.6 times higher than the other two approaches.

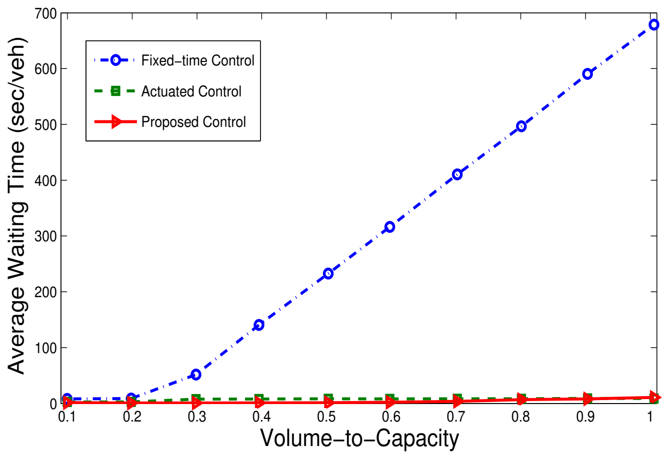

Figure 3.6: Comparison of average waiting time between fixed-time control, actuated control, and the proposed adaptive approach

Fig. 3.6 shows a comparison of the average waiting times between an FTC, an ATC, and our proposed adaptive scheme. From the figure, we can observe that our scheme can achieve the best throughput. When the *volume-to-capacity* increases, the average waiting time in FTC increases rapidly, much faster than in the other two schemes, where the average waiting time remains in intervals of [0,10] seconds.

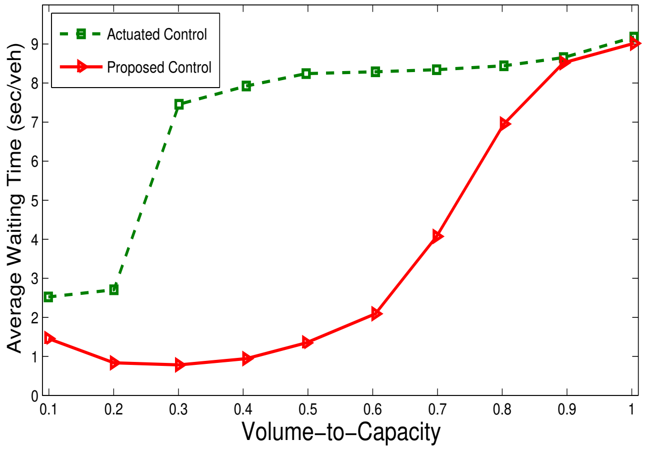

Figure 3.7: Comparison of average waiting time between actuated control and the proposed adaptive approach

Since it is hard to distinguish between ATC and our scheme in Fig. 3.6, we enlarge the performance of the two approaches in Fig. 3.7. From this figure, we can observe more clearly that our scheme can obtain a lower average waiting time. When the *volume-to-capacity* is lower than 0.2, both methods have very short waiting time of fewer than 3 seconds. When the *volume-to-capacity* is in intervals of [0.2, 0.3], the waiting time of ATC starts increasing rapidly while the waiting time of the proposed scheme remains lower than 1 second. Especially when the *volume-to-capacity* is 0.3, ATC has an average waiting time of 10 times as much as our scheme. When the *volume-to-capacity* is [0.3, 1] and there are more vehicles in the traffic environment, the waiting time in our scheme also begins to grow, but still remains lower than in ATC. When the *volume-to-capacity* is

1, there are too many vehicles, leading to an almost saturated intersection. At this time, the waiting time in our scheme approaches that in ATC, though remaining lower than 9 seconds.

Finally, from the results of the simulation, it is clear that our proposed approach can achieve higher throughput and lower average delay compared with a fixed-time traffic control and an actuated traffic control.

3.5 Summary

In this chapter, we have proposed an adaptive traffic light control scheme with the purpose of maximizing traffic throughput and minimizing average waiting time at an intersection. Our experimental results demonstrate that the proposed scheme can produce higher throughput and lower waiting time for vehicles in comparison with a fixed-time control approach and actuated control approach.

Chapter 4

ATLCMI: Adaptive Traffic Light Control for Multiple Intersections

In this chapter, we investigate the problem of adaptive traffic light control of multiple intersections using real-time traffic data collected by a wireless sensor network (WSN). Previous studies mainly focused on optimizing the intervals of green lights in fixed sequences of traffic lights and ignored the characteristics of traffic flow and special traffic circumstances. In this chapter, we propose an adaptive traffic light control scheme that adjusts the sequences of green lights in multiple intersections based on real-time traffic data, including traffic volume, waiting time, number of stops, and vehicle density. Subsequently, the optimal length of the green light can be calculated from the local traffic data and traffic conditions of neighboring intersections. The simulation results demonstrate that our scheme produces much higher throughput, lower average waiting time, and fewer number of stops, compared with the three control approaches: the optimal fixed-time control, an actuated control, and an adaptive control.

This chapter is organized as follows. In section 4.1, we briefly introduce the work. In section 4.2, we formulate the problem and define some notations. In section 4.3, we describe the proposed adaptive traffic light control scheme, which can detect and calculate the real-time traffic data, determine the sequence of green lights of multiple intersections, and then determine the optimal length of the green lights of these intersections. In section 4.4, we evaluate the performance of our scheme through simulations. In section 4.5, we summarize this chapter.

4.1 Overview

Intelligent Transportation System (ITS) refers to a system that integrates communications-based information and electronics technologies into transportation infrastructure and vehicles, to relieve traffic congestion, improve safety, reduce transportation times, and fuel consumption. The conventional surveillance methods used in ITS to detect real-time traffic data, e.g. video image processing and inductive loops detection, have several shortcomings, such as limited coverage and the high cost of implementation and maintenance [8]. Meanwhile, wireless sensor networks have the potential to provide real-time traffic data without these drawbacks. Hence, a WSN-based ITS has been proposed. Controlling traffic lights plays a key role in the system, in that an optimal traffic lights control scheme can increase traffic throughput and reduce delay by outputting different traffic signals.

A number of traffic control systems have been implemented worldwide, such as SCOOT [71] [18] and SCAT [55] [73]. Furthermore, various computational intelligence approaches have been applied to the optimization of designs for controlling traffic lights, such as Fuzzy Logic Control [52] [85], Neural Network [78] [74] [75], Genetic Algorithm [74] [41], and others. Most existing works [71] [18] [55] [73] [52] [85] [78] [74] [75] [41] consider cycle time, and split and offset optimizations with a fixed green light sequence, with the objectives of minimizing average waiting time and the number of vehicle stops. However, they pay little attention to the characteristics of traffic flow, and fail to consider special traffic circumstances, such as ambulances, fire engines, or traffic accidents.

In a previous work [93], we proposed an adaptive control scheme applied in an isolated intersection, and implemented into our WSN-based ITS testbed, iSensNet [87], which can schedule both the sequence and the length of green lights, taking into consideration discontinuous traffic flow and special traffic circumstances, to increase throughput and decrease average waiting times.

In this chapter, we extend the previous work to investigate controlling traffic lights in multiple intersections. We propose an adaptive traffic light control scheme to schedule both the sequences and lengths of green lights based on detected traffic information. The proposed scheme includes the detection of real-time traffic data, an algorithm to determine the sequence of green lights, and an algorithm to determine the length of lights.

The performance of our scheme is compared with those of an optimal fixed-time control, an actuated control, and an adaptive fuzzy control [52]. The results of the simulation show that our scheme can achieve higher throughput, lower average waiting time for vehicles, and fewer stops.

4.2 Problem Formulation and Notations

We are given a traffic network with five four-direction (east, west, north, and south) intersections, one central intersection C, and four minor intersections (C_e, C_w, C_n, C_s), which are neighbors of the central intersection. We use C_e, C_w, C_n, C_s to represent the four intersections at the east, west, north, and south of intersection C, respectively. Each direction has four lanes (see Fig. 4.1). Two of them are for vehicles to approach; we name them approaching lanes (*ALs*). Within the two *ALs*, one is for turning left and one for going forward. The other two lanes are for vehicles to leave the intersection; we name them leaving lanes (*LLs*). Meanwhile, the *LLs* of C also approach the neighbor intersection, so the *LLs* also are the *ALs* of the neighboring intersection. Each AL is controlled by a traffic light that offers two signals, red for stop and green for go. This traffic network is installed with several wireless sensor nodes to detect and monitor real-time traffic conditions.

The problem is to schedule the timing and length of the traffic lights of these intersections cooperatively, in order to improve efficiency of control under the constraint of guaranteeing fairness. Here, efficiency of control includes the network throughput, the average delay for vehicles, and the average number of stops.

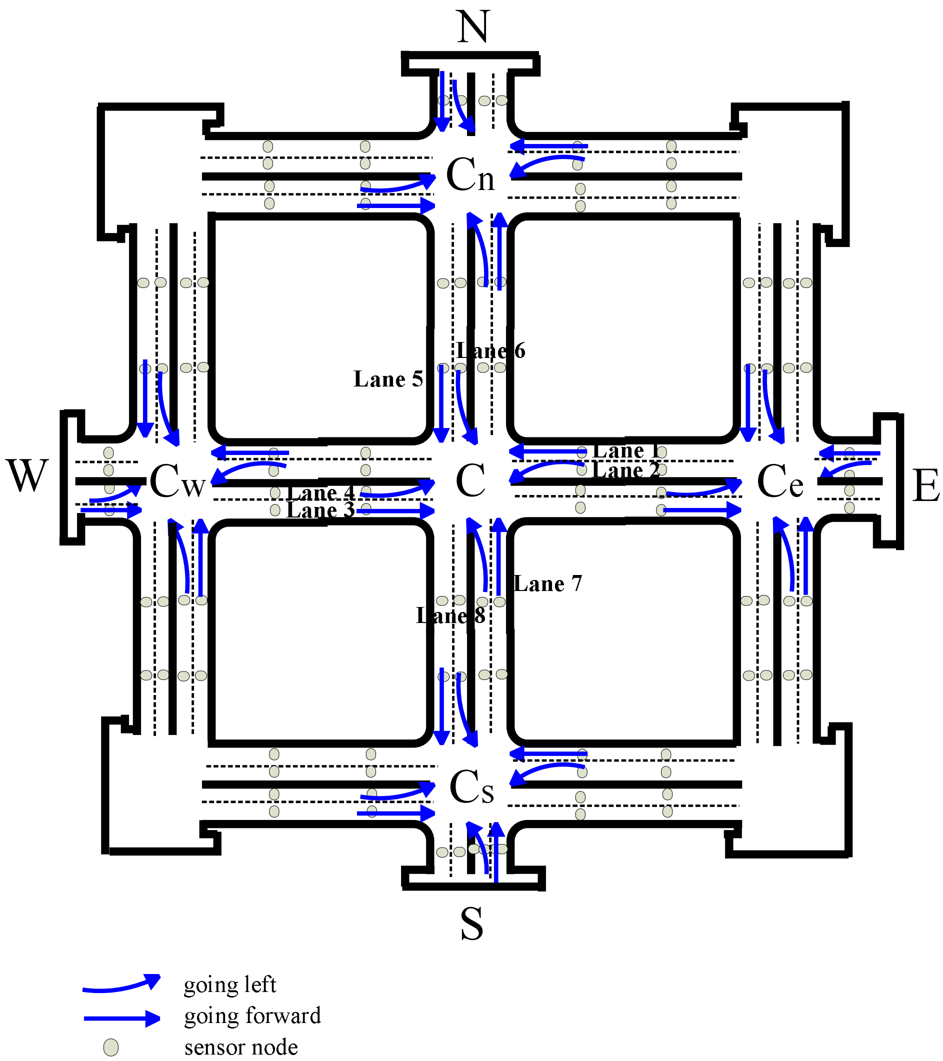

Figure 4.1: Multiple Intersections Model

Network throughput means the rate between the number of arriving vehicles and the number of departing vehicles in the traffic network. From Fig. 4.1, we see that each minor intersection contributes two entrances and two exits for this traffic network. Hence, there are a total of eight entrances and eight exits. Here, we define the vehicles entering this traffic network through these entrances as arriving vehicles, and the vehicles leaving this traffic network through these exits as departing vehicles. Average vehicle delay means the average waiting time of the vehicles in the traffic network. Average number of stops means the average number of times that the vehicles stop at each intersection to wait for the green light.

The constraints include an upper bound for the vehicle waiting time and an upper bound for the waiting time for each lane to guarantee that each vehicle will not wait too long. However, in order to reduce the cost of sensor nodes, we limit each lane to no more than two sensor nodes.

The input is the real-time detected traffic information, such as arrival rate and vehicle location. The output is a set of each intersection's control decision on which intersection should get a green light next, and then how for long the light should last.

As mentioned in our previous study [93], each intersection has a maximum of twelve different possible cases of green lights (see Fig. 3.2). Therefore, in the face of a dynamically changing traffic environment, the problem is transformed into a decision on which case should obtain a green light next in each intersection and for how long the light should last.

We define several assumptions in this problem. First, the speed is constant and all vehicles have the same speed. The specified speed in this problem can be obtained through historical data. Second, each vehicle will automatically choose the shortest path to its destination exit when enters this traffic network. This assumption is for computing the network throughput. Third, we use the same sensor node in this problem so that the sensing range of each sensor node will be the same.

To formulate the problem, we use the following notations:

$A = \{$north, south, east, west$\}$.

V: a set of intersections.

$L = \{1,2,3,...,8\}$. This is the set of lanes. 1, 3, 5, 7 mean the forwarding lanes in the direction east, west, north and south respectively; 2, 4, 6, 8 mean the left lanes in the direction east, west, north and south respectively (see Fig. 4.1).

$P = \{1,2,3,...,12\}$.

TP: total throughput.

R_s: sensing range of each sensor node.

Dst: distance between two adjacent intersections.

$AVGWT$: average waiting time.

$AVGNS$: average number of stops.

T: total time period.

t: time.

v: the element of set V.

k: the element of set P.

r: the element of set L.

$AR(v,r,t)$: number of vehicles approaching lane r of intersection v at time t.

$DP(v,r,t)$: number of vehicles in lane r leaving intersection v at time t.

$WTL(v,r,t)$: sum of the waiting time for vehicles in lane r of intersection v at time

t.

$TVL(v, r, t)$: number of vehicles in lane r of intersection v at time t.

$NSL(v, r, t)$: number of total times of vehicle stops in lane r of intersection v at time t.

Efficiency:

$$TP = \frac{\sum_{v \in V} \sum_{t=1}^{T} \sum_{i \in L} DP(v, i, t)}{T} \tag{4.2.1}$$

$$AVGWT = \frac{\sum_{v \in V} \sum_{t=1}^{T} \sum_{i \in L} WTL(v, i, t)}{\sum_{v \in V} \sum_{t=1}^{T} \sum_{i \in L} TVL(v, i, t)} \tag{4.2.2}$$

$$AVGNS = \frac{\sum_{v \in V} \sum_{t=1}^{T} \sum_{i \in L} NSL(TVL(v, i, t))}{\sum_{v \in V} \sum_{t=1}^{T} \sum_{i \in L} TVL(v, i, t)} \tag{4.2.3}$$

Equation 4.2.1 calculates the number of vehicles passing through all of the intersections within a unit of time (TP). Equation 4.2.2 calculates the average waiting time for vehicles ($AVGWT$) during time period T. Equation 4.2.3 calculates the average number of vehicle stops ($AVGNS$).

In order to maintain fairness for each case in each intersection, we define two upper bounds [93]: maximum vehicle waiting time and the upper bound of the hunger level.

4.3 The ATLCMI Scheme

In order to achieve the three objectives, we propose an adaptive traffic light control scheme, including real-time traffic detection, green light sequence determination, and light length determination. Real-time traffic detection is responsible for detecting and calculating traffic conditions in a real-time manner. Green light sequence determination is to determine which case should be assigned the next green light in each intersection based on the traffic data calculated, including traffic volume, waiting time, number of stops, hunger level, blank circumstance, and special circumstances. Light length determination is to determine the duration of the next green light in each intersection using local traffic volume and traffic conditions in neighboring intersections. It should be equal to the sum of a sufficient amount of time for the traffic volume of the next green case to leave the intersection and the minimum remaining time of the current green case of influential neighboring intersections.

4.3.1 Real-time Traffic Detection

Real-time traffic detection is carried out in order to detect real-time traffic conditions in the traffic network and calculate traffic information in real-time. In this part, three issues are considered: sensor deployment, vehicle detection, and data transmission. With regard to deployment, the relationship between R_s and Dst should be discussed first. In order to guarantee that a lane can be covered, R_s must be greater than $\frac{1}{4}Dst$. Under this circumstance, we have two discussions, one is $\frac{1}{4}Dst \le R_s \le \frac{1}{2}Dst$, the other is $R_s \ge \frac{1}{2}Dst$

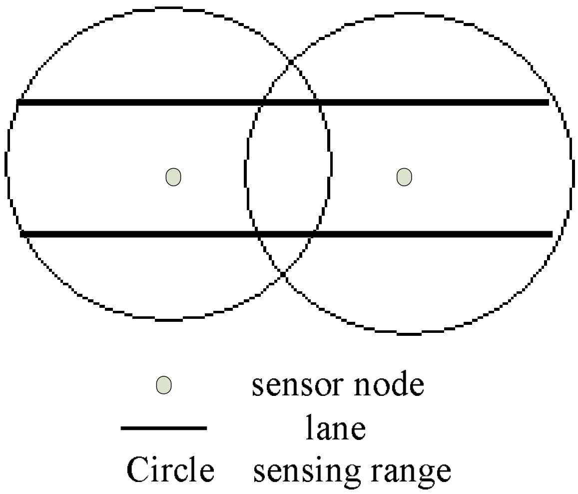

Figure 4.2: Sensor Deployment when $\frac{1}{4}Dst \le R_s \le \frac{1}{2}Dst$

When $\frac{1}{4}Dst \le R_s \le \frac{1}{2}Dst$, if only one sensor node is installed in each lane, some vehicles in the lane will out of the sensor node's sensing range. Therefore, it is necessary to install two sensor nodes in each lane, as shown in Fig. 4.2. We choose to install them at the two ends of each lane at a given distance from the intersection, so that all of the vehicles in the lanes can be detected. The given distance should be smaller than R_s and greater than $\frac{1}{4}Dst$. When a vehicle enters the AL, its type and ID can be detected by the upstream sensor node (USN). The vehicle length can be determined by the vehicle type, and the vehicle can be identified through the ID. Once detected, the USN will send the detected data, such as the arrival rate, to the intersection sensor node (ISN) to record. Meanwhile ISN will detect the departure rate in real-time. Based on the arrival rate and departure rate, the number of vehicles can be calculated in equation 4.3.1.

$$TVL(v, r, t) = max\{TVL(v, r, t-1) + AR(v, r, t) - DP(v, r, t), 0\} \tag{4.3.1}$$

As to the waiting time of each vehicle, we can calculate the total waiting time of the vehicles in each lane at time t. In the lane with the green light, the waiting time should be equal to zero; while in the lane with the red light, the waiting time of vehicles in time t should be equal to the number of vehicles in each lane at time t. When the ISN determines the $TVL(v, r, t)$, it can determine the current $WTL(v, r, t)$.

$$WTL(v, r, t) = TVL(v, r, t) \tag{4.3.2}$$

$$WTL(v, r, t) = 0 \tag{4.3.3}$$

The number of stops of each vehicle also can be calculated using the approach in [56]. If the USN detects the location of a vehicle did not change within a period Tns, we can determine that the vehicle stops within that period. Then, the times of the stops can be determined by checking whether the vehicle stops at the same lane. Once a vehicle stops in the lane, if it is in the sensing range of the USN, the USN will store the vehicle's ID and send the message to the corresponding ISN. Then, when the vehicle enters the sensing range of the ISN and stops again, the ISN will check whether it has stopped in the range of the USN before, by looking for a stored vehicle ID. If it finds this ID, the ISN will determine that the vehicle has stopped in the lane before; otherwise, the ISN will determine that this is the first time that the vehicle has stopped in the lane. Hence, the number of stops can be determined in this way. The value of Tns is constant; here, 1 second is a suitable value.

Finally, the ISN will send the real-time local traffic information to the intersection controller, including the traffic volume, the corresponding waiting time, and their number of stops.

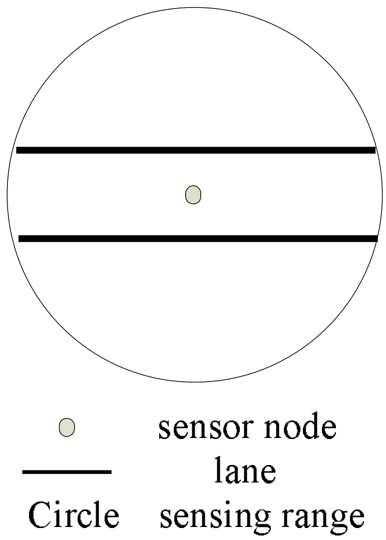

Figure 4.3: Sensor Deployment when $R_s \geq \frac{1}{2}Dst$

When $R_s \geq \frac{1}{2}Dst$, the lane can be covered by one sensor node installed at the middle of the lane, as shown in Fig. 4.3. Then, all the vehicles in the lanes can be detected. We use the USN to represent this sensor node. Similar to the circumstance when $\frac{1}{4}Dst \leq R_s \leq \frac{1}{2}Dst$, the vehicle type and ID are the items detected by the USN. Using the vehicle type and ID, the vehicle can be identified and the vehicle length can be determined. In this circumstance, the USN is responsible for detecting the real-time arrival rate and departure rate to calculate the number of vehicles.

Similarly, when the USN determines the $TVL(v, r, t)$, it can determine the $WTL(v, r, t)$, as shown in equation 4.3.2 and equation 4.3.3.

With regard to the number of stops of each vehicle, this can also be calculated using the approach in [56]. Similarly, if the USN detects that the location of a vehicle has not changed within a given period Tns, we can determine that the vehicle has stopped within that period. The times of the stops can then be determined by checking whether the vehicle is stopped at the same lane. Once a vehicle stops in the lane for the first time, the USN will store the vehicle's ID. When the vehicle stops in the lane again, the USN will check whether it has stopped before by checking the vehicle's ID. If so, the USN will determine that the vehicle has stopped in the lane before; otherwise, that is the first time that the vehicle has stopped in the lane. Hence, the number of stops can be determined.

Finally, the USN will send the real-time local traffic information to the intersection controller, including the traffic volume, the corresponding waiting time, and the number of stops.

4.3.2 Green Light Sequence Determination

Using the traffic data that has been detected and calculated, the local controller *LC* in each intersection can make a control decision about which case should get a green light next.

In each intersection, there are 12 cases that are candidates to be selected to have the green light next. We define a green light demand (*GLD*) to represent each case's demand for a green light. The case with greatest value can obtain the green light next. As in our analysis, there are six impact factors that will influence the computation of the *GLD* value: the traffic volume in each case, the average delay of the vehicles in each case, the average number of stops of each vehicle in each case, the hunger level of each case, the blank circumstance of each case, and the special circumstances of each case.

$$\begin{aligned} GLD(v,k,t) =& a_1 \times TV(v,k,t) + a_2 \times WT(v,k,t) + a_3 \times NS(v,k,t) \\ &+ a_4 \times HL(v,k,t) + a_5 \times BC(v,k,t) + a_6 \times SC(v,k,t) \end{aligned} \tag{4.3.4}$$

In equation 4.3.4, $GLD(v,k,t)$ is defined as the green light demand of case k at intersection v at time t. $TV(v,k,t)$, $WT(v,k,t)$, $NS(v,k,t)$, $HL(v,k,t)$, $BC(v,k,t)$, $SC(v,k,t)$ are defined as the weight of traffic volume, average waiting time, average number of vehicle stops, hunger level, blank circumstance, and special circumstance of case k at intersection v at time t, respectively. a_i is defined as the weight of each factor in the *GLD* computation, $i = 1,2,...,6$.

In each case, there are two lanes with green lights, we use Green Light Lane

(GLL) to represent each of them. CL is defined as a set of the corresponding lanes mapping to the case set C. Hence, $CL(k,1)$ and $CL(k,2)$ are two lanes belong to case k. We also define M to represent the direction of the cases. $M(k,i)$ is equal to $[\frac{CL(k,i)+1}{2}]$, $i = 1, 2$. $M(k,1)$, $M(k,2)$ are the directions of $CL(k,1)$, $CL(k,2)$, respectively.

$$CL = \begin{pmatrix} 1 & 3 \\ 3 & 4 \\ 2 & 4 \\ 1 & 2 \\ 5 & 7 \\ 5 & 6 \\ 6 & 8 \\ 7 & 8 \\ 3 & 6 \\ 4 & 7 \\ 1 & 8 \\ 2 & 5 \end{pmatrix}, \quad M = [\frac{CL+1}{2}] = \begin{pmatrix} 1 & 2 \\ 2 & 2 \\ 1 & 2 \\ 1 & 1 \\ 3 & 4 \\ 3 & 3 \\ 3 & 4 \\ 4 & 4 \\ 2 & 3 \\ 2 & 4 \\ 1 & 4 \\ 1 & 3 \end{pmatrix}$$

We then have a way of computing $TV(v,k,t)$, $WT(v,k,t)$, $NS(v,k,t)$, $HL(v,k,t)$, $BC(v,k,t)$, and $SC(v,k,t)$.

1) Traffic Volume Computation

To calculate $TV(v,k,t)$, we first need to obtain the corresponding $TVL(v,r,t)$, which can be calculated in equation 4.3.1. When in the lane with the red light, $DP(v,r,t)$ is equal to zero. The traffic volume in case k at intersection v can then be obtained in equation 4.3.5. $CL(k,1)$, $CL(k,2)$ are the two lanes of case k. A higher TV has more influence in the computation of GLD.

$$TV(v,k,t) = TVL(v, CL(k,1), t) + TVL(v, CL(k,2), t) \tag{4.3.5}$$

2) Waiting Time Computation

To calculate $WT(v,k,t)$, we first need to obtain the corresponding $WTL(v,r,t)$, which can be calculated in equation 4.3.2 and equation 4.3.3. The average waiting time in case k at intersection v can then be obtained in equation 4.3.6. A higher WT has more influence in the computation of GLD.

$$WT(v,k,t) = WTL(v, CL(k,1), t) + WTL(v, CL(k,2), t) \tag{4.3.6}$$

3) Number of Stops Computation

To calculate $NS(v,k,t)$, we first need to obtain $NSL(v,i,t)$. $NS(v,k,t)$ should be equal to the total number of stops of the vehicles in the two corresponding lanes (in equation 4.3.7).

$$NS(v,k,t) = NSL(v, CL(k,1), t) + NSL(v, CL(k,2), t) \tag{4.3.7}$$

4) Hunger Level Computation

To calculate $HL(v,k,t)$, we first need to obtain $N(v,k,t)$, which is defined as the times of the previous green lights of case k of intersection v at time t, so that $HL(v,k,t)$ is equal to the ratio between $N(v,k,t)$ and the total times of the previous green lights of all cases (in equation 4.3.8).

$$HL(v,k,t) = \frac{N(v,k,t)}{\sum_{k \in C} N(v,k,t)} \tag{4.3.8}$$

5) Blank Circumstance Computation

The consideration of blank circumstance relates to the discontinuous flow of traffic and the attempt to reduce the frequency of the circumstance in which there is no vehicle in the *GLL* to reach the intersection. Blank plays a rather important role in calculating $GLD(v,k,t)$. We try to minimize the frequency of the circumstance in which there is a blank at the intersection with the green light for a certain lane, which would lead to the waste of a green light. This means that, within the time period equal to the length of this blank $L(blank)$, the number of vehicles passing through the intersection is not as many as supposed, and the total waiting time of the vehicles in the other lanes would therefore increase. Thus, in order to increase the throughput and decrease the average waiting time, we try to release the blank by making the blank reach the intersection with the red light for that certain lane. We therefore need to calculate how many blanks there are in each lane, and the length of each blank.

Within a $T(blank)$ time, if a sensor node cannot detect a vehicle passing through, we decide that there is a blank of length $L(blank)$.

$$T(blank) \geq \frac{L_{vehicle}}{speed} \tag{4.3.9}$$

$$L(blank) = T(blank) \times speed \tag{4.3.10}$$

When the USN detects the blank at the upstream end, it will immediately inform the LC. The LC will then compare the possible traffic conditions at the following corresponding time, and try to select a case with the greatest need and without a blank at that time. After making the selection, the LC will give a short green. If all of the cases have a blank at that time, the LC will select the one with the shortest length, and assign the green light to that case.

6) Special Circumstance Computation

We would like to divide the problem of controlling traffic lights into two circumstances: one where some special circumstance has happened; and the other where there are no special circumstances, which means that the common circumstances prevail in scheduling the traffic lights.

As mentioned before, special circumstances refer to some situations where a green or red light must be activated urgently because of the presence of some special type of vehicle, such as an ambulance or fire engine; also, a traffic accident should receive special treatment. Hence, $SC(v, k, t)$ is a signum function (equation 4.3.11) with only three values, 1, 0, and 1.

$$SC(v, k, t) = \begin{cases} 1 & \text{if high green light priority vehicle detected} \\ -1 & \text{if high red light priority vehicle detected} \\ 0 & \text{otherwise} \end{cases} \tag{4.3.11}$$

Let us suppose an urgent circumstance in which an emergency vehicle requires a specific light. If a vehicle, such as an ambulance or fire engine, is detected by the USN, $SC(v, k, t)$ is equal to 1, and a green light will immediately be provided for certain lanes next. If a traffic accident has been detected, $SC(v, k, t)$ is equal to 1, to demonstrate that this circumstance has very little demand for a green light. If the difference in the signal strength of the USN or ISN received before and after $T_{control}$ time is within a small range, we can decide that a traffic accident has happened, and then give red light for the case until the accident solved. If there are no special circumstances, $SC(v, k, t)$ is equal to 0.

7) Coefficient Determination

Since there are so many coefficients here, we choose to simplify some to determine these coefficients a_i. We choose to compute them through simulations. In order to simplify the number of simulations and increase the processing speed, we would like to give different values to the coefficients of traffic volume (TV), average waiting time (WT), and the average number of stops (NS); and a fixed value to the coefficients of the other factors. Based on expert knowledge, TV, WT, NS are important; therefore, we define a range [0.15, 0.3] for them to adjust themselves.

After that, the GLD of each case in each intersection can be calculated, and the greatest one in each intersection is selected to obtain the next green light.

4.3.3 Light Length Determination

After determining the sequence of the green lights, the length of the green light in each intersection should be determined. $Len(v,t)$ is defined as the length of green light in intersection v at time t, which can take both local traffic conditions and neighboring traffic conditions into consideration.

First, we compute a preliminary value of the length in each intersection, based on the traffic conditions in the next green case (NGC). $Len_{pre}(v,t)$ is defined to represent it in intersection v at time t, and it should be equal to a sufficient amount of time for the vehicles in the two $GLLs$ to pass through the intersection (see equation. 4.3.12).

$$Len_{pre}(v,t) = \frac{max\{TVL(v, CL(g(v),1),t), TVL(v, CL(g(v),2),t)\}}{speed} \qquad (4.3.12)$$

Then, the offset between the adjacent intersections to get green waves is considered, which can lead to the vehicles meeting red lights as few times as possible when going through the intersections. $Len_{os}(v,t)$ is defined as a sufficient amount of time for the vehicles permitted from intersection v's neighboring intersection ($NeiInt$) to pass through intersection v. In order to guarantee fairness, we also define a maximum green light length T_{max} so that $Len(v,t)$ should be smaller than T_{max}.

$$Len(v,t) = Len_{pre}(v,t) + Len_{os}(v,t) \tag{4.3.13}$$

A. Analysis of the Neighboring Intersection's Influence

Before computing $Len_{os}(v,t)$, the influence from the neighboring intersection should be taken into account and analyzed. When Dst is smaller than $T_{control} \times speed$, the traffic conditions of neighboring intersections will have a significant amount of influence on the central intersection, which cannot be ignored. The shorter Dst has more influence from the four neighboring intersections. In each neighboring intersection, there are several possible impact factors, such as the vehicles from the neighboring intersection, the corresponding waiting time, their number of stops, remain current green light duration, and so on.

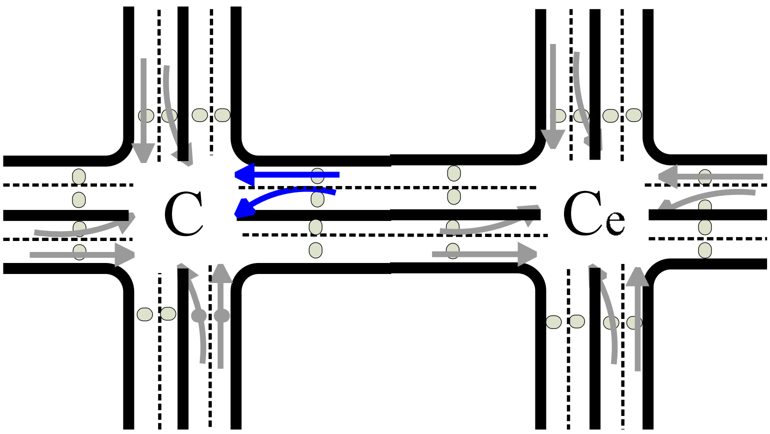

Figure 4.4: An Example of neighboring intersections

$Ne(v,t)$, $Nw(v,t)$, $Nn(v,t)$ and $Ns(v,t)$ are defined to represent the possible influence from the four $NeiInts$ of the four directions, east, west, north and south, respectively.

Let us take C_e as an example to analyze the effect of $Ne(v, t)$ from $NeiInt$ for C. In C_e, it would be possible for 12 cases to obtain the green light. Within them, 5 (see Fig. 4.5), cases 1,4,7,8,11, would allow vehicles to pass through and approach intersection C when getting a green light. We define a set of these five cases as the impact case set (ICS) and a set of the others as the non-impact case set ($NICS$). In ICS, when case 11 obtains a green light, vehicles in both $GLLs$ will approach intersection C, while in the other four cases, only one GLL will let the vehicles approach C. We define these impact $GLLs$ in $NeiInt$ s as possible impact lanes ($PILs$) for C. Hence, there are two $PILs$ for C when the green case of Ce is case 11, and there is only one PIL for C when cases 1, 4, 7, 8 have a green light in Ce.

We define a matrix NC to contain all of the cases taking PIL, including two block matrixes, NCT and NCS. NCT contains all of the cases that would take two $PILs$, and NCS contains all of the cases that would take only one PIL. The element in the i-th row of NC, NCT and NCS refers to the case in the $NeiInts$ s at the i direction of intersection C, i=1, 2, 3, 4.

$$NCS = \begin{pmatrix} 1 & 4 & 7 & 8 \\ 1 & 2 & 6 & 7 \\ 3 & 4 & 5 & 6 \\ 2 & 3 & 5 & 8 \end{pmatrix}, \quad NCT = \begin{pmatrix} 11 \\ 9 \\ 12 \\ 10 \end{pmatrix}$$

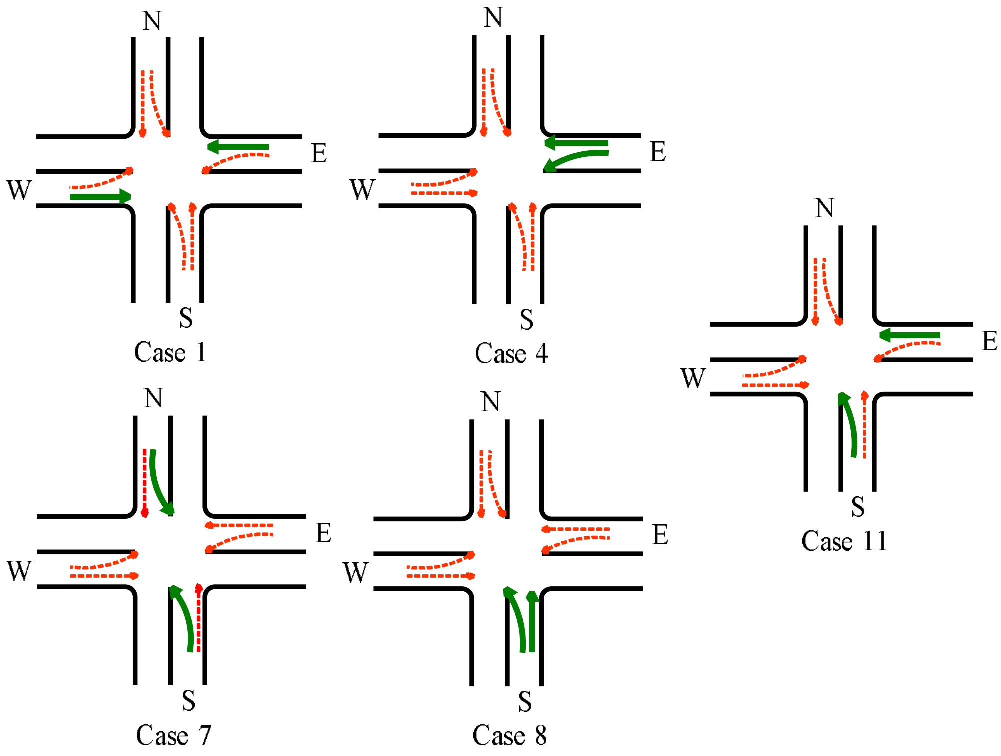

Figure 4.5: Five Impact Cases

$$NC = [NCS, NCT] = \begin{pmatrix} 1 & 4 & 7 & 8 & 11 \\ 1 & 2 & 6 & 7 & 9 \\ 3 & 4 & 5 & 6 & 12 \\ 2 & 3 & 5 & 8 & 10 \end{pmatrix}$$

On the other hand, there exist some forward-backward-lane (FBL) pairs between two adjacent intersections, which mean that vehicles in the backward lane of $NeiInt$ would approach the forward lane of the central intersection. A set F B is defined to include all possible $FBLs$. $FB(i, j)$ refers to the vehicles in the j lane of the $NeiInt$ at the i direction of intersection C.

$$FB = \begin{pmatrix} 1 & 8 \\ 3 & 6 \\ 2 & 5 \\ 4 & 7 \end{pmatrix}$$

Since at each time, each intersection always has one case to which to assign a green light, there are only two *GLLs*. We use $g(Int)$ to represent the case having a green light; *Int* refers to the intersection; it can be *C*, *EaNI*, *WeNI*, *NoNI* and *SoNI*. These two *GLLs* may be in the same direction or in different directions. When the two *GLLs* are in the same direction, if the current case of the neighboring intersection is the impact case, we can determine that only one neighboring intersection's traffic conditions would influence the central intersection; otherwise we can determine that no neighboring intersection is exerting an influence at that time. When the two *GLLs* are in two different directions, if one or more than one current case of a neighboring intersection is the impact case, we can determine that there is influence from the traffic conditions of one or more neighboring intersections; otherwise, we can determine that there is no influence from a neighboring intersection at that time.

Given the next green case candidate of the central intersection $g(v)$, which has been determined in the green light sequence determination algorithm, $CL(g(v), 1)$ and $CL(g(v), 2)$ are the two *GLLs*, $M(g(v), 1)$ and $M(g(v), 2)$ are the directions that the $CL(g(v), 1)$ and $CL(g(v), 2)$ are towards. Since each direction has two

ALs, the corresponding backward lanes of the $M(g(v), 1)$ direction neighboring intersection, in which the vehicles can pass through the neighboring intersection and enter into the ALs of the central intersection, are $FB(M(g(v), 1), 1)$ and $FB(M(g(v), 1), 2)$; and the corresponding backward lanes of the $M(g(v), 2)$ direction neighboring intersection are $FB(M(g(v), 2), 1)$, $FB(M(g(v), 2), 2)$.

On the other hand, based on the current green case of the neighboring intersections, we know the two $GLLs$, $CL(PN(M(g(v), 1)), 1)$ and $CL(PN(M(g(v), 1)), 2)$ in the $M(g(v), 1)$ direction neighboring intersection; and the two $GLLs$ in the $M(g(v), 2)$ direction neighboring intersection, $CL(PN(M(g(v), 2)), 1)$ and $CL(PN\ (M(g(v), 2)), 2)$.

Hence, we define NLS as representing the impact lanes set of the central intersection, $NLS\,(g(v), 1)$ as the impact lanes (ILs) that are the backward lanes of the two ALs in the $M(g(v), 1)$ direction of the central intersection, and $NLS\,(g(v), 2)$ as the ILs that are the backward lanes of the two ALs in the $M(g(v), 2)$ direction of the central intersection.

Based on the above analysis, we get that $NLS\,(g(v), 1)$ is equal to the *bigcap* of the backward lanes of two ALs in the M(g(v), 1) direction and the two $GLLs$ of the $M(g(v), 1)$ direction neighboring intersection (see equation 4.3.14). We use $n(NLS\,(g(v), 1))$ to represent the number of elements in $NLS\,(g(v), 1)$. From the analysis, we know that $n(NLS\,(g(v), 1)) \leq 2$; it can be 0, 1, or 2. When $n(NLS\,(g(v), 1))$ is equal to zero, the green case of the $M(g(v), 1)$ direction neighboring intersection should belong to $NICS$. When $n(NLS(g(v), 1))$ is equal to one, the green case of the $M(g(v), 1)$ direction neighboring intersection should be one element of matrix $NCS(M(g(v), 1))$. When $n(NLS(g(v), 1))$ is equal to two, the green case of the $M(g(v), 1)$ direction neighboring intersection should be one element of matrix NCT.

$$\begin{aligned} NLS(g(v), 1) =& \{FB(M(g(v), 1), 1), FB(M(g(v), 1), 2)\} \bigcap \\ & \{CL(PN(M(g(v), 1)), 1), CL(PN(M(g(v), 1)), 2)\} \end{aligned} \tag{4.3.14}$$

Similarly, $NLS(g(v), 2)$ is equal to the *bigcap* of the backward lanes of two *ALs* in the $M(g(v), 2)$ direction and the two *GLLs* of the $M(g(v), 2)$ direction neighboring intersection (see equation 4.3.15). We use $n(NLS(g(v), 2))$ to represent the number of elements in $NLS(g(v), 2)$. From the analysis, we can know that $n(NLS(g(v), 2)) \leq 2$, it can be 0, 1, or 2. When $n(NLS(g(v), 2))$ is equal to zero, the green case of the $M(g(v), 2)$ direction neighboring intersection should belong to $NICS$. When $n(NLS(g(v), 2))$ is equal to one, the green case of the $M(g(v), 2)$ direction neighboring intersection should be one element of matrix $NCS(M(g(v), 2))$. When $n(NLS(g(v), 2))$ is equal to two, the green case of the $M(g(v), 2)$ direction neighboring intersection should be one element of matrix NCT.

$$\begin{aligned} NLS(g(v), 2) =& \{FB(M(g(v), 2), 1), FB(M(g(v), 2), 2)\} \bigcap \\ & \{CL(PN(M(g(v), 2)), 1), CL(PN(M(g(v), 2)), 2)\} \end{aligned} \tag{4.3.15}$$

When the vehicles in the impact lanes *ILs* approach the central intersection, they will meet two *ALs* in a certain direction of the central intersection, the forwarding lane, and the left lane. In fact, only the vehicles in the *GLLs* of the central intersection would be able to influence the central intersection; others may not do so. Therefore, it is necessary to calculate the probabilities of which *AL* these vehicles will enter into. Using a large number of historical traffic data, the average probabilities of *FBLs* pairs can be calculated. Different time periods have different *FBLs* probabilities. Usually, the average probabilities can be achieved through historical data. In peak times, we can obtain some regular traffic patterns, such as that some *FBLs* pairs are rather busy so that the probabilities may be higher than usual, and that some *FBLs* pairs are rather free so that the probabilities may be lower than usual.

On the other hand, there are two possibilities for the two *GLLs*. First, both of them are in the same direction; second, they belong to different directions. In the first case, some of the vehicles from *ILs* would approach the forwarding lane, the rest would approach the left lane. Since both of the *ALs* are *GLLs*, all of the vehicles from *ILs* would influence the central intersection. In the second case, in one neighboring intersection, only the vehicles approaching the *GLL* would influence the intersection C. Hence, it is necessary to determine whether the two *GLLs* are in the same direction. We use another way of represent this. If the two *GLLs* are in the same direction, there is only one influential neighboring intersection (*INI*); otherwise, there are two $INIs$.

B. Offset Length Computation

After the analysis of the $INIs$ and ILs, the remaining duration of $g(M(g(v), i))$ at time t, defined as $RmT(M(g(v), i), t)$, also should be taken into account due to its significance in the offset length computation, i=1,2. If the remaining duration is large, the traffic conditions in the INI would have a significant effect on the central intersection. If the remaining duration is short, the traffic conditions in the INI would have a slight effect on the central intersection. Due to the existence of T_{max}, $Len_{os}(v, t)$ should be smaller than a threshold $Thd_{rmt}(v, t)$, which can be computed in equation. 4.3.16.

Based on the previous analysis, we can divide matters into one INI and two $INIs$ for discussion.

$$Thd_{rmt}(v, t) = T_{max} - Len_{pre}(v, t) \tag{4.3.16}$$

1. One Possible Influential Neighboring Intersection

It indicates that $g(v)$ is one of the cases of 2,4,6,8, $M(g(v), 1) = M(g(v), 2)$. We use $M(g(v), 1)$ to represent them. The current green case of INI is $g(M(g(v), 1))$, and two $GLLs$ are $CL(g(M(g(v), 1)), 1)$ and $CL(g(M(g(v), 1)), 2)$. There are three possibilities of ILs, no IL, one IL, and two ILs. Different circumstances would have different influences on the central intersection.

- None Impact Lane

In this circumstance, the current green case of INI, $g(M(g(v), 1))$, belongs to $NICS$ which means no influence from a neighboring intersection. Regardless of whether $RmT(M(g(v), 1), t)$ is greater than Thd_{rmt}, $Lenos(v, t)$ is equal to zero.

$$Len_{os}(v, t) \quad = 0 \tag{4.3.17}$$

- One Impact Lane

  In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCS(M(g(v), 1))$. $NL(g(v)$, 1) is defined as this IL. Regarding the Len_{os}, we divide it into two possibilities for discussion, when $RmT(M(g(v), 1), t) \geq Thd_{rmt}(v, t)$ and when $RmT(M(g(v)$, $1), t) < Thd_{rmt}(v, t)$ in equation 4.3.18.

$$Len_{os}(v, t) = \begin{cases} Thd_{rmt}(v, t) & \text{if } RmT(M(g(v), 1), t) \geq Thd_{rmt}(v, t) \\ RmT(M(g(v), 1), t) & \text{if } RmT(M(g(v), 1), t) < Thd_{rmt}(v, t) \end{cases} \tag{4.3.18}$$

- Two Impact Lanes

  In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCT(M(g(v), 1))$. $NL(g(v)$, 1) and $NL(g(v), 2)$ are defined as the two ILs. And $Len_{os}(v, t)$ can be computed in equation 4.3.18, as if only one IL existed.

2. Two Possible Influential Neighboring Intersections

When there are two $INIs$, this indicates that $g(v)$ is one of the cases of 1, 3, 5, 7, 9, 10, 11, 12. At this time, $M(g(v), 1) \neq M(g(v), 2)$. The next green case candidate of the two $INIs$ are $g(M(g(v), 1))$ and $g(M(g(v), 2))$ respectively. Subsequently, the two remaining green durations are $RmT(M(g(v), 1), t)$ and $RmT(M(g(v), 2), t)$. Two $GLLs$ of case $g(M(g(v), 1))$ are $CL(g(M(g(v), 1)), 1)$ and $CL(g(M(g(v), 1)), 2)$, and two $GLLs$ of case $g(M(g(v), 2))$ are $CL(g(M(g(v), 2)), 1)$ and $CL(g(M(g(v), 2)), 2)$. In each INI, there are three possibilities of ILs, none IL, one IL and two ILs, which lead to nine circumstances (see Table 4.1) for discussion.

Table 4.1: Number of Impact Lanes of the Central Intersection

Circumstance	$n(NLS(g(v), 1))$	$n(NLS(g(v), 2))$
1	0	0
2	1	0
3	2	0
4	0	1
5	1	1
6	2	1
7	0	2
8	1	2
9	2	2

- Circumstance 1

  In this circumstance, there is no IL for the central intersection, both $g(M(g(v), 1))$ and $g(M(g(v), 2))$ belong to $NICS$. There are no influence from neighbor intersection. Similarly, $Len_{os}(v, t)$ should be equal to zero, and $Len(v, t)$ should be equal to $Len_{pre}(v, t)$.

- Circumstance 2

  In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCS(M(g(v), 1))$ and $g(M(g(v), 2)) \in NICS$. There are two ALs for vehicles in $NL(k, 1)$ to approach, while there is only one GLL in the two ALs. The $Len_{os}(v, t)$ computation is similar to the circumstance of one IL of one INI (see equation 4.3.18).

- Circumstance 3

  In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCT(M(g(v), 1))$ and $g(M(g(v), 2)) \in NICS$. The $Len_{os}(v, t)$ computation is also similar to the circumstance of one IL of one INI (see equation 4.3.18).

- Circumstance 4

  This circumstance is similar to circumstance 2 when 2 $INIs$ exist. In this circumstance, $g(M(g(v), 1)) \in NICS$ and $g(M(g(v), 2))$ is an element of matrix $NCS(M(g(v), 2))$. There are two ALs for vehicles in $NL(g(v), 2)$ to approach; while there is only one GLL in the two ALs. Similarly, the $Len_{os}(v, t)$ can be computed in equation 4.3.19.

$$Len_{os}(v, t) = \begin{cases} Thd_{rmt}(v, t) & \text{if } RmT(M(g(v), 2), t) \geq Thd_{rmt}(v, t) \\ RmT(M(g(v), 2), t) & \text{if } RmT(M(g(v), 2), t) < Thd_{rmt}(v, t) \end{cases} \tag{4.3.19}$$

- Circumstance 5

In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCS(M(g(v), 1))$ and $g(M(g(v), 2))$ is an element of matrix $NCS(M(g(v), 2))$. For the offset length computations, we define $RmT(v, t)$ to represent the minimum of $RmT\ (M(g(v), 1), t)$ and $RmT(M(g(v), 2), t)$ as shown in equation. 4.3.20. $Len_{os}(v, t)$ can then be computed in equation 4.3.21.

$$RmT(v, t) = min\{RmT(M(g(v), 1), t), RmT(M(g(v), 2), t)\} \tag{4.3.20}$$

$$Len_{os}(v, t) = \begin{cases} Thd_{rmt}(v, t) & \text{if } RmT(v, t) \geq Thd_{rmt}(v, t) \\ RmT(v, t) & \text{if } RmT(v, t) < Thd_{rmt}(v, t) \end{cases} \tag{4.3.21}$$

- Circumstance 6

  In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCT(g(v), 1)$, and $g(M(g(v), 2))$ is an element of matrix $NCS(M(g(v), 2))$. The $Len_{os}(v, t)$ computation is similar to circumstance 5 in equation 4.3.21.

- Circumstance 7

  In this circumstance, $g(M(g(v), 1)) \in NICS$ and $g(M(g(v), 2))$ is an element of matrix $NCT(M(g(v), 2))$. The $Len_{os}(v, t)$ computation is similar to circumstance 3 and can be shown in equation 4.3.19.

- Circumstance 8

  In this circumstance, $g(M(g(v), 1))$ is an element of matrix $NCS(M(g(v), 1))$ and $g(M(g(v), 2))$ is an element of matrix $NCT(g(v), 2)$. The $Len_{os}(v, t)$ computation is similar to circumstance 5 (see equation 4.3.21).

- Circumstance 9

  In this circumstance, $g(M(g(v),1))$ is an element of matrix $NCT(M(g(v),1))$ and $g(M(g(v),2))$ is an element of matrix $NCT(M(g(v),2))$. The $Len_{os}(v,t)$ computation is similar to circumstance 5 (see equation 4.3.21).

Based on all of the above analysis, the next green light length in each intersection can be determined.

4.4 Performance Evaluation

To evaluate our scheme's performance, we conduct simulations using our proposed scheme, comparing it with a fixed-time traffic control (FTC), an actuated traffic control (ATC), and an adaptive fuzzy logic control (AFLC), which are based on the same random arrival rate of each lane, T, $speed$, $L_{vehicle}$ and L_{lane}.

The FTC used is the optimal fixed-time control in each intersection under the given traffic parameters, which can be achieved through simulations. The ATC used is a simplified SCOOT control scheme applied in this traffic network structure; and the AFLC used is from [52] because of the same traffic network.

Since our scheme schedules the traffic lights of multiple intersections in a distributed way, it can be applied in an arbitrary number of intersections. In this simulation, we define a traffic structure consisting of 13 inter-connected intersections. There is one central intersection connected to four major neighboring intersections. Meanwhile, each major intersection connects to its own three minor neighboring intersections. The five intersections (the central intersection and the major intersections) can schedule the traffic lights in a decentralized way.

Similar to Section 3.4, we use *volume-to-capacity* to indicate the degree to which traffic conditions are busy. Here, the average traffic volume would be equal to $\frac{TV}{capacity}$.

The performance metrics include throughput-to-volume, average waiting time, and average number of stops. The network throughput can be calculated using the total number of vehicles departing the network per unit of time, and would be expressed as the number of vehicles per second. Throughput-to-volume is defined as the ratio of departing vehicles to the total volume of traffic.

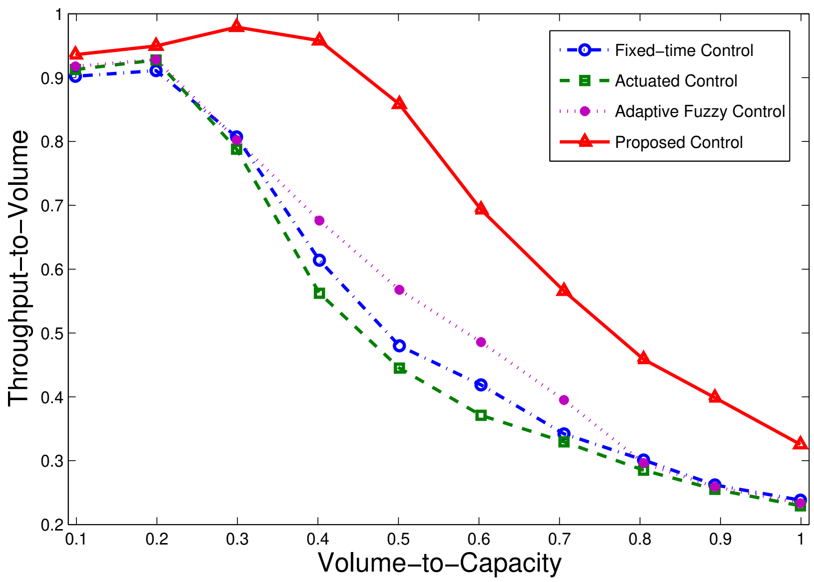

Figure 4.6: Throughput-to-volume comparisons between fixed-time traffic control, actuated traffic control, adaptive fuzzy logic control, and our proposed approach

Fig. 4.6 presents the throughput-to-volume comparisons between FTC, ATC, AFLC, and our proposed scheme. From the figure, it is apparent that our scheme can achieve the best throughput. When the *volume-to-capacity* is lower than 0.2, all of the approaches can achieve a good performance although ours obtains the best performance. When the *volume-to-capacity* is in the interval of [0.2, 0.4], the other three start to perform worse while our scheme obtains increasing throughput and achieves almost 100% throughput. The difference between our control the other three controls increases. When the *volume-to-capacity* is [0.4, 1], the traffic flow increases so that our scheme can no longer achieve the high throughput. The throughput-to-volume obtained by our scheme begins to decrease when the other three approaches continue getting lower throughput.

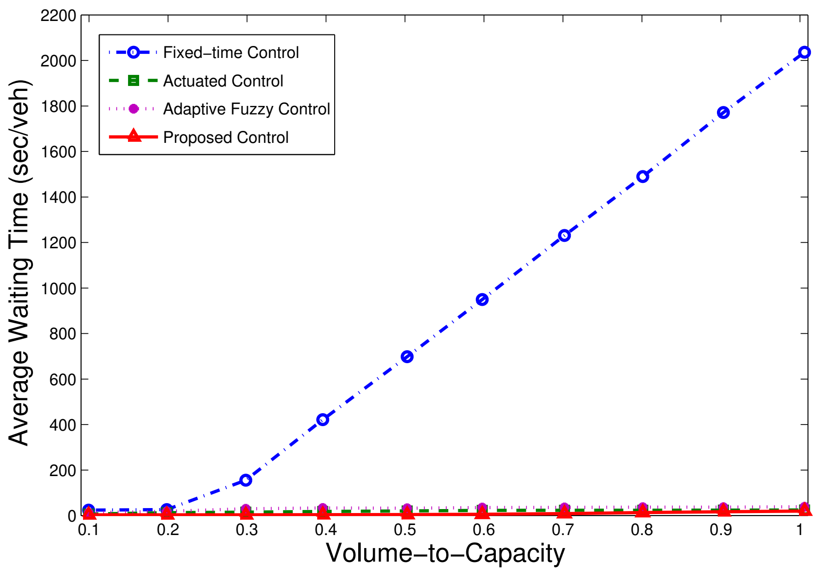

Figure 4.7: Comparisons of average waiting time between the fixed-time traffic control, actuated traffic control, adaptive fuzzy logic control, and our proposed approach

Fig. 4.7 shows comparisons of average waiting times. As seen in the figure, the average waiting time in FTC increases rapidly - much faster than the other three controls, when the volume-to-capacity is growing. Due to the difficulty of identifying ATC, AFLC, and our scheme in Fig. 4.7, we enlarge the performance of the three approaches in Fig. 4.8, making it easier to observe the comparisons of performance.

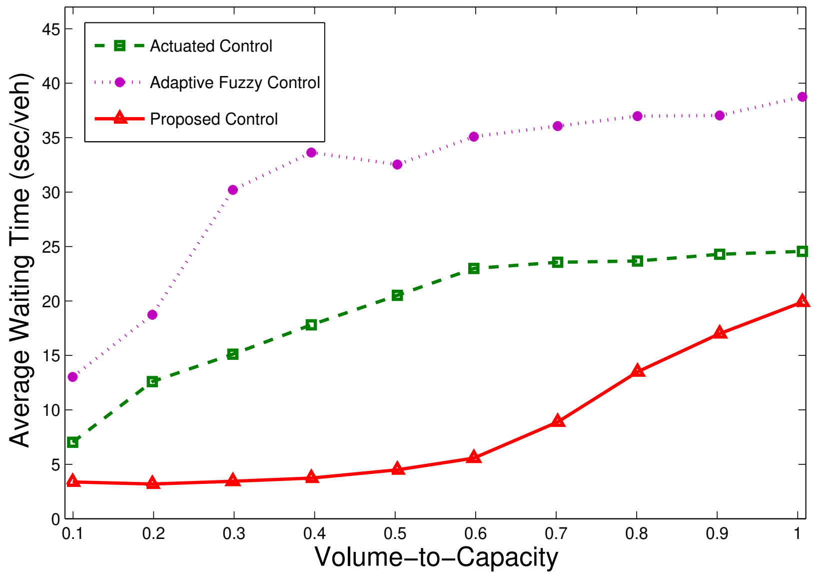

Figure 4.8: Comparisons of average waiting time between the actuated traffic control, adaptive fuzzy logic control and our proposed approach

With the *volume-to-capacity* increasing to 0.6, the average waiting time in our proposed scheme remains lower than 5 seconds, while the other two control approaches have a growing average waiting time reaching approximately 20 seconds and 35 seconds, respectively. Under this circumstance, our scheme can achieve at least a 75% reduction in average delay. As the *volume-to-capacity* increases from

0.6 to 1, the waiting time in all three approaches keeps growing, but our scheme always results in the shortest waiting time.

Fig. 4.9 shows the comparisons of the average number of stops among the four approaches. Before the *volume-to-capacity* increases to 0.4, the number of stops in the other three approaches is almost double or triple the number of stops in our scheme, which remains under 0.5. When the *volume-to-capacity* increases from 0.4 to 1, the number of stops in all of the approaches increase, although our scheme still has the fewest stops.

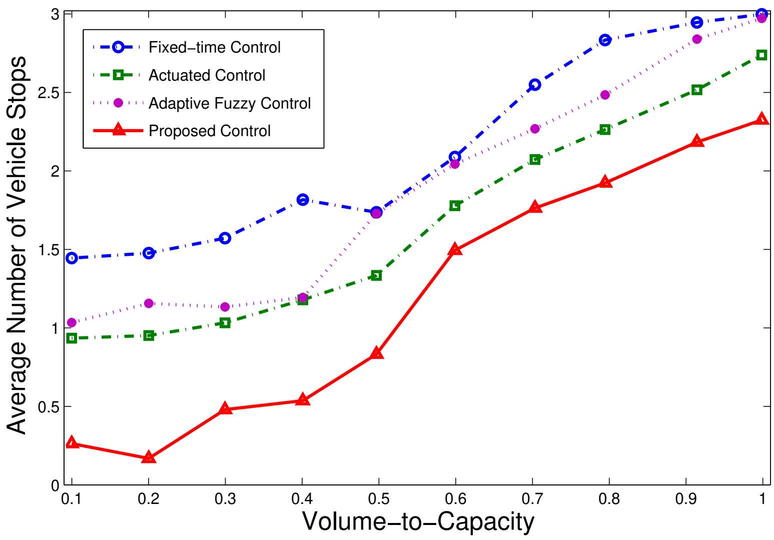

Figure 4.9: Comparisons of average number of stops between the fixed-time traffic control, actuated traffic control, adaptive fuzzy logic control, and our proposed approach.

Finally, from the results of the simulation, we can find that our proposed scheme can achieve higher throughput, lower average delay, and fewer average number of stops compared with the optimal fixed-time traffic control, an actuated traffic control, and an adaptive traffic control.

4.5 Summary

In this chapter, we have proposed an adaptive traffic light control scheme of multiple intersections with the purpose of increasing traffic throughput, reducing average waiting time, and the average number of stops. Our experimental results demonstrated that the proposed scheme could produce higher throughput, lower waiting time for vehicles, and fewer number of stops compared with the optimal fixed-time control, an actuated control, and an adaptive control.

Chapter 5

Prototype Implementation on iSensNet

In order to approach real world applications, we implement our proposed approaches into our WSN-based ITS testbed, the iSensNet (Intelligent Services with Wireless Sensor Network) platform, as shown in Fig. 5.1. The iSensNet platform is used for running various protocols or application programs in three layers: the MAC Layer, routing layer, and application layer [21]. There are two types of sensor nodes installed in this platform: stationery sensor nodes and mobile nodes. The stationery sensor nodes include detection nodes, which are installed under the platform; and roadside units, which are installed at each intersection. Mobile nodes refer to the sensor nodes installed at each mobile car model, and can communicate with the stationary nodes. Furthermore, iSensNet is able to evaluate performance in a real-time manner because of the dual-mote design which consists of two motes using different frequencies to communicate. One frequency is for normal functions of user programs, and the other frequency is for real-time performance evaluation.

Figure 5.1: iSensNet platform

On our iSensNet testbed, there are four four-leg intersections and two three-leg intersections equipped with traffic lights. Vehicles are allowed to move in the testbed in an adjustable speed. Each vehicle would choose an option (going forward or turning left) randomly when at the intersection such that the traffic environment is dynamically changing. In this way, the arrival rate of each lane of each intersection is able to be random since the vehicles move around in a random manner on the testbed.

Our iSensNet can satisfy the requirements of transportation and civil engineering applications by using a system approach. iSensNet has the following advantages. It can process complex events, which can significantly improve the usability of the WSNs and energy efficiency. Furthermore, it also can handle sampling and event processing at a very high rate, which is crucial to many practical applications.

5.1 A Simplified Intersection Model in iSensNet

Due to the physical constraints inherent in our platform, we have a simplified intersection model (see Fig. 5.2) that is different from the intersection model designed in previous chapters. In this intersection model, there are four approaches: east, south, west, and north; each approach has only one lane with going forward and left turns are permitted. Each lane of this intersection has a traffic light that can provide two signals: red (stop) and green (go).

In this scenario, we install three categories of sensor nodes because of their different functionalities: Detection Sensor Nodes (DSNs), Vehicle Nodes (VNs) and Roadside Units (RNs).

DSNs, which are installed under the two ends of each lane, are responsible for detecting whether the vehicles passing through these locations in the lanes are equipped. If any vehicle is detected, the number of vehicles in the lane can be calculated in real-time. Currently, we use an IC card.

VNs, which are installed in each vehicle, can communicate with each other and with other sensor nodes. In this way, vehicles can know the current surrounding traffic conditions to avoid possible traffic congestion.

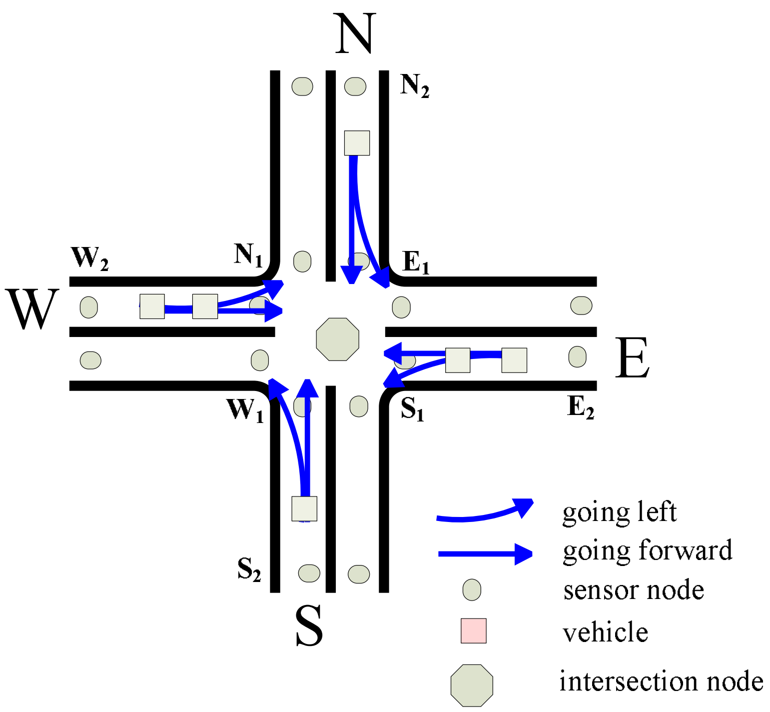

Figure 5.2: A simplified Intersection Model in iSensNet

RNs, which are installed at the intersection to control the traffic lights, can communicate with each other. RNs have two functions in this implementation. First, they are responsible for helping vehicles to register and for giving them information about current traffic lights to guarantee the reliability of the wireless communication. Second, it can make traffic light scheduling decisions based on our proposed approaches.

5.2 Work Flow

Under the consideration of maximizing the number of vehicles going through the intersection and avoiding any traffic congestion at the intersection, there exist two

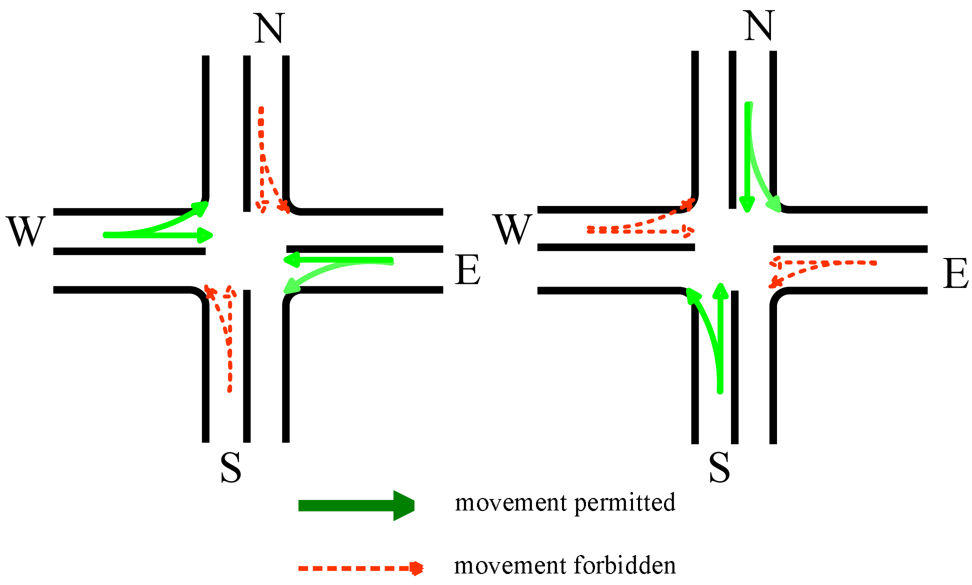

Figure 5.3: All Possible Cases of the Intersection in iSensNet

different combinations of traffic green lights, which from different our previous research work. Another difference in this model is that the vehicles run left; while in our research, we assume that all vehicles run right. Hence, there are only two cases of the corresponding configurations of traffic green lights, as shown in Fig. 5.3.

Therefore, our task is to schedule the timings and periods of the traffic lights adaptively to maximize the intersection throughput and minimize the average delay in a dynamic traffic environment. Similarly, there also exist two constraints: the maximum vehicle waiting time at the intersection and the hunger level of all cases, which could be considered as system fairness.

The **working process** is designed as follows.

Step 1. When a vehicle passes through the DSN installed below, the VN can identify the current location. Then, VN will broadcast a packet to all one hop neighbors; the packet contains its own VN id and the DSN id.

Step 2. If the vehicle is in the approaching lane of the intersection, the corresponding RN can receive the packet, and help the VN to register for its arrival. Meanwhile, the RN will send information on the current traffic lights to the vehicles so that vehicles will know whether they should stop at the intersection or can pass through the intersection.

Step 3. At the same time, the RN can process the information to calculate the number of vehicles currently in the four approaching lanes. Then, the RN makes a control decision using our proposed approach. Based on the real-time traffic volume, we can compute the two case's respective green light demand according to the same principle proposed in chapter 3. Next, the RN makes a control decision that approaches the objective while satisfying the constraints. After that, the RN will schedule the traffic lights accordingly.

With the process working when it is repeated, it seems clear that the traffic light control system run properly and adaptively using our proposed approach.

5.3 Demonstration

We define several different traffic conditions to evaluate our approaches' performance. The result show that our approach can achieve a real-time schedule and adjust the case sequence and durations adaptively.

Figure 5.4: Sequence Adjustment, before the light changes

Figure 5.5: Sequence Adjustment, after the light changes

From Fig. 5.4 and Fig. 5.5, we can observe the adjustment to the case sequence. Fig. 5.4 shows the traffic conditions before the light changes. There is only one vehicle in this approaching lane that needs a green light, while the other approaching lane has no vehicles and does not need a green light. Under this kind of traffic situation, the RN makes a control decision to assign the green light to the case with the vehicle, allowing it to pass through. Fig. 5.5 shows the green light being assigned to that certain case for a duration of 5 seconds, which is defined initially as the minimum green light length.

From Fig. 5.6 and Fig. 5.7, we can observe the adjustment in the duration of the case. Fig. 5.6 shows the traffic conditions before the light changes. There are two vehicles in this approaching lane, which needs the green light again, while the other approaching lane is not occupied by any vehicle. Under this kind of traffic situation, the RN makes a control decision to assign the green light for the case again with a longer green light length that admits the vehicles to pass through.

Figure 5.6: Length Adjustment, before the light changes

Figure 5.7: Length Adjustment, after the light changes

Fig. 5.7 shows the green light assigned to that case for a duration of 6 seconds, which is different from the initial value, in order to let more vehicles pass.

Figure 5.8: Sequence and Length Adjustment, before the light changes

Figure 5.9: Sequence and Length Adjustment, after the light changes

From Fig. 5.8 and Fig. 5.9, we can observe both the case sequence and the adjustments to the duration of the case. Fig. 5.8 shows the traffic conditions before the light changes. There are two vehicles in this approaching lane that need a green light, while the other approaching lane is not occupied by any vehicle and does not need a green light. Under this kind of traffic situation, the RN makes a control decision to assign the green light for the case with the vehicles, with a longer green light duration. Fig. 5.9 shows the green light assigned to that case for a duration of 6 seconds, which is different from the initial value in order to let the current two vehicles leave.

Apart from the above three scenarios, we also define more complex traffic conditions to evaluate the performance of our approach. A demonstration [5] with different types of traffic conditions shows that our approach is both effective and practical in our platform.

5.4 Summary

In this chapter, we implemented our approach into our iSensNet platform to evaluate the performance of the approach and investigate the possibility of applying it to the real world. Our experimental results demonstrate that our approach can deal with a number of different types of traffic conditions in an effective and practical manner with fewer average delay and more throughput than other approaches.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this chapter, we first summarize the works included in this thesis. Then, we discuss the future directions of our current work.

In this thesis, we first investigated the characteristics of WSN-based ITS, in the aspects of surveillance, communication, and traffic light control, and identified the challenging issues in the problem of optimizing the control of traffic lights. Based on the investigations, we addressed the issue of how to design adaptive traffic light control approaches for an isolated intersection and for multiple intersections.

Regarding to the traffic light control in an isolated intersection, we proposed an adaptive traffic light control scheme, with a wireless sensor network to detect and transmit real-time traffic data, to increase the throughput and decrease the average delay. First, we defined twelve green lights cases, under the constraints of a given intersection model and subject to traffic safety rules, and then made control decisions determining the sequence and length of the traffic lights based on the cases. The scheme contains three steps: real-time traffic detection, green light sequence determination, and light length determination. Real-time traffic detection involves detecting and calculating traffic information in real-time, e.g., traffic volumes, waiting time, and characteristics of traffic flow. Based on the calculated data, all of the cases will compute their green light demand, and the case with the greatest value will be assigned the next green light. The length of the light will be determined by the minimum of upper bound of the green light, and sufficient time will be given for vehicles to pass through.

On the subject of traffic lights in multiple intersections, we proposed an adaptive traffic light control scheme, using traffic data provided by a wireless sensor network. The control scheme also includes real-time traffic detection to collect traffic data in a real-time manner, green light sequence determination to decide the sequence of the traffic lights of multiple intersections using the traffic information that was collected, and light length determination to determine the length of the traffic lights of multiple intersections based on the local traffic volume and the remaining duration of the green lights in neighboring intersections.

Extensive evaluations, including simulations and implementations, were conducted to examine the performance of our proposed approaches. The results show that our objectives have been well fulfilled, and that our approaches outperform the previous approaches in terms of throughput, average delay, and average number of stops.

6.2 Future Works

Our research in this thesis mainly focused on the adaptive traffic light control in a WSN-based ITS. In future work, we hope to improve the proposed approaches and to investigate related research directions.

One issue that deserves further study is the need to change the assumption of constant speed. In our current models, the speed of all vehicles is treated as the same, which does not reflect traffic conditions in the real world. Changing the consideration of speed would make our models more dynamic and complicated, and increase the complexity of the design of our approach. How to achieve an adaptive and real-time traffic light control for vehicles with dynamic speed is worth investigating.

Controlling traffic lights in over-saturated intersections is another promising research direction. In this thesis, we only consider under-saturated traffic conditions, which is common in daily life except during the rush hours. How to improve traffic control performance in peak periods is a significant and attractive issue for research, which still needs further study.

Finally, we would like to take pedestrians into account when controlling traffic lights. In this thesis, we only take vehicles into consideration. However, most of urban traffic networks include pedestrian traffic, which cannot be ignored in the real world. Hence, the subject of controlling traffic lights with both vehicle and pedestrian traffic in mind should be studied further.

Chapter 7

Appendix

Proposition 1. Two traffic lights at most have green signals simultaneously.

Proof. At first, let us assume that there exists one case of three traffic lights having green signals at same time. Let us divide this situation into three cases.

First, the three green lights are all for forward lanes. This is obviously impossible. If one green light is in the approach west, then the approach north and south cannot have green signals; if there is a red light in the approach west, this means that green lights are in the other three approaches. This is also impossible. Hence, the first case is impossible.

Second, the three green lights are all for left lanes. This is also impossible. Similar to the first case, the second case is also impossible.

Therefore, only one case is possible, including going forward and turning left. So in the case of three green lights, there is at least one in the forward lane and one in the left lane. If the remaining green light is for the forward lane, there would be two green lights for the forward and one for the left lane. In this case, the two green lights should be in the forward lane in opposite approaches.

Then, it is not difficult to find that left turns would be unavailable, regardless of the approach. If the remaining light is for the left lane, there would be two green lights for the left lanes and one for the forward lane. This is similar to the former case. Therefore, the third case is also impossible.

Hence, at most two traffic lights can simultaneously have green signals. □

Definition 1. Let $F(x)$, $L(x)$ be admission for the movement going forward and turning left in direction x, respectively. $x \in I$.

Proposition 2. There exist twelve combinations of movements taking into consideration maximum throughput and congestion avoidance.

Proof. Let us observe going forward in the approach north $F(n)$. Three movements are available for it at same time, $F(s)$, $L(n)$, $L(e)$. Then, let us observe going left in the approach north $L(n)$. Three movements are available for it at same time, $F(n)$, $F(w)$, $L(s)$. A similar situation can be seen in the other three approaches. Meanwhile, the movements going left have been counted twice: one is in its own calculation, the other is together with the other available movements of going forward. Similarly, therefore, there exist $\frac{6\times4}{2} = 12$ cases. □

Bibliography

[1] http://en.wikipedia.org/wiki/Intelligent_transportation_system.

[2] http://en.wikipedia.org/wiki/Sensor_node.

[3] http://en. wikipedia. org/wiki/Wireless_Sensor_Network.

[4] http://en.wikipedia.org/wiki/Passive_infrared_Sensor.

[5] *Adaptive Traffic Light Control demo in iSensNet testbed [Online].* Available: http://imc.comp.polyu.edu.hk/isensnet/doku. php?id= demonstrations#Adaptive_Traffic_Light_Control.

[6] *Brief introduction to Intelligent Transportation System, ITS*. http://www.freeway.gov.tw/UserFiles/File/Traffic/A1%20Brief%20introduction %20to%20Intelligent%20Transportation%20System,%20ITS.pdf.

[7] User guide to transyt, version 8. In *TRRL Report LR888, Transport and Road Research Laboratory*, 1980.

[8] A summary of vehicle detection and surveillance technologies use in intelligent transportation systems. The Vehicle Detector Clearinghouse, 2007.

[9] M. M. Abbas. *A Real Time Offset Transitioning Algorithm for Coordinating Traffic Signals*. Ph.d. dissertation, Department of Civil Engineering, Purdue University, 2001.

[10] B. Abdulhai, R. Pringle, and G. J. Karakoulas. Reinforcement learning for true adaptive traffic signal control. In *Journal of Transportation Engineering*, volume 129, pages 278–285, 2003.

[11] Federal Highway Administration. Traffic control systems handbook. report no. fhwa-sa-95-032.

[12] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. In *IEEE Communication Magazine*, volume 40, pages 102–114, 2002.

[13] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. In *Elsevier Computer Networks*, volume 38, pages 393–422, 2002.

[14] H. R. Berenji and P. Khedkar. Learning and tuning fuzzy controllers through reinforcement. In *IEEE Transactions on Neural Networks*, volume 3, pages 724–740, 1992.

[15] B. H. Bing and A. Carter. Scoot: The worlds foremost adaptive traffic control system. In *Traffic Technology International*, 1995.

[16] E. Bingham. Reinforcement learning in neurofuzzy traffic signal control. In *European Journal of Operation Research*, volume 131, pages 232–241, 2001.

[17] K. R. Bisset and R. L. Kelsey. Simulation of traffic flow and control using fuzzy and conventional methods. In *Fuzzy Logic and Control: Software and Hardware Applications*, 1993.

[18] R. D. Bretherton. Current developments in scoot: Version3. In *Transportation Research Record*, number 1554, page 48C52, 1996.

[19] R. D. Bretherton, G. TBowen, P. G. Burton, and K. Wood. The use of scoot for traffic management. In *IEEE Second International Conference on Road Traffic Control*, 1986.

[20] N. Bulusu and S. Jha. *Wireless Sensor Networks*. Boston, MA, Artech House, 2005.

[21] J. Cao, H. Wu, X. Liu, and Y. Lai. isensnet: an infrastructure for research and development in wireless sensor networks. *Frontiers of Computer Science in China*, 4(3):339–353, 2010.

[22] W. Chen, L. Chen, Z. Chen, and S. Tu. Wits: A wireless sensor network for intelligent transportation system. In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, 2006.

[23] X. Chen and Z. Shi. Real-coded genetic algorithm for signal timings optimization of a signal intersection. In *Proceedings of the First International Conference on Machine Learning and Cybernetics*, 2002.

[24] X. Cheng and Z. Yang. Intelligent traffic signal control approach based on fuzzy-genetic algorithm. In *Fifth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD '08*, pages 221 – 225, 2008.

[25] S. Y. Cheung and P. Varaiya. Traffic surveillance by wireless sensor networks: Final report. In *California PATH Research Report, UCB-ITS-PRR-2007-4*, January 2007.

[26] S. Chiu. Adaptive traffic signal control using fuzzy logic. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 1992.

[27] S. Chiu and S. Chand. Adaptive traffic signal control using fuzzy logic. In *Proceedings of 2nd IEEE International Conferences on Fuzzy Systems*, pages 1371–1376, 1993.

[28] C. Chou and J. Teng. A fuzzy logic controller for traffic junction signals. In *Information Sciences*, volume 143, page 73C97, 2002.

[29] M. C. Choy, R. L. Cheu, D. Srinivasan, and F.Logi. Real-time coordinated signal control using agents with online reinforcement learning. In *82nd Annual Meeting of the Transportation Research Board*, Washington, D.C., 2002.

[30] M. C. Choy, D. Srinivasan, and R. L. Cheu. Cooperative, hybrid agent architecture for real-time traffic control. In *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, volume 33, pages 597–607, 2003.

[31] I. Day. Scoot-split, cycle and offset optimization technique. In *http://www.signalsystems.org.vt.edu/documents/July1998AnnualMeeting/scoot.pdf*.

[32] C. DiTarnato and V. Mauro. Utopia. In *Control, Computers, Communications and Transportation (CCCT 89)*, 1989.

[33] F. Donati, V. Mauro, G. Roncolini, and M. Vallauri. A hierarchical-decentralized traffic light control system. In *Proceedings of IFAC 9th Triennial World Congress*, 1984.

[34] S. M. Elahi, A. E. Radwan, and K. M. Goul. Knowledge-based system for adaptive traffic signal control. In *Transportation Research Record: Journal of the Transportation Research Board*, number 1324, pages 115–122, 1987.

[35] European Telecommunications Standards Institute (ETSI). *Intelligent Transportation Systems*. http://www.etsi.org/WebSite/technologies /Intelligent-TransportSystems.aspx.

[36] M. D. Foy, R. F. Benekohal, and D. E. Goldberg. Signal timing determination using genetic algorithms. In *Transportation Research Record 1365, Transportation Research Board*, pages 108–115, 1992.

[37] G. R. Gajjar, S. A. Khaparde, P. Nagaraju, and S. A. Soman. Application of actor-critic learning algorithm for optimal bidding problem of a genco. In *IEEE Transactions on Power Systems*, volume 18, pages 11–18, 2003.

[38] N. H. Gartner. Opac: A demand-responsive strategy for traffic signal control. In *Transportation Research Record*, number 906, pages 75–81, 1983.

[39] D. E. Goldberg. Genetic algorithms in search, optimisation and machine learning. 1989.

[40] M. A. Hadi and C. E. Wallace. Hybrid genetic algorithm to optimize signal phasing and timing. In National Research Council, editor, *Transportation Research Record 1421, Transportation Research Record*, page 104C112, Washington, D.C., 1993. National Academy Press.

[41] M. A. Hadi and C. E. Wallace. Optimization of signal phasing and timing using cauchy simulated annealing. In National Research Council, editor, *Transportation Research Record 1456, Transportation Research Record*, page 64C71, Washington, D.C., 1994. National Academy Press.

[42] D. Heidemann. Queue length and delay distributions at traffic signals. In *Transportation Research B 28(5)*, pages 377–389, 1994.

[43] D. Heidemann. A queueing theory approach to speed-flow-density relationships. In *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 103–118, 1996.

[44] J. J. Henry, J. L. Farges, and J. Tuffal. The prodyn real time traffic algorithm. In *Proceeding of 4th IFAC/IFORS Conf. Control Transportation Systems*, pages 305–309, 1983.

[45] P. B. Hunt. A traffic responsive method of coordinating signals. In *Report No. LR1014. Transport and Road Research Laboratory*, 1981.

[46] P. B. Hunt, D. L. Robertson, and R. D. Bretherton. The scoot on-line traffic signal optimization technique. In *Traffic Engineering Control*, volume 23, page 190C192, 1982.

[47] J. Jeong. *Wireless Sensor Networking for Intelligent Transportation Systems*. PhD thesis, December 2009.

[48] R. Kay and F. Mattern. The design space of wireless sensor networks. In *IEEE Wireless Communications*, volume 11, December 2004.

[49] L. A. Klein. Data requirements and sensor technologies for its. 2001.

[50] L. A. Klein, D. Gibson, and M. K. Mills. Traffic detector handbook. In *FHWA-HRT-06-108*, 2006.

[51] G. Leduc. Road traffic data: Collection methods and applications. In *JRC Technical Notes*, 2008.

[52] J. H. Lee, K. M. Lee, and H. Leekwang. Fuzzy controller for intersection group. In *Proceedings of International IEEE/IAS Conference on Industrial*

Automation and Control: Emerging Technologies, page 376C382, Taipei, Taiwan, May 1995.

[53] R. Li, J. Li, and H. Lu. multi-layer traffic signal control model based on fuzzy control and genetic algorithm. In *Ameriacan Society on Civil Engineering, Part of: Applications of Advanced Technology in Transportation*.

[54] F. B. Lin. Comparative analysis of two logics for adaptive control of isolated intersections. In *Transportation Research Record: Journal of the Transportation Research Board*, number 1194, pages 6–14, 1988.

[55] P. R. Lowrie. The sydney co-ordinated adaptive traffic systemcprinciples, methodology, algorithms. In *Proceedings of the International Conference on Road Traffic Signaling*, pages 67–70, 1982.

[56] H. Lu, S. Zhang, X. Liu, and X. Lin. Vehicle tracking using particle filter in wi-fi network. In *IEEE 72nd Vehicular Technology Conference, Ottawa*, 2010.

[57] Z. Lu and P. Hao. Bi-system fuzzy control for single intersections. *Information and Control*, 2001.

[58] P. Mirchandani and L. Head. A real-time traffic signal control system: Architecture, algorithms, and analysis. In *Transportation Research Part C*, volume 9, pages 415–432, 2001.

[59] T. M. Mitchell. Machine learning. In *McGraw-Hill*, 1997.

[60] Y. S. Murat and E. Gedizlioglu. A fuzzy logic multi-phased signal control model for isolated junctions. In *Transportation Research Part C*, volume 13, pages 19–36, 2005.

[61] M. Nakatsuyama, H. Nagahashi, and N. Nishizuka. Fuzzy logic phase controller for traffic junctions in the one-way arterial road. In *Proceedings of the IFAC Ninth Triennial World Conference*, 1984.

[62] J. Niittymaki. Installation and experiences of field testing a fuzzy signal controller. *European Journal of Operation Research*, 131, 2001.

[63] J. Niittymaki and M. Pursula. Signal control using fuzzy logic. In *Fuzzy Sets and Systems*, volume 116, 2000.

[64] J. Niittymaki and E. Turunen. Traffic signal control on similarity logic reasoning. In *Fuzzy Sets and Systems*, volume 133, 2003.

[65] L. E. Owen and C. M. Stallard. Rule-based approach to real-time distributed adaptive signal control. In *Transportation Research Record: Journal of the Transportation Research Board*, number 1683, pages 95–101, Washington, D.C., 1995.

[66] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsoalos, and Y. Wang. Review of road traffic control strategies. In *Proceedings OF The IEEE*, volume 91, December 2003.

[67] C. P. Pappis and E. H. Mamdani. A fuzzy logic controller for a traffic junction. In *IEEE Transactions on Systems, Man and Cybernetics*, 1977.

[68] F. Ren, N. Huang, and C. Lin. Wireless sensor networks. In *Journal of Software*, volume 14, pages 1282–1291, 2003.

[69] A. Rhodes, D. M. Bullock, J. Sturdevant, Z. Clark, and Jr. D. G. Candey. Evaluation of stop bar video detection accuracy at signalized intersections. In *84th Annual Meeting, Transportation Research Board*, 2005.

[70] D. I. Robertson and R. D. Bretherton. Optimal control of an intersection for any known sequence of vehicle arrivals. In *Proceedings of the 2nd IFAC-IFIP-IFORS Symposium of Traffic Control and Transportation Systems*, pages 3–17, 1974.

[71] D. I. Robertson and R. D. Bretherton. Optimizing networks of traffic signals in real timełthe scoot method. In *IEEE Trans. Veh. Technol.*, volume 40, pages 11–15, 1991.

[72] H. Salem and N. Mohamed. Middleware challenges and approaches for wireless sensor networks. In *IEEE Distributed Systems Online*, volume 7, 2006.

[73] A. G. Sims and K. W. Dobinson. The sydney coordinated adaptive traffic (scat) system philosophy and benefits. In *IEEE Trans. Veh. Technol.*, volume 29, pages 130–137, 1980.

[74] D. Srinivasan and M. C. Choy. Cooperative multi-agent system for coordinated traffic signal control. In *IEE Proceedings - Intelligent Transport Systems*, volume 153, pages 41–50, 2006.

[75] D. Srinivasan, M. C. Choy, and R. L. Cheu. Neural networks for real-time traffic signal control. In *IEEE Transactions on Intelligent Transportation Systems*, volume 7, pages 261–272, 2006.

[76] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. In *The MIT Press*, 1998.

[77] T. Thorpe. Vehicle traffic light control using sarsa. Master thesis, Department of Computer Science, Colorado State University, 1997.

[78] T. L. Thorpe. Vehicle traffic light control using sarsa. In *www.cs. colostate. edu/ anderson/pubs/thorpems.ps.gz.*, December 2006.

[79] T. L. Thorpe and C. Andersson. Traffic light control using sarsa with three state representations. In *Technical report, IBM corporation*, 1996.

[80] A. Tiwari, P. Ballal, and F. L. Lewis. Energy-efficient wireless sensor network design and implementation for condition-based maintenance. In *ACM Transactions on Sensor Networks (TOSN)*, volume 3, March 2007.

[81] M. B. Trabia, M. S. Kaseko, and M. Ande. A two-stage fuzzy logic controller for traffic signals. In *Transportation Research Part C*, 1999.

[82] A. Tzes, W. R. McShane, and S. Kim. Expert fuzzy logic for traffic signal control for transportation networks. In *Proceeding of Institute of Transportation Engineers 65th Annual Meeting*, pages 154–158, 1995.

[83] W. Wei and Y. Zhang. Fl-fn based traffic signal control. In *Proceeding of FUZZ-IEEE*, volume 1, pages 296–300, 2002.

[84] W. Wei, Y. Zhang, J. B. Mbede, Z. Zhang, and J. Song. Traffic signal control using fuzzy logic and moga. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1335 – 1340, 2001.

[85] R. J. WEILAND and L. B. PURSER. Intelligent transportation systems. In *A5009: Committee on Intelligent Transportation Systems*, 2000.

[86] M. Wiering, J. v. Veenen, J. Vreeken, and A. Koopman. Intelligent traffic light control. In *Techinical Report UU-CS-2004-029*, July 2004.

[87] H. Wu, J. Cao, and W. Chen. Dynamic collaborative event processing in wireless sensor networks. In *Technical Report*. IMCL Lab, 2010.

[88] Y. Xie. *Development and Evaluation of an Arterial Adaptive Traffic Signal Control System Using Reinforcement Learning*. Ph. d. dissertation, Department of Civil Engineering, Texas A&M University, December 2007.

[89] L. A. Zadeh. Fuzzy sets. In *Information and Control*, volume 8, pages 338–353, 1965.

[90] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 28–44, 1973.

[91] L. Zhang, H. Li, and P. D. Prevedouros. Signal control for oversaturated intersections using fuzzy logic. In *84th Annual Meeting of the Transportation Research Board*, 2005.

[92] W. Zhang, B. Wu, and W. Liu. Anti-congestion fuzzy algorithm for traffic control of a class of traffic networks. In *Proceedings of IEEE International Conference on Granular Computing, GRC 2007*, page 124C128, San Jose, California, 2007.

[93] B. Zhou, J. Cao, X. Zeng, and H. Wu. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In *IEEE 72nd Vehicular Technology Conference, Ottawa*, 2010.