Dynamic Traffic Light Optimization and Control System using Model-Predictive Control Method

You-Ren Chen

A Thesis Submitted to the
Institute of Computer Science and Information Engineering of
National Chung Cheng University for the
Degree of Master in
Computer Science and Information Engineering

June 2015

摘要

隨著近年來汽機車的數量逐漸增加，越來越多的交通議題受到重視，其中交通壅塞是一個重要的問題。然而現今的交通系統大多使用固定時制的紅綠燈設定，當面臨交通事故、天然災害、突發事件等狀況時，往往無法藉由動態調整紅綠燈的方式快速應變並紓解交通。但另一方面，隨著科技的進步，我們也可利用感測器、相機等設備蒐集到越來越多的交通數據，諸如車流量、速度、等待時間。

藉由以上兩個特點，我們提出了一套基於模型預測控制的交通號誌優化方法 (MPC-based Traffic Light Control System，簡稱 MTLCS)。這個方法包含兩個主要模型，分別為車流量預測模型及號誌優化模型，我們希望透過歷史交通資訊預測未來車流量，藉由 MPC 的策略優化號誌時間後，能針對未來交通狀況設置合適的號誌時間，並且能在短時間內視交通變化動態調整號誌時間，迅速因應即時的交通狀況，以達到改善交通壅塞的目標。

實驗結果顯示，我們的車流量預測模型平均絕對百分比誤差 (MAPE) 可低於 12%，與固定時制號誌系統相比，我們的交通號誌優化方法降低了 29.70% 的平均等待率，以及 26.93% 的平均等待時間。

關鍵字： *智慧交通、模型預測控制、交通號誌、降低壅塞、車流量預測*

Abstract

In recent years, with the increase in number of vehicles, more and more traffic issues are becoming the focus of attention world wide. One important problem is traffic congestion. However, most traffic systems still use fixed-time setting for a very long cycle. These systems cannot dynamically adjust traffic light timing in response to unexpected situations such as traffic accidents, natural calamities, or sudden incidents. On the other hand, with advances in technology, traffic data such as traffic volume, speed, and waiting time can now be gathered by sensors or cameras.

Due to the above two observation, a novel MPC-based Traffic Light Control System (MTLCS) is proposed. This method contains two main models, including traffic flow prediction model and traffic light optimization model. Historical traffic data is used to predict future traffic volumes. An MPC-based traffic light optimization method is proposed to obtain appropriate time settings that can reduce overall congestion. Our method also has the ability to dynamically adjust traffic light timings. It can rapidly respond to real-time traffic conditions to reduce traffic congestion.

Experiments show that the Mean Absolute Percentage Error (MAPE) of our traffic flow prediction model is less than 12%. Using the proposed MPC-based optimization method, the average waiting rate is reduced by 29.70% and the average waiting time is reduced by 26.93%, when compared with the fixed-time traffic light control system.

Keywords: *Smart Traffic*, *Model Predictive Control*, *Traffic Light*, *Congestion Reduction*, *Traffic Flow Prediction*

Contents

1 Introduction **1**

1.1 Background 3

1.1.1 Architecture of Intelligent Transportation System 3

1.1.2 Countermeasures for Traffic Congestion 3

1.1.3 Traffic Light Control System 5

1.1.4 Example 6

1.2 Motivation 8

1.3 Goal 8

1.4 Thesis Organization 9

2 Related Work **10**

2.1 Researches on MPC 10

2.2 Researches on Prediction Models 13

2.3 Researches on Optimization Methods 16

2.4 Researches on Traffic Simulators and Traffic Light Control Systems 20

3 Preliminaries **23**

3.1 Terminology 23

3.2 Problem Formulation 26

3.3 Parameter 26

3.3.1 Prediction Model . . . 26

3.3.2 Optimization Method . . . 27

3.3.3 MPC-based Control Method . . . 27

3.4 Assumptions . . . 28

3.5 Traffic Simulator . . . 28

4 MPC-based Traffic Light Control System **32**

4.1 Prediction Model Design . . . 34

4.1.1 Training Method for Adjusting Weights in BPNN . . . 34

4.1.2 Traffic Flow Prediction in BPNN . . . 41

4.2 Optimization Method Design . . . 43

4.2.1 Parameter Settings and Operations in GA . . . 43

4.2.2 Fitness Function Design in GA . . . 47

4.2.3 Green Signal Time Calculation . . . 49

4.3 MPC-based Control Method Design . . . 52

5 Experiments **55**

5.1 Experiment Setup . . . 55

5.1.1 Experimental Environment . . . 55

5.1.2 Testing Platform and Data . . . 56

5.2 Experimental Results . . . 58

5.2.1 Traffic Flow Prediction Model . . . 58

5.2.2 Traffic Light Optimization Method . . . 67

6 Conclusions and Future Work **73**

Bibliography **75**

List of Figures

1.1 Total number of registered vehicles in Taiwan, 1998-2014. 2

1.2 Intelligent Transportation System (ITS) architecture [1]. 4

1.3 Traffic flow in Zhongshan Rd, Kaohsiung City, 7am-10pm, 2005/09/23. 7

1.4 Proposed traffic light control and optimized framework. 8

2.1 Scheme of Model Predictive Control (MPC). 11

2.2 Flow diagram of Model Predictive Control (MPC). 12

2.3 Flow diagram of traffic prediction model. 14

2.4 Traffic light control loop. 18

3.1 Simple diagram of common traffic elements. 25

3.2 Our traffic simulator. 30

4.1 MTLCS basic framework. 33

4.2 Diagram of traffic flow statistics. 34

4.3 BPNN prediction model framework design. 35

4.4 The structure of BPNN. 37

4.5 An example of BPNN with initial inputs/outputs and weights. . . . 38

4.6 An example of BPNN with hidden layer and output layer calculations. 39

4.7 An example of BPNN with error calculation and weights updating. 40

4.8 GA optimization method framework design. 44

4.9 Diagram of intersections relation. 46
4.10 Diagram of fitness function. 48
4.11 Post processing after GA. 50
4.12 MPC-based control method framework design. 53
4.13 Diagram of MPC-based traffic light control method. 54

5.1 Testing map used in our experiment. 56
5.2 Testing traffic flow sample used in our experiment. 58
5.3 Trend chart of MAPE using different number of input nodes. 61
5.4 Trend chart of training errors and MAPE using different number of training samples. 63
5.5 Training errors of training cycles from 1 to 20. 65
5.6 Training errors of training cycles from 1 to 100,000. 65
5.7 The comparison between actual flow and prediction flow. 67
5.8 Considered future time slots for MPC method. 68
5.9 IAWR results compared with other two systems. 69
5.10 Comparison chart of IAWR results of 6 intersections. 70
5.11 Waiting time results compared with two other systems. 71
5.12 Comparison chart of waiting time of 4 divided time sections. 72

List of Tables

1.1 Example of traffic light control logic. 6

1.2 Average waiting data using fixed time traffic light setting. 7

2.1 Some applications using MPC model in traffic domain. 13

2.2 Researches on prediction models. 17

2.3 Researches on traffic light optimization methods. 20

2.4 Summary of traffic simulation program features. 22

3.1 Simulation statistics for an approach. 31

3.2 Simulation statistics for an intersection. 31

4.1 Updated weights and outputs of training model. 41

4.2 Chromosome design. 45

4.3 Coefficients in fitness calculation. 49

5.1 Experimental environment. 55

5.2 Vehicle expected values (λ) of Poisson distribution in the entry points. 57

5.3 Parameter settings of number of input nodes experiment. 60

5.4 MAPE of different number of input nodes in BPNN. 61

5.5 Parameter settings of training sample experiment. 62

5.6 Training errors and MAPE of different number of training samples. 63

5.7 Parameter settings of training cycle experiment. 64

5.8 Training errors of different number of training cycles from 10 to 100. 65

5.9 Parameter settings of final prediction model experiment. 66

5.10 IWAR results and reduced percentages of 6 intersections. 70

5.11 Waiting time results and reduced percentages of 4 divided time sections. 72

Chapter 1

Introduction

In recent years, with the advances in technology and urbanization in a lot of countries, more and more people have their own cars or motorcycles. Figure 1.1 shows the total registered vehicle statistics in Taiwan [2]. The public transportation systems are also rapidly growing. Traffic issues have become very important problem nowadays.

The traffic issues involve many domains. Following are some basic and common topics. In the road infrastructure part, such as road type, communication between primary roads and secondary roads, and structural design of intersections. In the traffic situation part, camera monitoring, sensor applications, traffic light control systems and travel routing plans are some popular topics. On the side of public transport, there are some problems such as suitable transport type in each area, transport scheduling, route design, and so on. Other famous topics like green transport, pedestrian issues, parking problems, etc. There are many related researches and systems trying hard to solve these issues and improve the traffic situation.

In this Thesis, we focus on the purpose of reducing traffic congestion and letting the traffic become more smooth. So we develop a dynamic adjustment

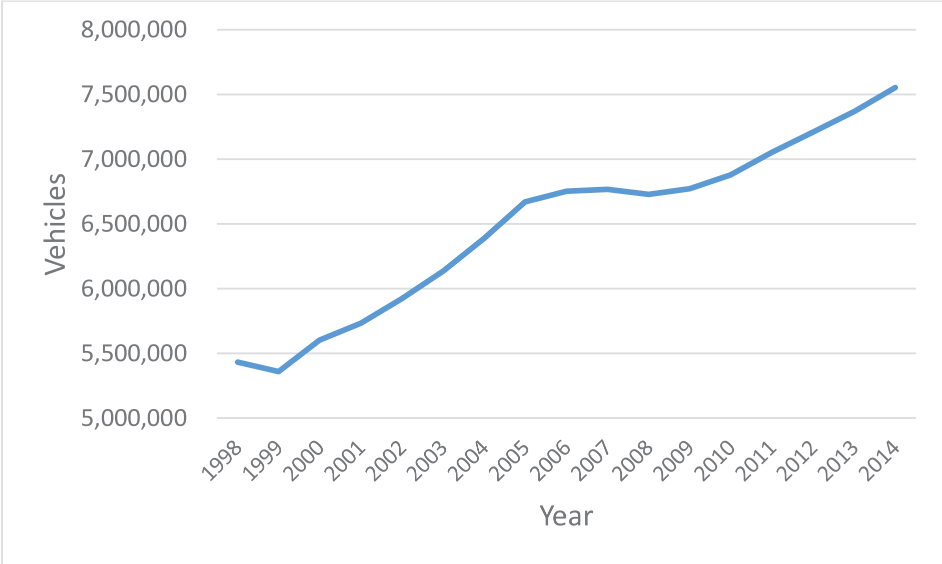

Figure 1.1: Total number of registered vehicles in Taiwan, 1998-2014.

system using *Model Predictive Control* (MPC) [3]. There are two main parts in our method. We adopt the *Back-Propagation Neural Network* (BPNN) [4] to design the prediction model and the *Genetic Algorithm* (GA) [5] to design the optimization method. The prediction model is trained using the historical data of traffic flow information. The trained prediction model is then used to predict the flow in different traffic situations like peak or off-peak. For the optimization method, the prediction results, previous period traffic time settings and the neighbour intersections time settings are considered in a GA fitness function for calculating the new time setting for the next period. In addition to the above, we also develop a traffic simulator to help us implement our method and verify if the optimization result works or not.

1.1 Background

In this section, we describe some issues of intelligent transportation and traffic congestion solving method.

1.1.1 Architecture of Intelligent Transportation System

Intelligent Transportation System (ITS) [1] [6] is now the hottest topic in addressing traffic issues. There exist some serious traffic problems, such as traffic congestion, traffic accidents, air pollution, and energy consumption. To solve these problems, the United States Department of Transportation (DoT) and ITS-America formulate the structure of ITS. Figure 1.2 shows some main systems and service in ITS, including ATMS, ATIS, APTS, CVOS, EPS, EMS, and AVCSS. With the related service of ITS, we can promote traffic safety, reduce environmental impact, improve transport efficiency, and so on.

Among the above ITS subsystems, the *Advanced Traffic Management System* (ATMS) [7] is more associated with the traffic congestion problem. The traffic light control system is one of the research items in ATMS. Using traffic sensors on a road, we can collect different traffic statistics and data, such as traffic speed, traffic flow rate, and occurrence of traffic accidents. By integrating and analyzing these traffic information, we can have more accurate traffic management and decision.

1.1.2 Countermeasures for Traffic Congestion

Following are some strategies to improve traffic congestion.

- Public transport

  It is a shared passenger transport service that can significantly reduce number of private vehicles. Public transport include bus, train, *Mass Rapid*

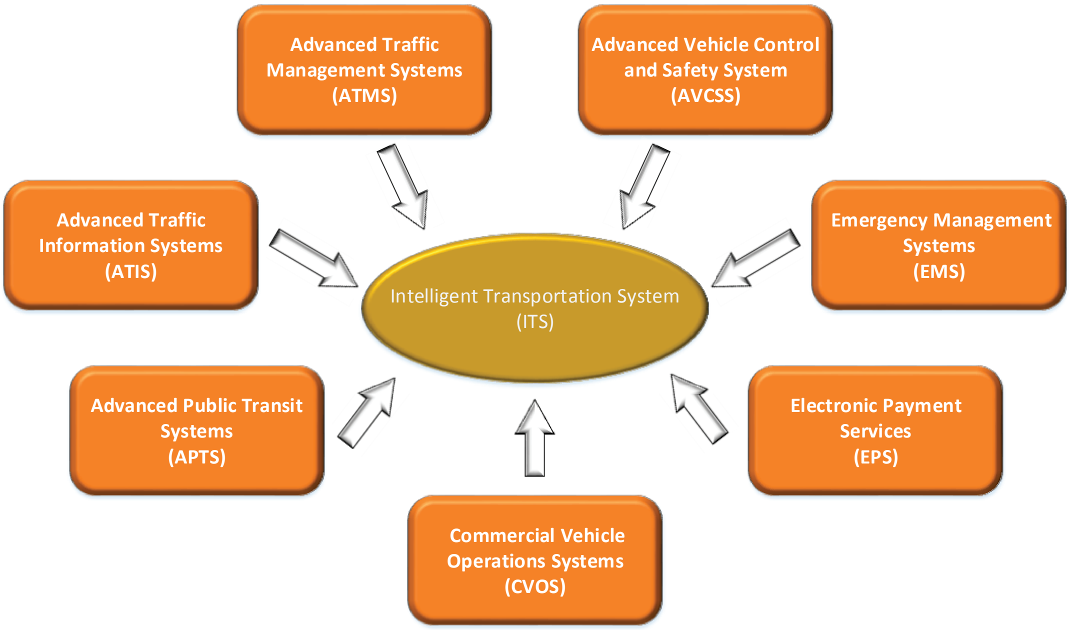

Figure 1.2: Intelligent Transportation System (ITS) architecture [1].

Transit (MRT), *High-speed rail* (HSR), tram, ferry, etc.

- Route planning

  By doing route planning before travel in advance. We can use the related traffic data to calculate the expected travel time, and also avoid the road section which has accident. Then we can choose one of the best feasible routes and might avoid congestion. It can also be collocated with the *Global Positioning System* (GPS) to dynamically adjust travel path during the journey.

- Traffic light control

  Adjusting traffic light period is one method to reduce the number of cars waiting in queue. Nowadays, most traffic signal control systems use different period settings between peak and off-peak times.

- Traffic prediction

Traffic prediction techniques use historical traffic data obtained from sensors or cameras. The predicted future traffic states are then provided to the route planning or traffic light setting system.

- Pricing strategies

  Through charging fees from road users at certain times, it can limit the volume of vehicles accessing roads. For example, it is now practiced in Singapore, London, and some other cities. Another system is called "Cap and trade", which only allows licensed cars that have paid money on the road. Charging higher parking fees in congestion sections is also a good strategy to reduce user's desire to cross or go into easily-congested popular section.

- Road infrastructure

  The government or related organization can broaden or build new roads to increase the number of roads, or redesign roads to reduce junctions, such as elevated or underground roads. It can also build highways without traffic light to improve traffic situation.

1.1.3 Traffic Light Control System

In our work, we focus on the traffic light control system. There are many researches or systems developing the related technique about this.

Some state-of-the-art traffic control software, including TRANSYT-7F, PASSER-II, SOAP, SIGOP-III, UTCS, COMDYCS, and TRUSTS [8]. In the part of control logic, Miller's Algorithm, OPAC, ARTC, SCOOT, SCATS, MOVA, SAST, and TOL are some common methods [8] [9]. Table 1.1 shows the full name of these methods. There are also some simulation models, such as CORSIM, VISSIM,

Table 1.1: Example of traffic light control logic.

Abbreviation	**Full name**
OPAC	Optimization Policies for Adaptive Control
ARTC	Areawide RealTime Traffic Control
SCOOT	Split, Cycle, Offset Optimization Technique
SCATS	Sydney Coordinated Adaptive Traffic System
MOVA	Microprocessor Optimized Vehicle Actuation
SAST	Stepwise Adjustment of Signal Timing
TOL	Traffic Optimization Logic

PARAMICS and SIMTRAFFIC [10].

1.1.4 Example

Before presenting our work, we perform a simple experiment to observe some phenomena. Table 1.2 gives some statistical data obtained from our own traffic simulator [11]. Assume that currently the traffic flow we set to the simulator along the horizontal direction is at its peak, and that along the vertical direction is at off-peak. In fixed time setting system, we use the same number of seconds as the traffic light setting time in each traffic light period. Then calculating the waiting rate and waiting time per 20 cycles. We can see that the *Intersection Average Waiting Rate* (IAWR) and *Intersection Average Waiting Time* (IAWT) defined in Chapter 3 are similar in the four iteration of data calculation. If we do not change the traffic time setting, the IAWR and IAWT will all remain quite high. Thus, we need to dynamically adjust the traffic light time settings according to the traffic flow, and thus improve the traffic congestion. This table also shows the improving results in our method.

In another example, Figure 1.3 shows the vehicles amount per hour on Zhong-

Table 1.2: Average waiting data using fixed time traffic light setting.

	Fixed Time System		Our method	
Cycle	**IAWR**	**IAWT**	**IAWR**	**IAWT**
0-20	60.34 %	9.53	61.66%	9.83
21-40	59.89 %	9.94	50.08%	7.92
41-60	58.10 %	9.46	48.48%	6.87
61-80	59.06 %	9.50	48.16%	6.71

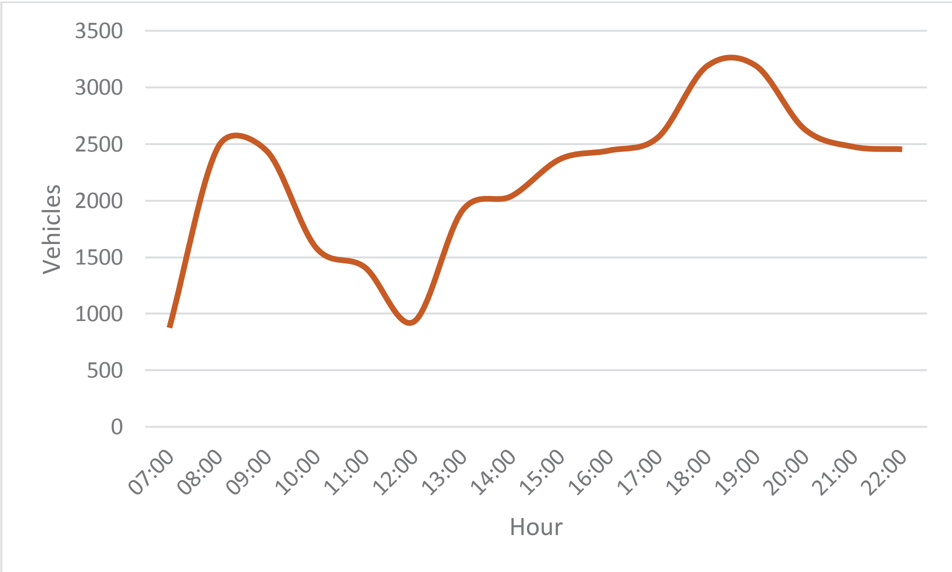

Figure 1.3: Traffic flow in Zhongshan Rd, Kaohsiung City, 7am-10pm, 2005/09/23.

shan road in the Kaohsiung city on September 23, 2005 [12]. We can see that the traffic flow may have significant fluctuation in short time. Especially in the city, because of the peak and off-peak state, the traffic flow is generally unstable. The congestion situation often happen in these areas, too. Thus, we can also try to implement a traffic flow prediction model to help us predict the traffic flow in the future and set the traffic light with suitable timings.

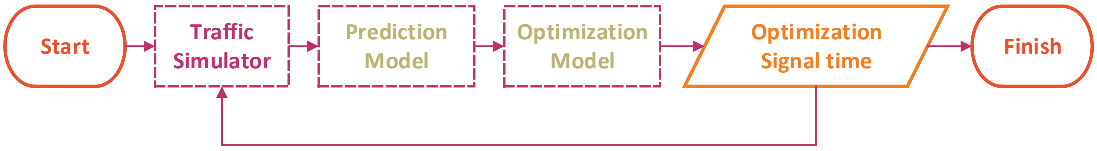

Figure 1.4: Proposed traffic light control and optimized framework.

1.2 Motivation

The control of traffic signal timing is an important issue in the design of traffic control systems, because an inappropriate timing could lead to heavy congestion in the traffic flow. Currently, most traffic control systems still use preset time periods to control traffic lights. A traffic light is generally divided into green, yellow and red signals. The time length of a traffic light is normally set depending on the type or location of roads and on the peak or off-peak times. But these systems mostly have time settings fixed for a very long cycle. Often, the same setting time is applied for several hours. Thus, it cannot be adjusted dynamically to address the issue of varying degrees of traffic congestion. However, in the real world, there could be some unexpected situations such as traffic accidents, natural calamities, or sudden incidents. In such cases, if the traffic light control system cannot adjust dynamically and adaptively, traffic congestions will occur.

1.3 Goal

In this Thesis, we want to propose a traffic light optimization system using MPC. There are two main things we want to accomplish. One is predicting the traffic flow in the future, and the other one is optimizing the traffic timing. Figure 1.4 shows the proposed traffic light control and optimization framework. The final goal we want to attain is to reduce traffic congestion.

1.4 Thesis Organization

The rest of the Thesis is organized follow. Chapter 2 introduces the related works of traffic systems, including the prediction model, optimization model and MPC system. Chapter 3 gives some definitions and parameter settings used in our method, and also gives some assumptions for our work. Chapter 4 presents the overall framework and the proposed algorithms for our traffic light control systems. Chapter 5 provides the experiment results in our work, and also the analysis results compared with other researches. Chapter 6 concludes the Thesis and gives some future work.

Chapter 2

Related Work

In this chapter, we review some previous approaches about the issues of MPC applying on traffic, traffic prediction models, traffic light optimization methods, traffic simulators, and control systems.

2.1 Researches on MPC

Different from other traditional control methods, MPC depends on dynamic models of the process. The main function of MPC is that it can optimize the target in the current time slot and also take future time slot in account. It can just adjust or implement the setting at the current time slot but also optimize a finite time horizon in the future. In other words, MPC can anticipate future events and take corresponding control actions. Figure 2.1 shows a conceptual picture of MPC and Figure 2.2 shows the typical structure of MPC.

A lot of researches about applying MPC to traffic have been proposed. Kim et al. [13] implemented a collision free ground transportation system and an autonomous intersection management framework. In the MPC system, the vehicles can automatically determine its own motion locally by communicating with neigh-

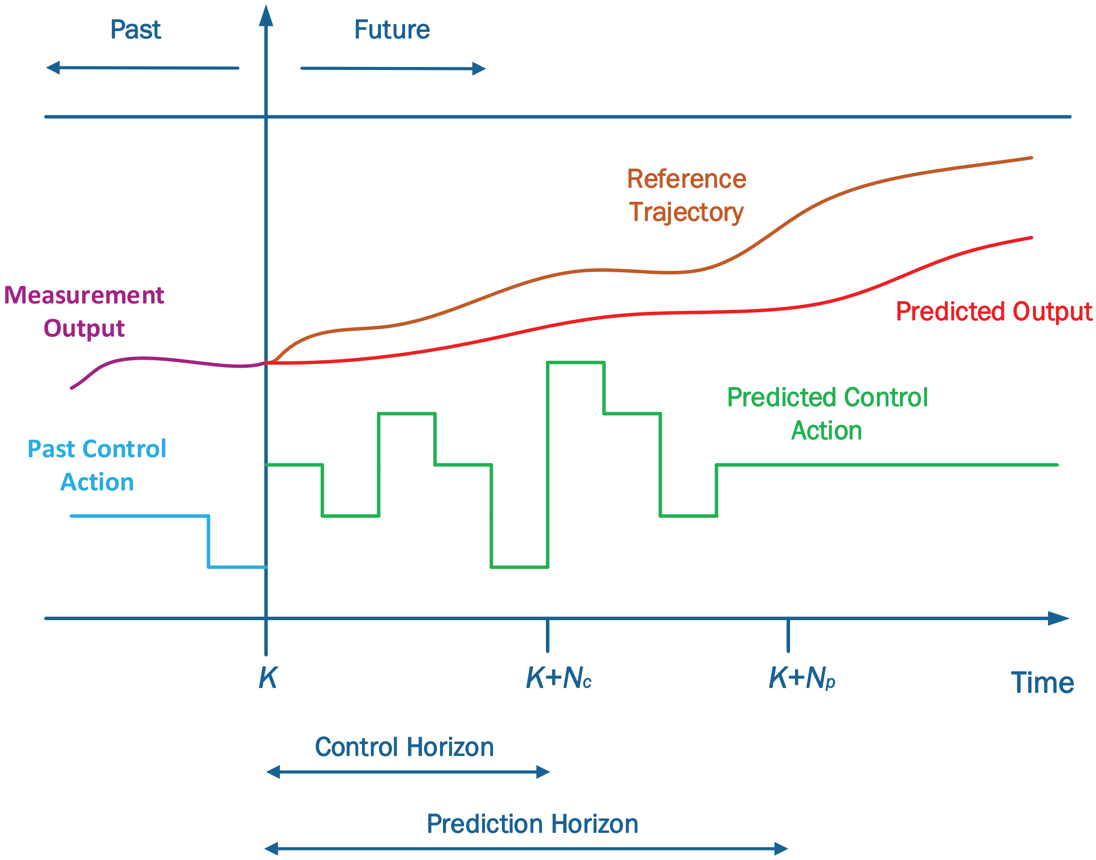

Figure 2.1: Scheme of Model Predictive Control (MPC).

bor vehicles, the road speed limit, acceleration, and deceleration values, etc. Then improving the safety of traffic. Khodayari et al. [14] proposed a car following control system based on MPC. They employed the relative distance and acceleration of vehicles to construct a linear and continuous model. Then using MPC model to control the behavior of car following and ensure road safety. Frejo et al. [15] proposed a MPC approach applying on freeway traffic control system. They considered the elements, such as density, speed, ramp flow, and queue to design the related equations. Using GA to estimate the control results and then implement ramp metering control management.

Some researches also focused on the purpose which relating to our works. Zegeye et al. [16] proposed a MPC-based approach that can reduce travel time and vehicular emissions. They also presented a *linear parameter varying* (LPV) formu-

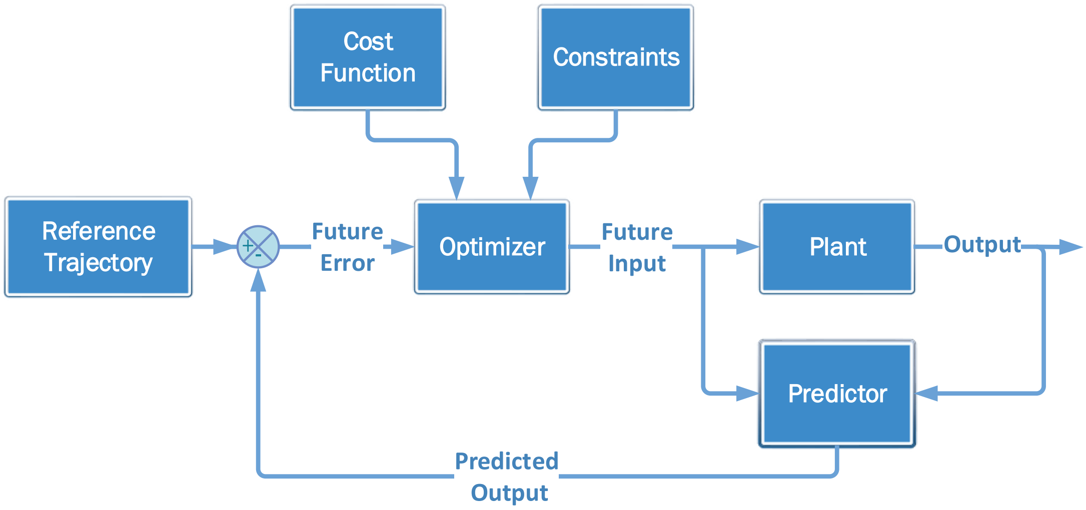

Figure 2.2: Flow diagram of Model Predictive Control (MPC).

lation as traffic flow model to approximate the emission model and traffic flow. In the paper of Tettamanti et al. [17], a MPC-based control system was established to relieve traffic congestion. By estimating with the state feedback of real traffic environment and limiting the traffic light time, it can improve homogeneous traffic flow and reduce travel time. In the paper of Zhou et al. [18], they brought up the shortcoming of traffic light control system based on centralised MPC. Because it is impractical and unsuitable to control the complex traffic signal network in a wide area. They proposed a decentralised coordinate MPC method to handle these problems. In this algorithm, by dividing the whole network into some subsystems, they can calculate data separately. But also having ability to communicate the decisions with neighbourhood subsystems. They also showed that the new algorithm will converge to a global optimum solution after enough iterations. Table 2.1 lists the above MPC-based applications.

Table 2.1: Some applications using MPC model in traffic domain.

Year	Literature	Application	Simulation model or platform
2008	[17]	signal control system	VISSIM
2010	[16]	reducing vehicular emissions	MEATNET, VT-macro
2012	[14]	car-following behavior	not mentioned
2012	[18]	decentralised coordinate signal control system	VISSIM
2013	[15]	freeway ramp metering control management	not mentioned
2014	[13]	collision free autonomous ground transportation system	self made simulator

2.2 Researches on Prediction Models

Prediction model is one of the most significant issues in traffic, especially in the traffic systems nowadays. More and more instruments, such as cameras, sensors, loop detectors or other detection tools are widely applied in the ITS. The traffic information and data which gathered from these equipment provide an important application used in related topics. Among these, traffic prediction model has been extensively researched.

Figure 2.3 shows the basic traffic prediction model diagram. Using the historical traffic data as the inputs of prediction model, we can forecast the speed, vehicle volume, and traffic wave in the future. With these results, we can solve some problems in traffic issues, such as scheduling, routing, traffic light controlling, etc. In order to deal with these road traffic big data, Chung et al. [19] proposed a basic framework of traffic big data analysis processing. They used distributed *Complex Event Processing* (CEP) to process massive real-time data and *Enterprise Service Bus* (ESB) to integrate with other related services. They also used Hadoop and HBase as tools to analyze and store real-time collision data.

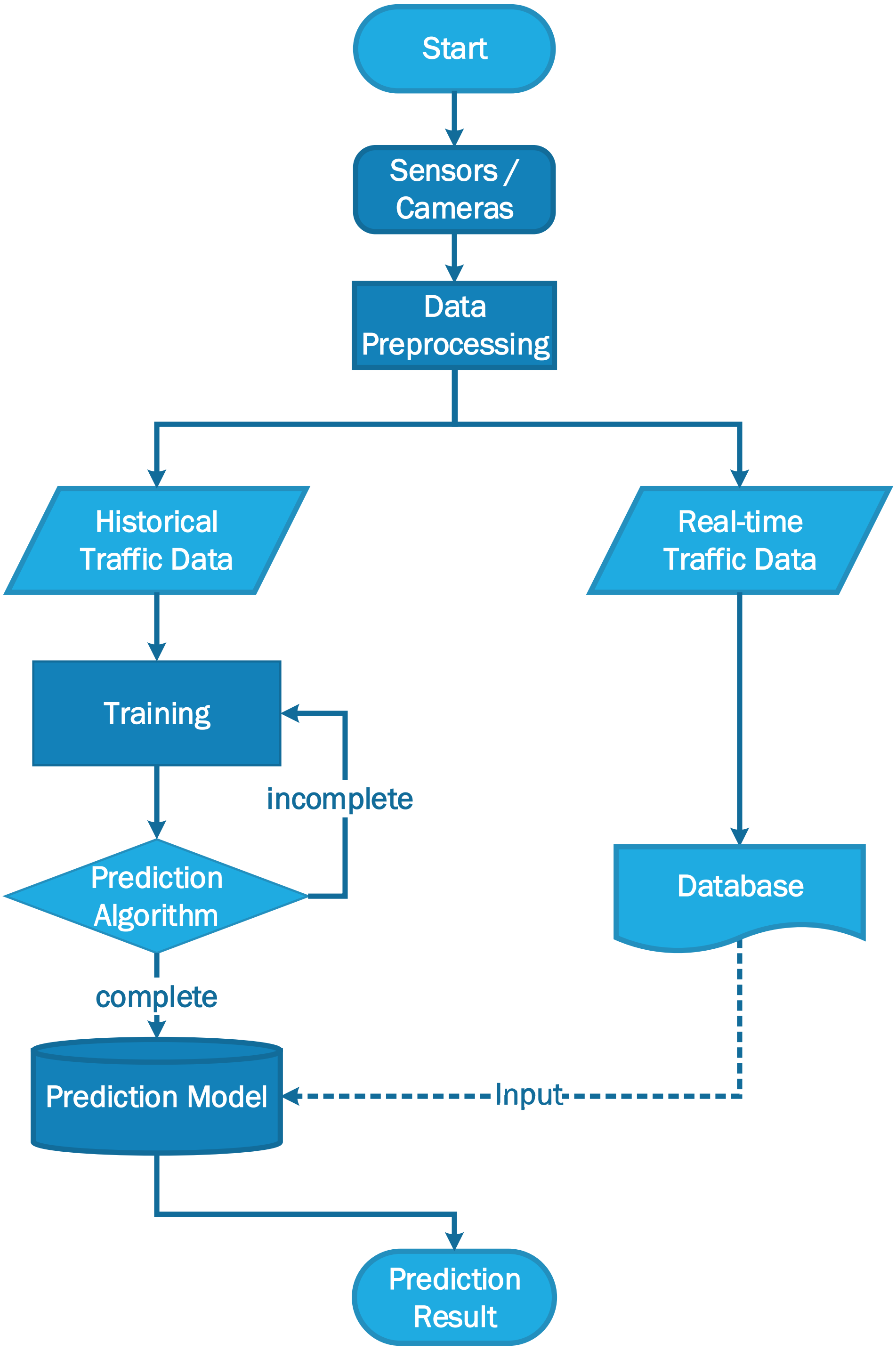

Figure 2.3: Flow diagram of traffic prediction model.

In the topics of traffic prediction model, there are a lot of researches have been proposed many algorithms or methods. Lin et al. [20] introduced a model based on *urban traffic network* (UTN). The method focuses on the mechanism of traffic flow movement and establishes a network topology to link the entire urban traffic. Then Kong et al. [21] added a speed-density model which based on fundamental diagram (FD) into UTN to decrease the vehicle delay time and improve traffic congestion.

Some existed algorithms in common domain are also applied into traffic prediction models. Zhu et al. [22] developed a model based on Kalman filtering theory to forecast traffic volume. Guo et al. [23] gave a traffic predictor using *k-Nearest Neighbour* (kNN) with *Singular Spectrum Analysis* (SSA) and having a great improvement in the incident case. When the traffic status is suddenly changed by traffic incident, this model can provide a quick response. Wang et al. [24] used the data mining technique to analyze the large amount of historical traffic flow data. Two data mining methods are contained in this model. One is clustering analysis, and the other one is classification analysis. Then using K-means algorithm for clustering the different data and decision tree (C4.5 algorithm) as the classification model. Finally predicting the traffic volume and occupancy by searching the decision tree.

Artificial neural network (ANN) is also an excellent algorithm in prediction model. With its transformation, BPNN adds the training model and can get the more accurate results. In this Thesis, we choose the BPNN algorithm to develop our prediction model. Following are some works focus on this algorithm. Li et al. [25] proposed a basic framework of BPNN applied to traffic prediction models. For the preparing work, users should initial the connection weights, scale parameters, and translation parameters and also determine the iterating times and learning factors. Then computing the output results and error signal from neural network. Finally using the error signal to adjust the weights to let the output results more close to the expectation outputs period by period. Hu et al. [26] used BPNN to train the suitable weights to gradually adjust the prediction model. Then dividing the time into 10 minutes and using four time zones before to predict the next 10 minutes traffic volume. In Park et al. [27] model, their method can predict the speed of entire route before journey. First, they divided the route according to the sensor location. Then if users want to predict the speed in the future time $t + i\Delta t$ (t means the time right now) of specific sensor on the route, they should use the current speed of this sensor, one next sensor, and i previous sensor to predict the speed at time $t + i\Delta t$. Finally using these results to reversely conjecture the corresponding speed and arrival time of every sensors. Tong et al. [28] combined the *Fuzzy Network* (FN) and *Neural Network* (NN) to a model *Fuzzy Neural Network model* (FNNM). They used FN module to do fuzzy clustering and supervise the learning of NN. Then finally getting the prediction flow data. In the work of Li et al. [29], they gave a new design of transfer function in NN. This model has a great reduction of training error and prediction error and then significantly enhance the convergence speed. Table 2.2 shows the list of above prediction models.

2.3 Researches on Optimization Methods

Currently, traffic flow is getting more saturated nowadays. The large number of vehicles in cities is increasingly difficult to load. Traffic congestion is becoming a serious problem in many countries. Although more and more roads and flyovers are established, but it's still not a fundamental method to solve congestion issues.

Table 2.2: Researches on prediction models.

Year	Literature	Scheme	Considerations	Objectives
1999	[22]	Kalman filtering theory	traffic flow, time interval	traffic volume
2006	[28]	Fuzzy neural network (FNN)	traffic flow, time interval	traffic volume
2006	[24]	Data mining	traffic flow	traffic volume
2008	[20]	UTN model	speed, traffic flow, vehicle length, road capacity, time interval	traffic volume
2010	[26]	BP neural network (BPNN)	traffic flow, time interval	traffic volume
2011	[27]	Neural network (NN)	speed, time interval, distance	traffic speed travel time
2012	[23]	Singular spectrum analysis (SAA) + k-nearest neighbour (KNN)	traffic flow, occupancy	traffic volume
2013	[21]	UTN model + fundamental diagram (FD)	speed, traffic flow, vehicle length, road capacity, time interval, jam density	traffic volume
2014	[25]	Wavelet neural network (WNN)	traffic flow, time interval	traffic volume

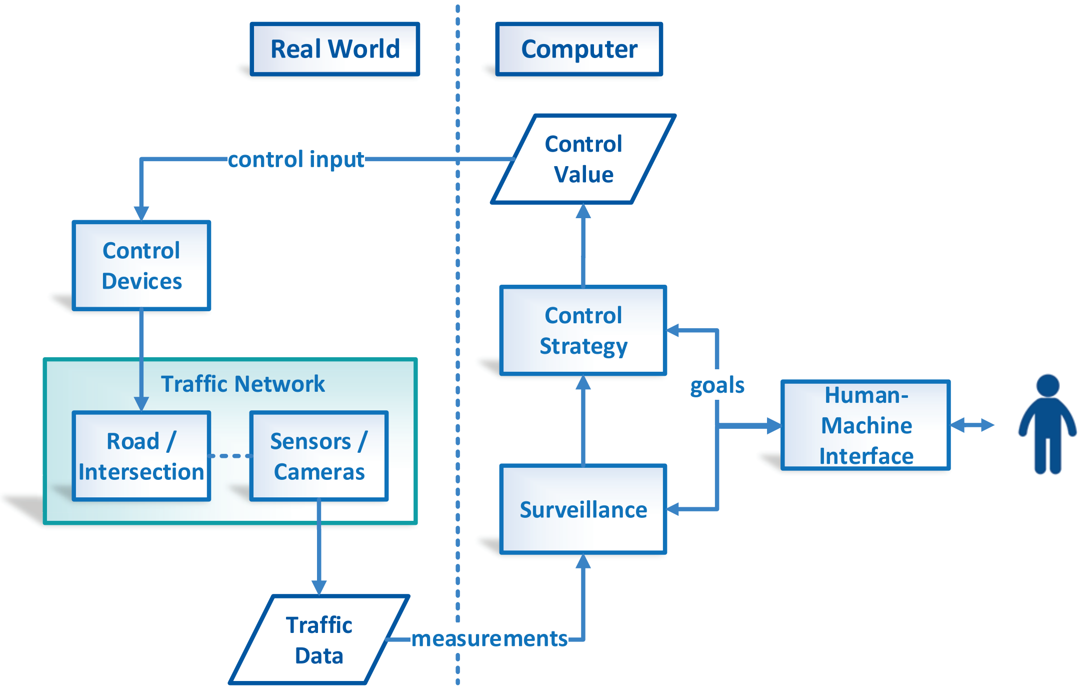

Figure 2.4: Traffic light control loop.

A lot of traffic light controllers are proposed now in order to completely alleviate the traffic congestion. The traditional traffic light control systems mostly use fixed setting time and lack the ability to variate with real-time traffic situation. So the studies of dynamic adjustment traffic light setting time are now popular. We can utilize the traffic data collected from real roads and intersections. Then calculating or inferring the optimized setting time. Figure 2.4 shows the basic traffic light control loop. The main purpose is to decrease the delay time of vehicles or queue length behind red light.

Some related studies have been proposed in this area. Tawara et al. [30] applied a pheromone model to predict the degree of traffic congestion. They used the vehicle speed and the type of vehicles to calculate and accumulate the amount of pheromone in a road section. Then using the result to determine the traffic congestion status and react the setting time to the traffic light. Tan et al. [31] gave an traffic light control system based on fuzzy logic technology. Different from the fixed-time controllers which have fixed extension time, the fuzzy logic controller uses fuzzy logic, such as small, medium and long. The traffic flow are also converted into fuzzy values as input in the controller. Then deriving the traffic light setting time through the fuzzy rule base. Srinivasan et al. [32] designed a multiagent system with NN algorithm. They used *Simultaneous Perturbation Stochastic Approximation* (SPSA) to update the weights of neurons. Then constructing the agent using SPSA-NN.

In the study [10], we can see that the comparison results between GA and other methods, the optimization methods in GA is more dominant. In addition, the signal time of the specific traffic light to be optimized and its neighbors have a lot of permutation and combination, and GA is also a proper algorithm to solve these problems which has massive population. So we choose GA for our optimization method designed in this Thesis.

Following are some works using GA in traffic light optimization systems. Teo et al. [33] introduced a basic framework of GA applied to the traffic light optimization systems. They gave some suggested parameter setting of GA, and also proposed a easy formula to simply predict the queue length of all phases in an intersection. Turky et al. [34] proposed a model using the input variables, which are the number of vehicles behind red light (vehicle queue), the number of vehicles passes a green signal and the average arrival rate of vehicles to the red light. Two genes are contained in a chromosome, one gene is green time, and the other one is red time. Then using above parameters to design the fitness function of GA. The output results are the optimized green and red time after calculating with specific period. Singh et al. [35] used the traffic flow data and different weights in different roads as input variables. Giving the random extension time to the

Table 2.3: Researches on traffic light optimization methods.

Year	Literature	Scheme	Considerations
1996	[31]	Fuzzy logic	traffic flow, vehicle queue
2006	[32]	Neural network (NN) + Fuzzy logic	occupancy, traffic flow, loading, average arrival rate, cooperation
2009	[34]	Genetic algorithm (GA)	vehicle queue, traffic flow, average arrival rate
2009	[35]	Genetic algorithm (GA)	traffic flow, road capacity
2010	[30]	Pheromone model	vehicle type, speed, time interval
2010	[33]	Genetic algorithm (GA)	traffic flow, vehicle queue, time interval

fixed minimum green time in each road. Then calculating the *Performance Index* (PI) with above parameters. Finally accumulating all the PI of four roads in a four-way intersection, and finding the minimum result as traffic light setting time. Table 2.3 gives a summarization of above optimization methods.

2.4 Researches on Traffic Simulators and Traffic Light Control Systems

Traffic simulation models provide a fast, convenient, cheap and risk-free platform without implementing in the real environment. It can simply and quickly simulate for different scenarios that is hard accomplished in real world conditions and also quickly calculate the related data we need.

There are some business software of traffic simulators have been used for a long time. The most widely used simulators are CORSIM, VISSIM, PARAMICS, and SIMTRAFFIC. Park et al. [36] and Ratrout et al. [37] introduced the basic functions of these simulators and compared them in some level. But because of the custom measure statistics we need in our experiments and the convenience to insert ourselves algorithms, we still develop a self made simulator which has been simply introduced in our published paper [11]. It will give detail introduction in Chapter 3 and Chapter 5. Table 2.4 shows the reduced table of simulation program features from report [36]. We also referred to some main features to design our traffic simulator.

Traffic light control systems are also important in traffic. They are used to set the green splits, cycle length, offsets, etc. Study [38] and [39] have presented the introduction and comparison of some popular traffic light control software, such as TRANSYT-7F, SYNCHRO, and PASSER-II.

Table 2.4: Summary of traffic simulation program features.

Model Comparison		CORSIM	VISSIM	Pmamics	SIMTRAFFIC
Input	Background Image	Bitmap	Bitmap	Bitmap	DXF, Bitmap
	GUI	Yes	Yes	Yes	Yes
Modeling	Pre-timed Signal	Yes	Yes	Yes	Yes
	Actuated Signal	Yes	Yes, VAP code provided	Yes, API need to be developed	Yes
	Transit	Yes	Yes	Yes	No
	Pedestrian	Delay Factor	Yes	Yes	Yes
Output	2D Animation	Yes	Yes	Yes	Yes
	3D Animation	No	Yes	Yes	No
	Performance Measures	travel time, delay, queue time, stop time, speed, percent stops, fuel consumption	travel time, delay time, volume, speed, position of vehicle	travel time, stop time, saturation flow, volume, pollution	delay, travel time, queue time, stop time, speed, percent stops, fuel consumption

Chapter 3

Preliminaries

In this chapter, we give a simple formulation for the problem addressed by our proposed system and the definitions of terminologies and parameters. Some assumptions are also listed for our method. Finally, we will introduce the traffic simulator used in the Thesis as the experimental platform.

3.1 Terminology

Following are some basic traffic parameters for which we give the introductions and definitions.

- **Traffic Queue Length**

  It represents the number of vehicles waiting at an intersection signal with red light.

- **Speed**

  It is the speed of vehicles on a road. It is defined as the average speed of traffic flow in a specific road section.

- **Arrival rate**

  It is the average number of vehicles entering a road section per unit of time. It is often used to estimate the length of the traffic queue at a red light.

- **Waiting Time**

  It is the average waiting time of vehicles at a red light of an intersection.

- **Traffic Situation**

  It is the degree of traffic congestion. It is usually classified into congestion and non-congestion or peak and off-peak. It can aid in determining the traffic situation and adopting the appropriate measures to reduce traffic congestion.

Following are the definition of some basic elements of general traffic models. Figure 3.1 shows a simple diagram of common traffic elements.

- **Intersection**

  An intersection is the junction of two or more roads. It often means the region of these roads meeting or crossing. Most intersections usually have the traffic light or roundabout to control the traffic flow along the intersecting roads. In our system, we define the intersection for two typical types, namely four-way crossroads and three-way T junctions.

- **Traffic signal**

  Traffic lights are usually set beside the intersections to control the traffic flow. In our system, there are three states of traffic light, namely red, yellow, and green.

- **Period**

  It means a complete cycle of traffic light. We define the period as the total

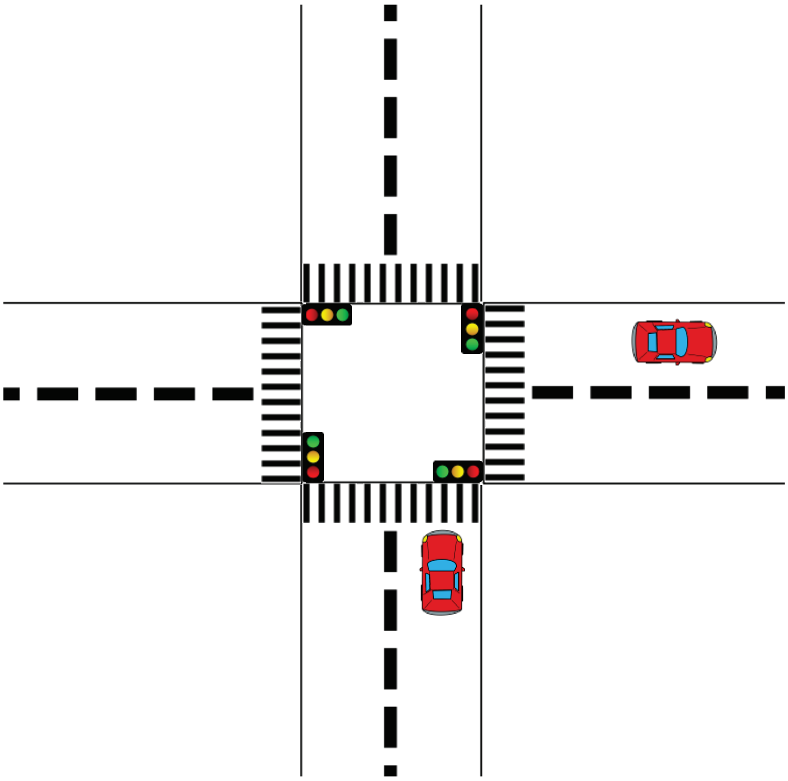

Figure 3.1: Simple diagram of common traffic elements.

time of red, yellow, and green signals once each. The period can help us determining the updating time to adjust the traffic light.

- **Road Lanes**

  It is the number of lanes on a road. In this work, we assume all roads only have one lane in each direction.

- **Traffic Data**

  The data and statistics gathered from sensors or cameras on roads, such as speed, traffic volumes, waiting time. These data can be used in traffic prediction and optimization.

3.2 Problem Formulation

Our target problem is defined as follows. Given the traffic data T_{data} and previous green signal time GT_{past}, the goal is to predict traffic flow and optimize traffic signal timing such that the traffic congestion, if any, is reduced.

To solve the problem we construct a system called MPC-based Traffic Light Control System (MTLCS), where $MTLCS = \langle Pre(T_{data}), Opt(T_{data}, GT_{past})\rangle$.

- $Pre(T_{data})$ is a prediction model for traffic volumes,
- $Opt(T_{data}, GT_{past})$ is an optimization method for traffic signal timing,

The goal is to optimize the traffic signal timings such that the traffic congestion, if any, can be reduced.

We use an in-house traffic simulator to validate our proposed solution.

3.3 Parameter

In this section, we list the variables and parameters used in the traffic flow prediction model and the traffic light optimization method, which will be described in Chapter 4. Other algorithm parameters of BPNN and GA will be introduced in Chapter 4.

3.3.1 Prediction Model

1. T_{train}: Training time threshold, that is, after this threshold is reached the prediction model must be retrained.
2. W_t: Dynamic weights in BPNN that are adjusted in each training period.
3. W_p: Final weights in the prediction models.

4. E: Training error between training outputs and desired results.

5. δ: Error signal for adjusting weights.

6. T_{data}: Data of traffic volumes used for training BPNN.

7. P_{data}: Inputs of traffic volumes for prediction.

8. P_{flow}: Prediction output of traffic volumes.

3.3.2 Optimization Method

1. Q_{right}, Q_{left}: Number of vehicles waiting in the parallel direction of the traffic signal we want to optimize.

2. AVG_{up}, AVG_{down}: Average arrival rate of vehicles in the perpendicular direction of the traffic signal we want to optimize.

3. GT_{p_x}: Green signal time at intersection x in the previous period.

4. TFP: Time for a vehicle passing an intersection.

5. GT_{GA}: The result of GA optimization method in Section 4.2.

6. GT_{final_EW}: Final green signal time for the East-West direction

7. GT_{final_NS}: Final green signal time for the North-South direction

3.3.3 MPC-based Control Method

1. T_{opt}: Optimization threshold, i.e., the time limit after which optimization is performed again.

2. F: Number of time intervals to be predicted.

3. $Flow_{p\_x}$: Predicted traffic volumes in the future x-th time interval.

4. $Flow_{p\_AVG}$: Future average traffic flow.

5. GT_{set}: Final green signal setting time.

3.4 Assumptions

- Traffic data: For our system, some types of traffic data are necessary for prediction and optimization, including the queue length, arrival rate, and waiting rate. Existing traffic monitoring systems or simulators must be able to provide each traffic data before they can use our traffic light control system.

- Traffic lights: Our method applies to general urban roads with traffic lights. The proposed method adjusts traffic signal timing so as to resolve congestion problems,thus roads should have traffic lights. Freeway, roundabout, or country roads without traffic light are out of scope here.

- Intersection : The intersections in most countries are designed into the typical form, that is four-way crossroads and three-way T junction. In other words, the traffic lights have two different setting time between two perpendicular directions. So in our method, the intersections that we can handle are the types mentioned above.

3.5 Traffic Simulator

In order to implement and verify our method, we developed an in-house comprehensive traffic simulator. It contains most components of traffic, including traffic lights, roads, intersections, vehicles, and some useful functions for adjustment and monitoring. Figure 3.2 shows the graphical user interface of our simulator.

The upper toolbar consists of four main parts, which include all functions of our simulator. The first part has some execution buttons including start, stop, restart, and next simulation. The second part is the map and task editors. The third part is the most important part. It includes three managers and one monitoring window. The signal manager can adjust green and yellow signal times for each traffic light in the map. The intersection manager can set optimization threshold and interval of each intersection. The vehicle manager can set vehicle speed, congestion level, and scheduling of traffic flow. The simulator can use these settings with "Poisson distribution" to generate vehicles. The monitoring window can display the statistics of waiting rate of whole intersections and roads. The details of these statistics will be introduced later. The fourth part includes some simulation setting, such as display mode, simulation speed, and graph updating rate. The left side shows the system status, execution information, and congestion degree of each intersection. The middle part is the main graphics display of our simulator.

To check the degree of improvement after optimization, we develop some statistics related to vehicles waiting rate. Following are the parameter definitions and formulas. Table 3.1 and Table 3.2 show the derivation of $IAWT$ and $IAWR$ we used in our experiments.

- PV: Number of vehicles exiting each road segment.
- EV: Number of vehicles entering each road segment.
- TWT: Total waiting time of all vehicles approaching the intersection.
- WV: Total number of vehicles waiting to cross the intersection.

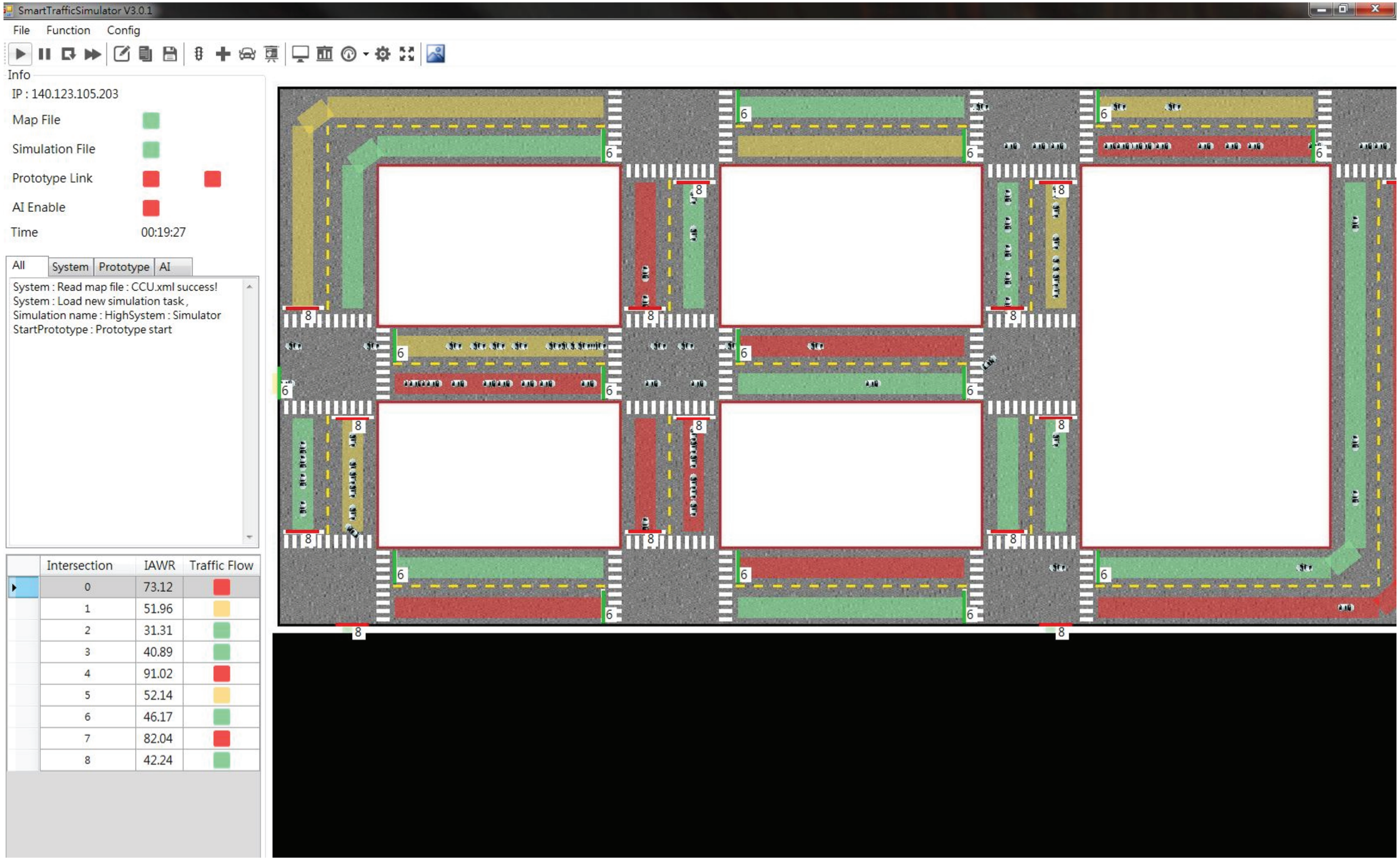

Figure 3.2: Our traffic simulator.

Table 3.1: Simulation statistics for an approach.

Single Cycle Data (i^{th} cycle)	
Waiting Rate (WR_i)	$WR_i = \frac{WV_i}{EV_i}$
Optimization Window Data	
Optimization Window Length in cycles (C)	Number of cycles in optimization window
Average Entering Vehicles (AEV)	$AEV = \frac{\sum_{i=1}^{C} EV_i}{C}$
Average Waiting Time (AWT)	$AWT = \sum_{i=1}^{C} \frac{TWT_i}{EV_i}$
Average Waiting Rate (AWR)	$AWR = \frac{\sum_{i=1}^{C} WR_i}{C}$

Table 3.2: Simulation statistics for an intersection.

Number of Approaches (A)	The number of approaches for an intersection
n^{th} Approach Weight (RW_n) ($1 \leq n \leq A$)	$RW_n = \frac{AEV_n}{\sum_{i=1}^{A} AEV_i}$
Intersection Average Waiting Time ($IAWT$)	$IAWT = \sum_{n=1}^{A} RW_n \times AWT_n$
Intersection Average Waiting Rate ($IAWR$)	$IAWR = \sum_{n=1}^{A} RW_n \times AWR_n$

Chapter 4

MPC-based Traffic Light Control System

In this chapter, we will give a detailed introduction to our method called MPC-based Traffic Light Control System (MTLCS). There are two main parts in MTLCS including a traffic flow prediction model and a traffic light optimization method. The prediction model is designed based on BPNN, while the optimization method is designed based on GA. In addition, there is also a MPC-based controller to handle the communication between these two main functions. Our method uses future prediction data to help optimize traffic light time settings for the next time interval. It is built on a in-house simulator and the traffic data for inputs are also generated from the simulator. Figure 4.1 shows the basic framework of our method proposed in this Thesis.

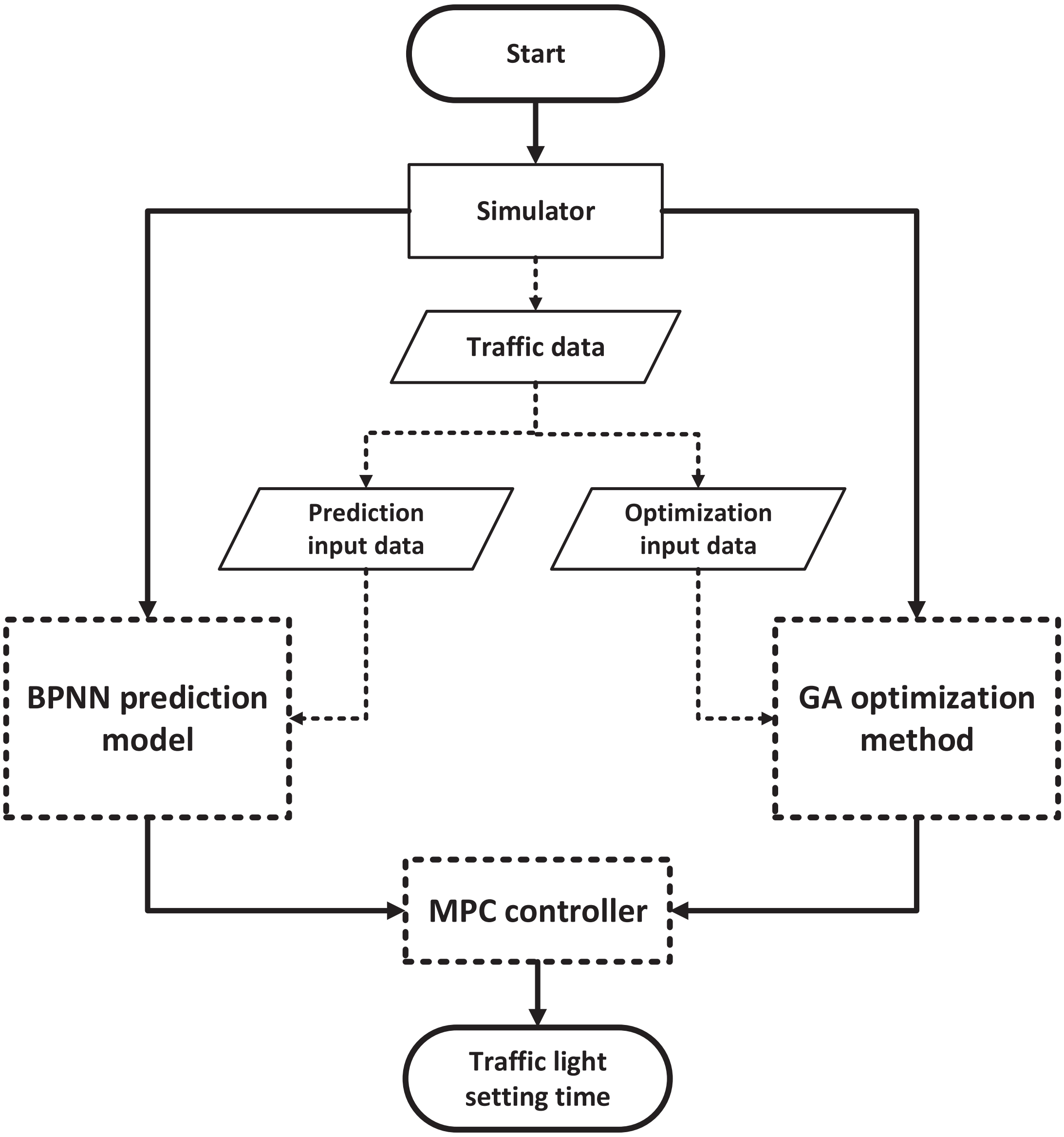

Figure 4.1: MTLCS basic framework.

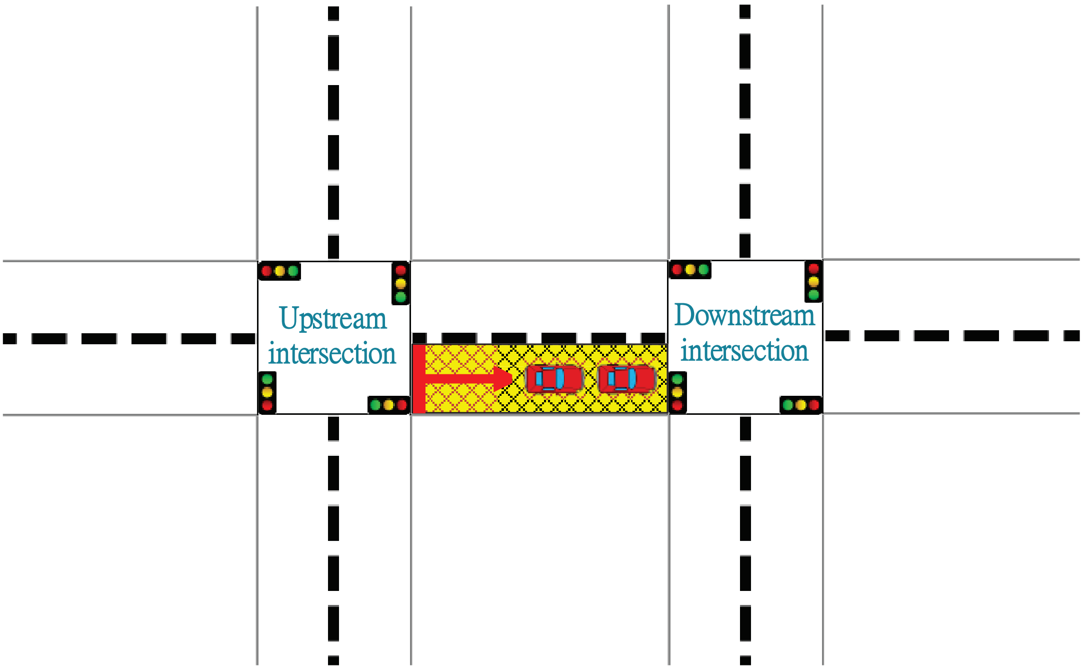

Figure 4.2: Diagram of traffic flow statistics.

4.1 Prediction Model Design

Traffic data are required as inputs for both training and prediction, so we need to calculate the accumulated number of vehicles in a certain road section for a particular time interval. In Figure 4.2, to estimate the average flow data in the cross-checked road section, we count the number of vehicles passing the line besides the upstream intersection, which is then used to predict the future traffic flow for controlling the signal besides the downstream intersection. Figure 4.3 shows the framework of BPNN-based traffic flow prediction model.

4.1.1 Training Method for Adjusting Weights in BPNN

In this section, we explain the basic computation procedures of BPNN first. Figure 4.4 shows the basic structure of BPNN. It has three types of layers, which are the input layer, hidden layer, and output layer. Each pair of nodes in adjacent layers is linked by a weight. The values in all nodes in a previous layer and the

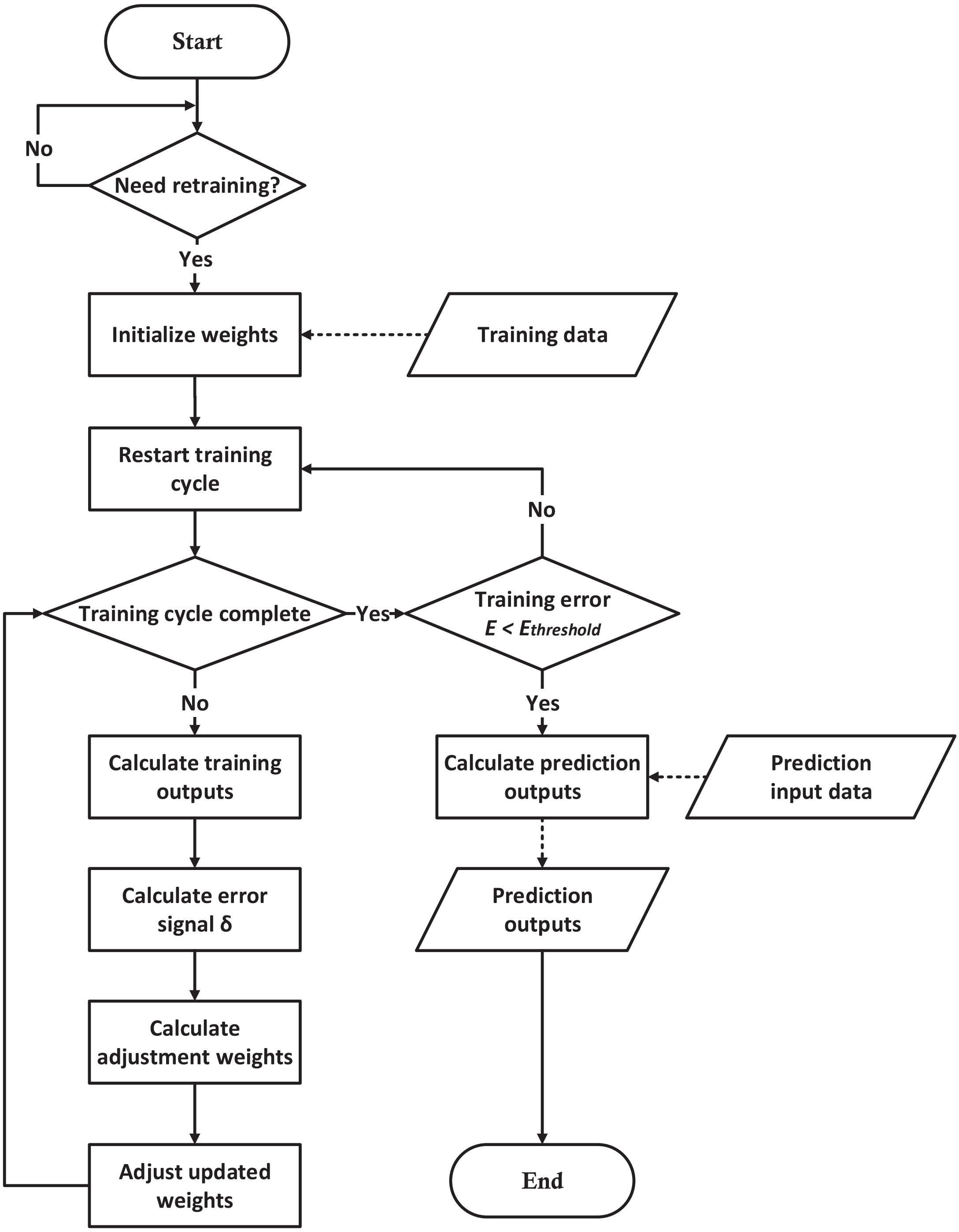

Figure 4.3: BPNN prediction model framework design.

Algorithm 1: Prediction Model Algorithm.

Input:
T_{train}: Training time threshold;
T_{data}: Data of traffic volumes used for training BPNN;
P_{data}: Inputs of traffic volumes for prediction;
$E_{threshold}$: Training error threshold;
Output:
P_{flow}: Prediction output of traffic volumes;
Variable:
W: Weights of BPNN;
E: Training error between training outputs and desired results;
δ: Error signal for adjusting weights;
c_{sat}: 1: Training cycle is complete, 0: Training cycle is incomplete;

if $NeedRetraining(T_{train})$ **then**
    `// Training model`
    Initialize weight W;
    while $E > E_{threshold}$ **do**
        $c_{sat} = 0$;
        while $c_{sat} = 0$ **do**
            $CalculateBPNN(T_{data}, W)$; `// Equation 4.1 and Equation 4.2`
            Calculate error signal δ ; `// Equation 4.3 and Equation 4.4`
            Calculate adjustment weight W ; `// Equation 4.5`
            if *All samples are trained* **then**
                $c_{sat} = 1$;
        Calculate training error E ; `// Equation 4.6`
else
    `// Do prediction`
    $P_{flow} = CalculateBPNN(P_{data}, W)$;
    return P_{flow};

weights are multiplied and accumulated as the input for a node of the next layer. The inputs are then given to an activation function to calculate the output value of the node. Repeating the above operations layer by layer from input layer to output layer, the final output can be derived.

In this Thesis, we use previous traffic flow data to predict future traffic flows via an BPNN model. To dynamically determine suitable weights for different traffic scenarios such as peak or off-peak, urban traffic or country traffic, etc, so we apply

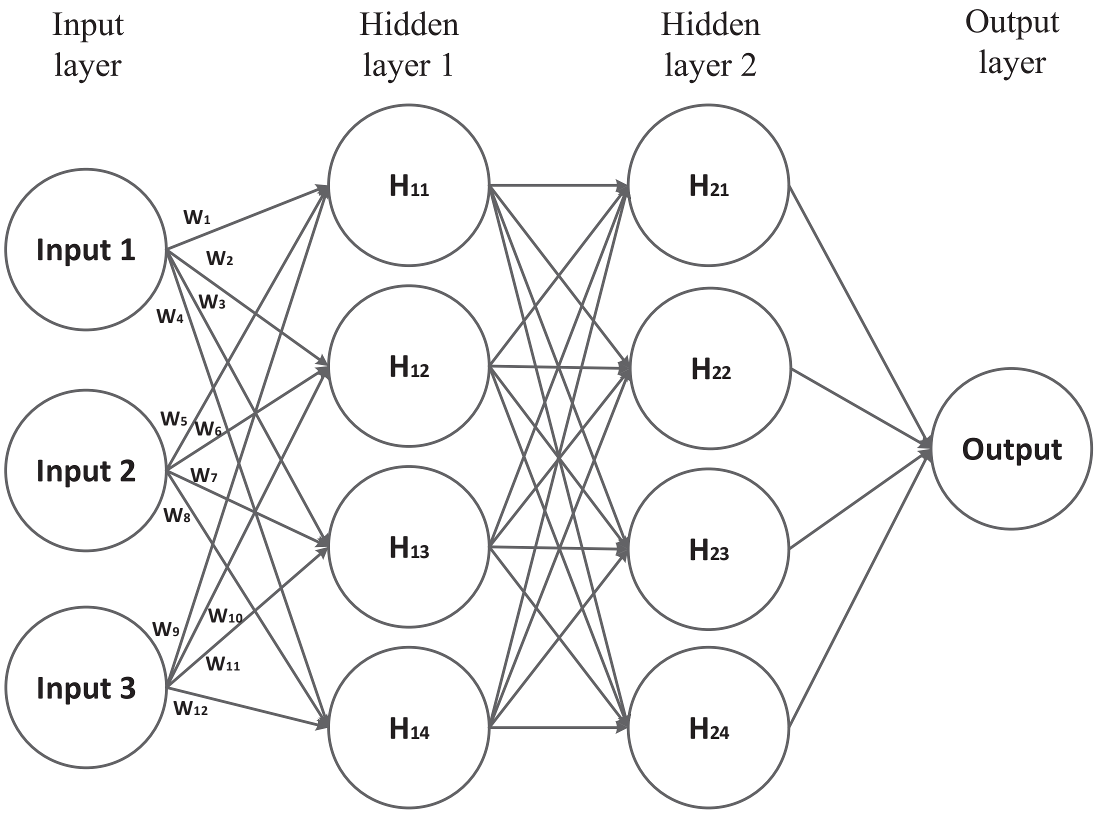

Figure 4.4: The structure of BPNN.

a back propagation method to train and update the weights before prediction. We set the training time threshold T_{train} to one week, it means we will retrain the BPNN model every week and also add new traffic flow data into training model as training samples.

In order to completely illustrate the details of weight training. We give a simple example to show the steps. In Figure 4.5, the two input values 1 and 1.6 and one output value 0.9 are known previously presenting traffic flow and these values mean a set of training sample. Initially, all weights are assigned randomly.

Figure 4.6 shows the calculation method of node values in the hidden layer and the output layer. As shown in Equation4.1, we can calculate the node input value $node_{next}$ by accumulating the products of previous layer node values $node_{pre\_i}$ and the weights linked to the same node in the hidden layer or the output layer

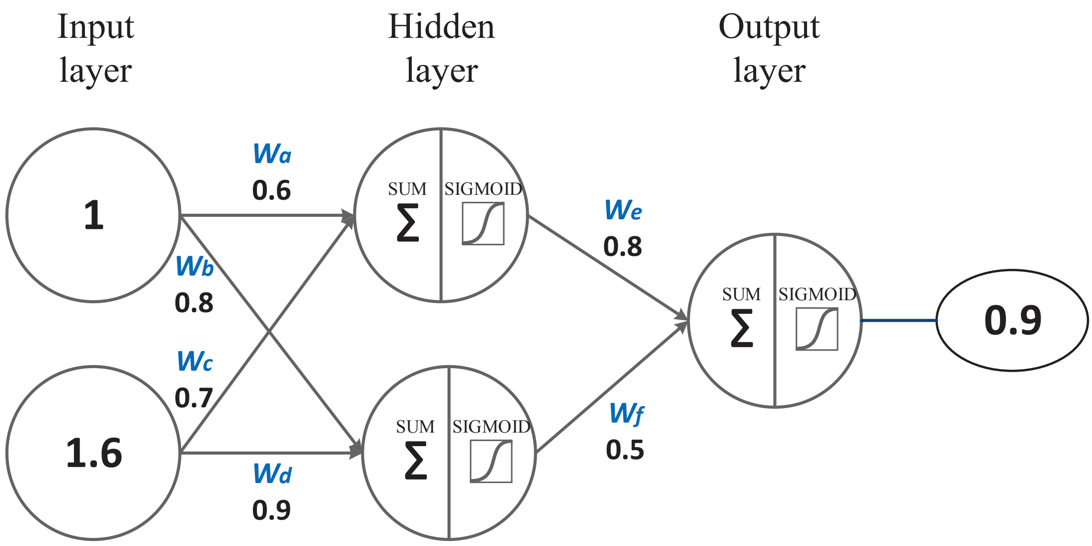

Figure 4.5: An example of BPNN with initial inputs/outputs and weights.

$weight_{pre\_i \rightarrow next}$, I means the number of nodes of previous layer. Upon summation of product using Equation4.1, the two nodes in the hidden layer have as input values of 1.72 and 2.24. Then these values are passed to the activation function in each node, which is the sigmoid function in Equation 4.2 in this BPNN model. Upon calculation, the output values of the two nodes in the hidden later are 0.848 and 0.904. Similarly, the calculation process is repeated between the hidden layer and the output layer resulting in the final output of 0.756.

$$node_{next} = \sum_{i=1}^{I}(weight_{pre\_i \rightarrow next} \times node_{pre\_i}) \tag{4.1}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{4.2}$$

The difference between the predicted result (0.756) and the actual result (0.9) is called the result error. In order to reduce result error, the weights need to be updated. Figure 4.7 shows the weight updating steps. First, we need to calculate the error signal δ. There are two types of error signal, namely that for output layer

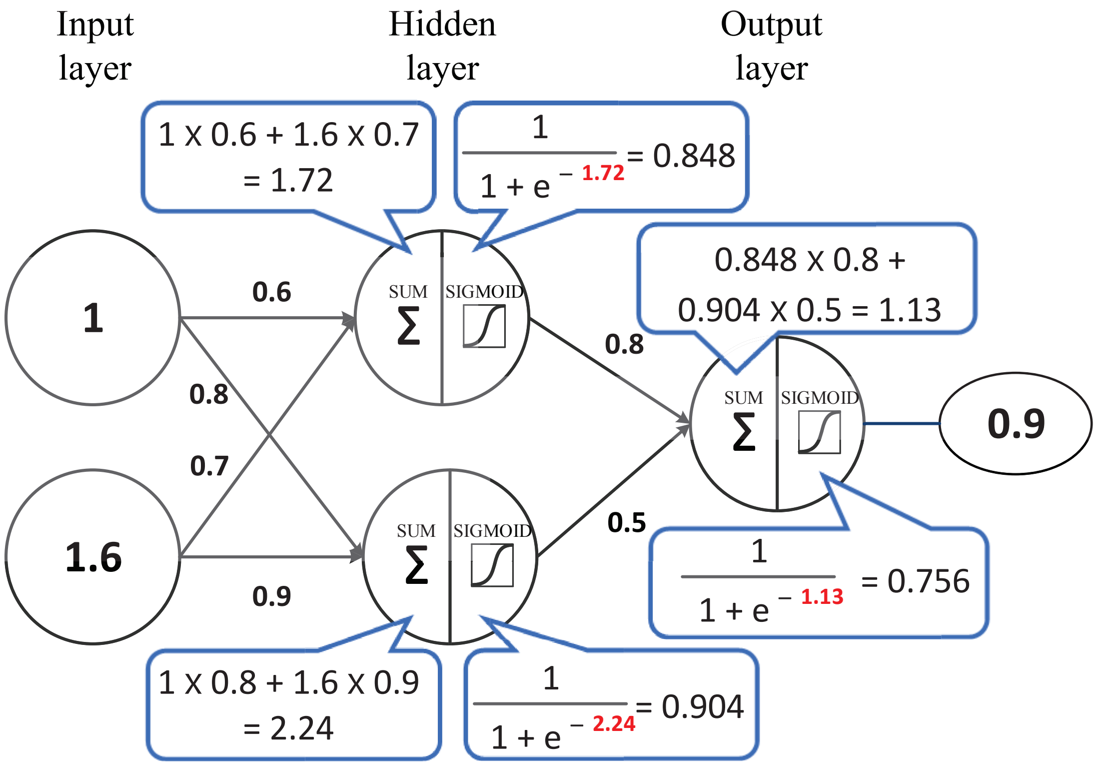

Figure 4.6: An example of BPNN with hidden layer and output layer calculations.

using Equation 4.3 and that for a hidden layer using Equation 4.4. The weights are thus modified by adding the products of errors and outputs to the previous weights shown as Equation 4.5. Every training samples are trained once each means a training cycle. After a training cycle, the training error E are calculated by Equation 4.6 to determine whether the training step is convergence. If it is not less than the training error threshold $E_{threshold}$, restarting the training cycle. We set $E_{threshold}$ to 0.005 in our method. Table 4.1 shows the updated weights and outputs.

$$\delta_{output} = (output_{des} - output_{cal}) \times output_{cal} \times (1 - output_{cal}) \tag{4.3}$$

$$\delta_{hidden} = \sum_{k=1}^{K} (\delta_{next\_k} \times weight_{next\_k}) \times output_{cal} \times (1 - output_{cal}) \tag{4.4}$$

$$weight_{new} = weight_{old} + (\delta \times output_{cal}) \tag{4.5}$$

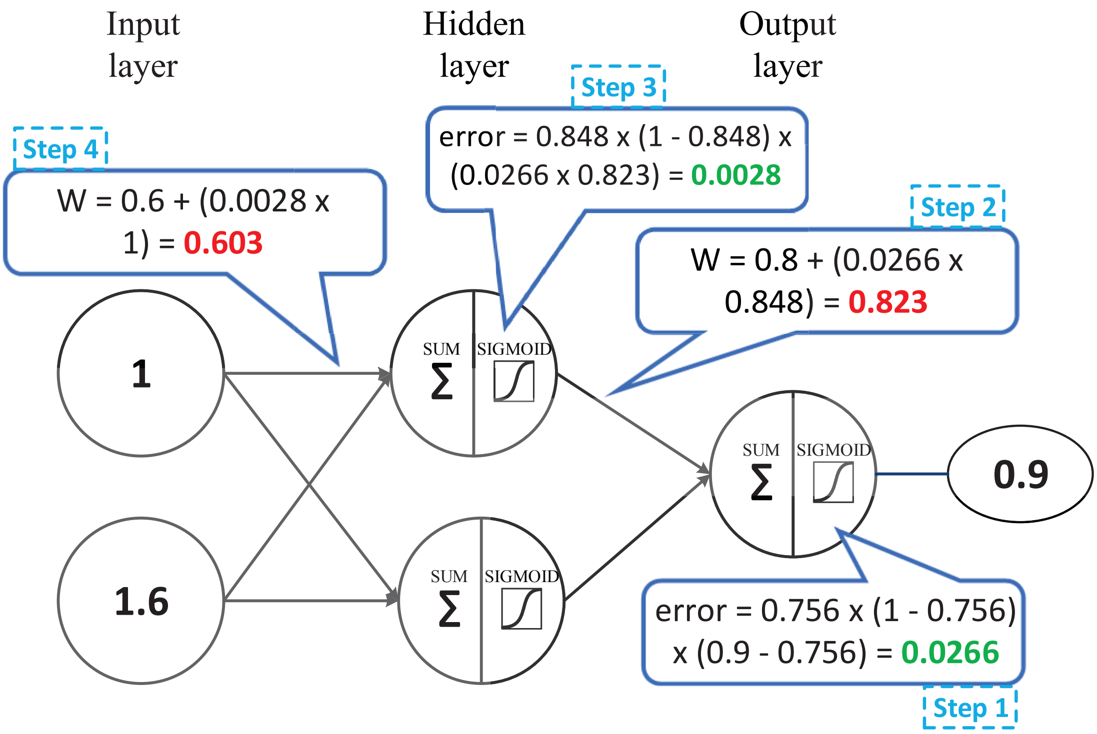

Figure 4.7: An example of BPNN with error calculation and weights updating.

$$E = \frac{1}{M}\sum_{m=1}^{M}(A_m - P_m)^2 \tag{4.6}$$

where

- $output_{des}$: Desired output.
- $output_{cal}$: Calculated output of node in hidden layers and output layer.
- δ_{next_k}: Error signal of node k in next layer.
- $weight_{next\_k}$: Weight linked to node k in next layer.
- K: Number of nodes in next layer.
- $weight_{old}$: Weight before updating.
- $weight_{new}$: Weight after updating.
- A_m: Actual traffic volume value of training sample m.
- P_m: Predicted traffic volume value of training sample m.
- M: Number of training samples.

Table 4.1: Updated weights and outputs of training model.

iteration	w_a	w_b	w_c	w_d	w_e	w_f	output
1	0.600	0.800	0.700	0.900	0.800	0.500	0.756
2	0.603	0.801	0.704	0.902	0.823	0.524	0.763
3	0.605	0.802	0.709	0.904	0.843	0.546	0.770
4	0.608	0.803	0.712	0.905	0.863	0.567	0.777
5	0.610	0.804	0.716	0.907	0.881	0.586	0.783
6	0.612	0.805	0.720	0.909	0.898	0.604	0.788
			...				
19	0.633	0.816	0.753	0.925	1.050	0.765	0.830

4.1.2 Traffic Flow Prediction in BPNN

The traffic flow data of traffic volumes is divided into t_{int} minute intervals, where t_{int} is usually set to 10, 20, etc. We use these time interval values of traffic data as the node values in BPNN. The number of nodes in the input layer Num_{input} is determined by the history data we want to consult for helping us in predicting the future traffic situation. Thus, the reference data set is for the following intervals $\{(t_{now} - 1 \times t_{int}), (t_{now} - 2 \times t_{int}), ..., (t_{now} - Num_{input} \times t_{int})\}$ and the predicted traffic flow is only one output for $(t_{now} + t_{int})$. The number of nodes in hidden layers N_h is calculated by Equation 4.7.

$$N_h = \frac{N_s}{\alpha \times (N_i + n_o)} \tag{4.7}$$

- N_i: number of input nodes.
- N_o: number of output nodes.
- N_s: number of training samples.
- α: scaling factor, which can be elastically adjusted by users, usually 2-10.

The weight adjustment method described in previous section can be applied to an BPNN constructed using the above parameters so as to dynamically train it to model different traffic scenarios (peak traffic, off-peak traffic, etc). The trained BPNN can then be used as a predictor for traffic flow. Different predictors might be required for different traffic scenarios. Feeding history traffic data into the trained prediction models, future traffic flow can thus be predicted.

4.2 Optimization Method Design

We adapt GA as the method to optimize and adjust the traffic signal timings. We consider the queue length of roads in the same direction with traffic signal, the average arrival rate of roads in the vertical direction, and the previous green time as the factors to design our fitness function. Figure 4.8 shows the framework of GA-based traffic light optimization method.

4.2.1 Parameter Settings and Operations in GA

Following are the parameters and operations in the GA. The three typical operations used in GA including crossover, mutation and selection are designed and described here. The parameters and example setting of values in our experiments are also given.

- Chromosomes

  Table 4.2 shows the format of our chromosomes. For calculating the green signal time of intersection A in Figure 4.9, the adjacent intersections B, C, D, and E are also considered. The values of the chromosome content are randomly set from zero to a preset maximum time.
- Population size: 100
- Crossover rate: 0.8
- Number of crossover points: 1
- Mutation rate: 0.1
- Mutation method: Uniform
- Selection method: Roulette wheel
- Replace: 30% for retention, 70% for replacement
- Generation period: 30

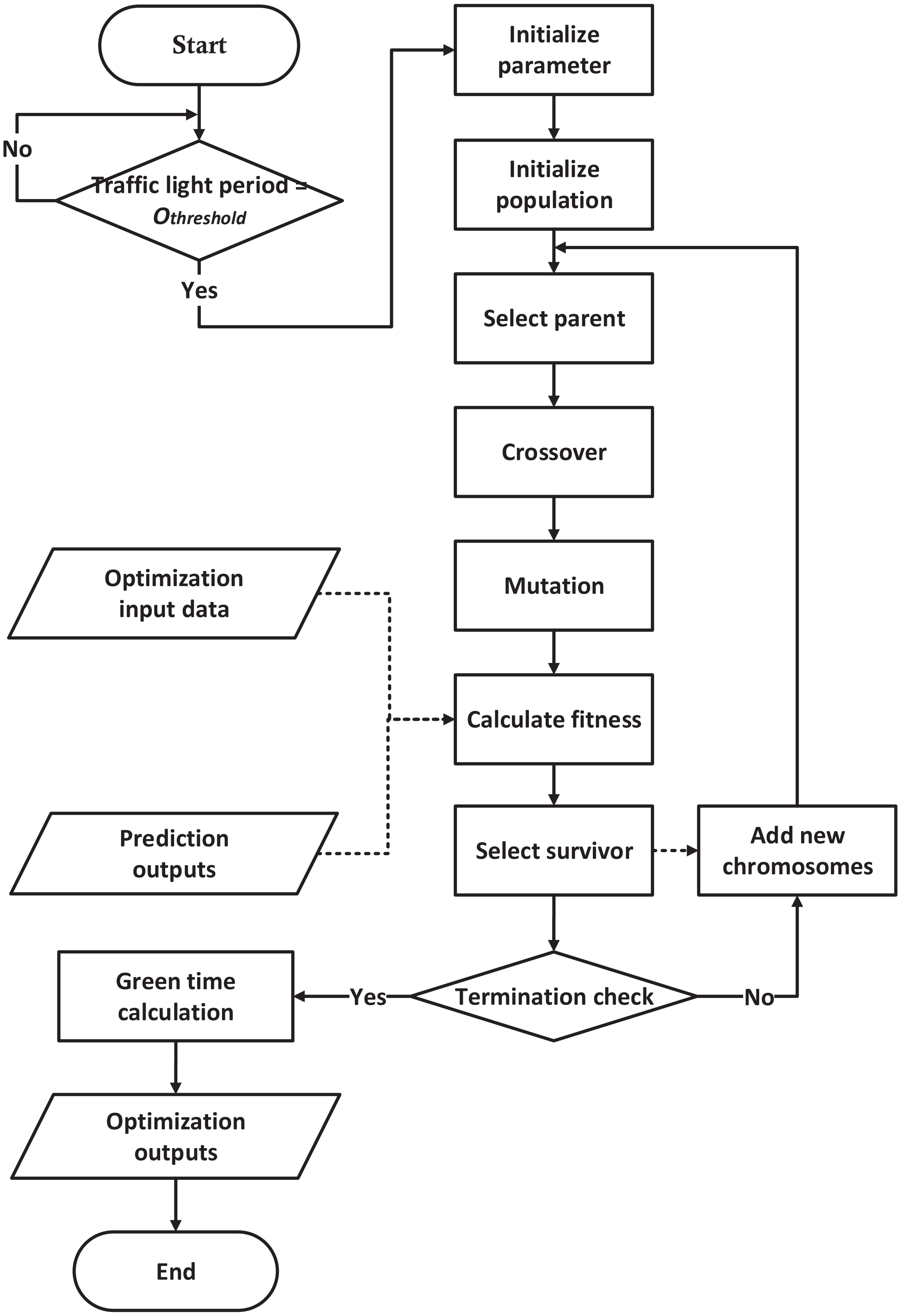

Figure 4.8: GA optimization method framework design.

Algorithm 2: Optimization Method Algorithm.

Input:
Q_{right}, Q_{left}: Number of vehicles waiting in the parallel direction of the traffic signal we want to optimize;
AVG_{up}, AVG_{down}: Average arrival rate of vehicles in the perpendicular direction of the traffic signal we want to optimize;
GT_{p_x}: Green signal time at intersection x in the previous traffic signal period;
TFP: Time for a vehicle to pass an intersection;
GT_{ref}: Reference green signal times ; `// Equation 4.13`
Output:
GT_{final_EW}: Final green signal time for the East-West direction;
GT_{final_NS}: Final green signal time for the North-South direction;
Variable:
N_{gen}: Generation of GA;
$Chrom$: Chromosome of GA;
GT_{GA}: The result of GA optimization;
GT_{upd}: Updated green light time;
GT_{nor}: Normalized green light time;

begin
  Randomly initialize chromosome, $Chrom$;
  for $i = 1$ **to** N_{gen} **do**
    `// GA calculation steps, Section 4.2.1`
    $SelectParent(Chrom)$;
    $Crossover(Chrom)$;
    $Mutation(Chrom)$;
    $CalFitness(Chrom, Q_{right}, Q_{left}, AVG_{up}, AVG_{down}, GT_{p\_x}, TFP)$;
    $SelectSurvivor(Chrom)$;
    $AddNewChrom(Chrom)$;
  $GT_{GA} = SelectBest(Chrom)$;
  $GT_{upd} = CalUpd(GA_{GA}, GT_{ref})$; `// Equation 4.14`
  $GT_{nor} = AdjustSplit(GT_{upd\_EW}, GT_{upd\_NS})$; `// Equation 4.15 and Equation 4.16`
  $GT_{final} = BoundaryCheck(GT_{nor\_EW}, GT_{nor\_NS})$; `// Equation 4.17 and Equation 4.18`
  return GT_{final_EW} and GT_{final_NS};

Table 4.2: Chromosome design.

Contents	0-Max	0-Max	0-Max	0-Max	0-Max
Intersection	Intersection A (self)	Intersection B (right 1st)	Intersection C (left 1st)	Intersection D (right 2nd)	Intersection E (left 2nd)

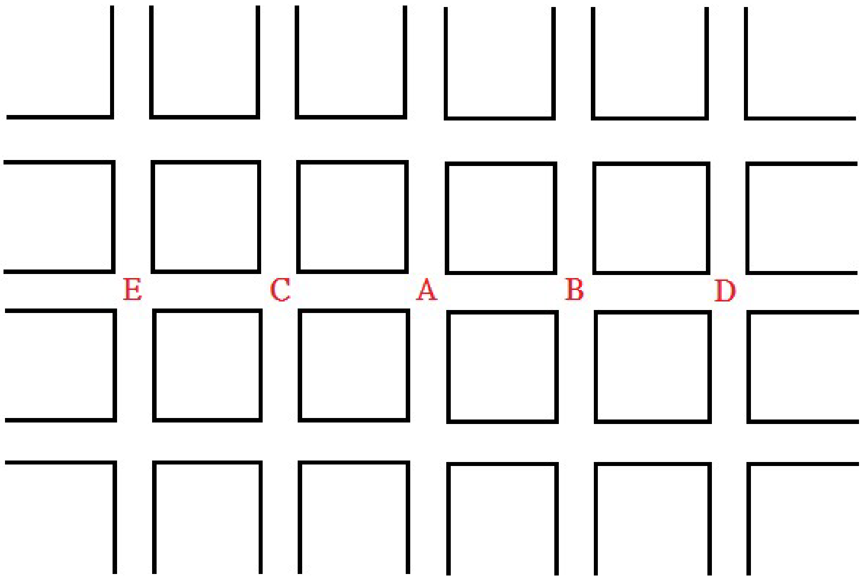

Figure 4.9: Diagram of intersections relation.

For doing the GA in our proposed method, we have to randomly generate 100 chromosomes at beginning. There are three main operations: selection, crossover, and mutation. The first step, we select the parents have good fitness values for reproduction. Then using these selected chromosomes to do crossover and mutation. The crossover method used in our method is 1-point crossover which means two parents both have a single crossover point and they are swapped by this point for recombination. The mutation step is used for genetic diversity by replacing one of the chromosome fragment (gene) in our method. The probability of crossover and mutation is set to 0.8 and 0.1. Finally, using "Roulette wheel" method to select the high quality chromosomes which have lower fitness values in our method for next generation, and replace 70% of population with new random chromosomes. Fitness function will be introduced in the following section.

4.2.2 Fitness Function Design in GA

In GA, we need to use traffic related parameters to calculate the fitness values which can help us to find approximate optimal solutions. The objective function in our proposed GA method is to minimize the fitness value, which is calculated as follows.

$$Fitness = E_1 \times C_1 + E_2 \times C_2 + E_3 \times C_3 + E_4 \times C_4 \tag{4.8}$$

$$E_1 = |GT_{n\_A} - Q_{right} \times TFP| - |GT_{n\_A} - Q_{left} \times TFP| \tag{4.9}$$

$$E_2 = (GT_{n\_A} \times AVG_{up}) + (GT_{n\_A} \times AVG_{down}) \tag{4.10}$$

$$E_3 = (|GT_{n\_B} - GT_{n\_A}| + |GT_{n\_B} - GT_{p\_B}|) + (|GT_{n\_C} - GT_{n\_A}| + |GT_{n\_C} - GT_{p\_C}|) \tag{4.11}$$

$$E_4 = (|GT_{n\_D} - GT_{n\_A}| + |GT_{n\_D} - GT_{p\_D}|) + (|GT_{n\_E} - GT_{n\_A}| + |GT_{n\_E} - GT_{p\_E}|) \tag{4.12}$$

where GT_{n_x} is the chosen green signal time at intersection x in the current period, GT_{p_x} is the green signal time at intersection x in the previous period, Q_{right}, Q_{left} are the number of vehicles waiting on the right and left, (the two approaches in the horizontal direction), respectively, of the intersection x, where $x \in \{A, B, C, D, E\}$, TFP is the time for a vehicle to pass an intersection, and AVG_{up}, AVG_{down} are the average arrival time of vehicles on the two approaches

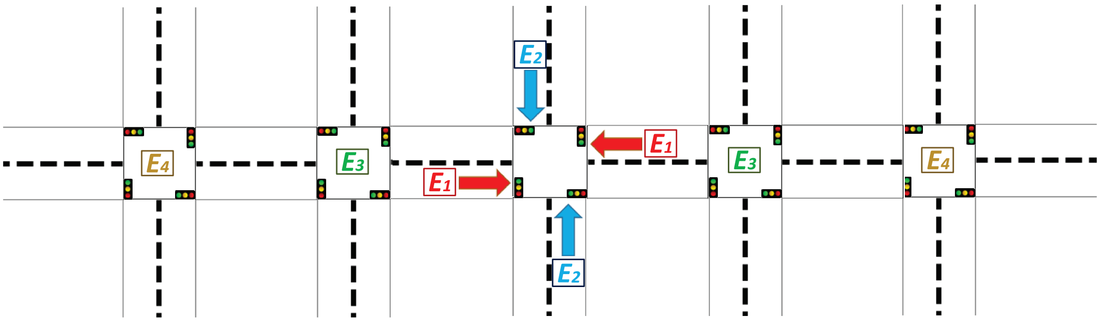

Figure 4.10: Diagram of fitness function.

(up, down) in the vertical direction.

Figure 4.10 shows the corresponding locations of fitness function. Equation 4.9 tries to find a green split such that the chosen GT_{n_A} minimizes the difference between meeting the demands of green time by the left approach $(Q_{right} \times TFP)$ and by the right $(Q_{left} \times TFP)$ approach. Equation 4.10 tries to minimize the total number of vehicles waiting in the up and down approaches due to red light in the vertical direction (green light in the horizontal direction). Equation 4.11 tries to minimize the time difference between adjacent green lights $(|GT_{n\_B} - GT_{n\_A}| + |GT_{n\_C} - GT_{n\_A}|)$ and between consecutive green light changes $(|GT_{n\_B} - GT_{p\_B}| + |GT_{n\_C} - GT_{p\_C}|)$. Equation 4.12 is similar to Equation 4.11, except the coordination is between signals that are two road segments apart (with one signal in-between). Since the four components E_1, E_2, E_3, and E_4 in the formula for fitness value have different ranges and different impacts on the optimal green signal value at intersection A, different normalization ratios (r), weights (w), and coefficients are assigned to each as shown in Table 4.3. Finally, we calculate the fitness value by Equation 4.8.

Table 4.3: Coefficients in fitness calculation.

	E_x		C_x	
x	range	ratio (r)	weight (w)	$r \times w$ (coefficient)
1	0-45	9	4	36
2	0-112	4	3	12
3	0-150	3	2	6
4	0-150	3	1	3

4.2.3 Green Signal Time Calculation

The green signal time is calculated as shown in Figure 4.11, where the steps are as follows.

1. The optimization result GT_{GA} is a 5-tuple (GT_{n_A}, GT_{n_B}, GT_{n_C}, GT_{n_D}, GT_{n_E}), where the latter four are called new reference green signal time for adjacent intersections B, C, D, and E.

2. The reference green signal times (GT_{oldref_x}) of neighbor intersections $x \in \{B, C, D, E\}$ are updated by taking the average of the old and new reference values. Note that the new reference values are from the optimization result in the previous step.

$$GT_{ref\_x} = (GT_{oldref\_x} + GT_{newref\_x}) \div 2 \tag{4.13}$$

3. Update the green signal time for intersection A as follows.

$$GT_{upd\_A} = (GT_{n\_A} \times 3 + GT_{ref\_A} \times 1) \div 4 \tag{4.14}$$

4. Suppose the two updated green signal times for the East-West and North-South approaches are $GT_{upd\_A\_EW}$ and $GT_{upd\_A\_NS}$, respectively. To calculate the green split for a fixed cycle length, these values are then normalized

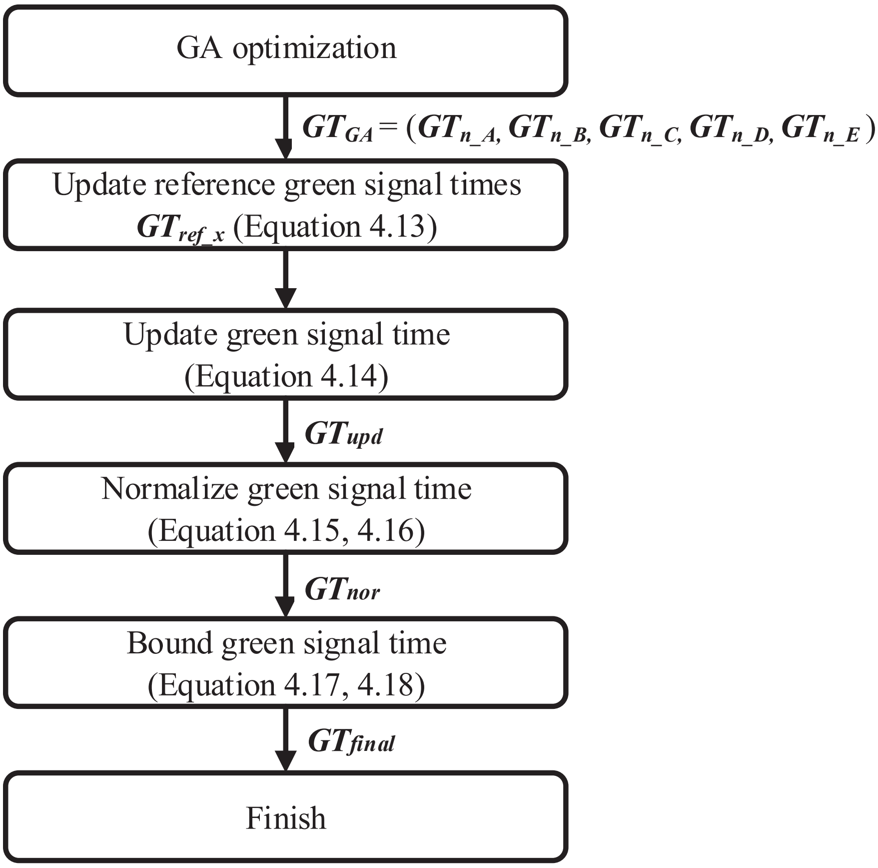

Figure 4.11: Post processing after GA.

with respect to the period length (P). In other words, the normalized green times are calculated as follows.

$$GT_{nor\_A\_EW} = GT_{upd\_A\_EW} \div (GT_{upd\_A\_EW} + GT_{upd\_A\_NS}) \times P \quad (4.15)$$

$$GT_{nor\_A\_NS} = GT_{upd\_A\_NS} \div (GT_{upd\_A\_EW} + GT_{upd\_A\_NS}) \times P \quad (4.16)$$

5. For traffic safety and smooth optimization, we need to ensure the difference between two successive time changes is not too large. In this work, the difference between two successive periods is limited to L_{diff} seconds. Further, for traffic safety and smooth traffic flow, the green signal time is also bounded

between GT_{max} and GT_{min} seconds. The final green signal time given to the simulator is denoted as $GT_{final\_A\_EW}$, $GT_{final\_A\_NS}$.

$$GT_{final\_A} = \begin{cases} GT_{nor\_A}, & \text{if } \left|GT_{pre\_A - GT_{nor\_A}}\right| \leqslant L_{diff} \\ GT_{pre\_A} - L_{diff}, & \text{if } \left|GT_{pre\_A - GT_{nor\_A}}\right| > L_{diff}, GT_{pre\_A} > GT_{nor\_A} \\ GT_{pre\_A} + L_{diff}, & \text{if } \left|GT_{pre\_A - GT_{nor\_A}}\right| > L_{diff}, GT_{pre\_A} < GT_{nor\_A} \end{cases} \tag{4.17}$$

$$GT_{final\_A} = \begin{cases} GT_{max}, & \text{if } GT_{final\_A} > GT_{max} \\ GT_{min}, & \text{if } GT_{final\_A} < GT_{min} \end{cases} \tag{4.18}$$

4.3 MPC-based Control Method Design

Given the above described prediction and optimization methods, we can dynamically adjust the traffic signal timings, instead of traditional fixed time system. However, we propose to not only just use a single prediction result to control the traffic signal, but a more time-spanning method based on *Model Predictive Control* (MPC). Figure 4.12 shows the proposed MPC-based control method.

Algorithm 3: MPC-based Control Method Algorithm.

Input:
T_{opt}: Optimization threshold time;
$Flow_{history}$: Historical traffic flow data;
F: Number of time intervals to be predicted;
$Data_{traffic}$: Traffic data, including queue length and average arrival rate;
Output:
GT_{set}: Final green signal time;
Variable:
$Flow_{p\_x}$: Prediction traffic flow value in the future x-th time interval;
$Flow_{p\_AVG}$: Future average traffic flow;

if $NeedOpt(T_{opt})$ **then**
    for $f = 1$ *to* F **do**
        $Flow_{p\_x} = PredictionModel(Flow_{history}, Flow_{p\_1}, ..., Flow_{p\_(x-1)})$;
    $Flow_{p\_AVG} = CalAvgPred(Flow_{p\_1}, ..., Flow_{p\_f})$; `// Equation 4.19`
    $GT_{set} = OptimizationModel(Flow_{p\_AVG}, Data_{traffic})$; `// Section 4.2`
    return GT_{set};

The total execution time of prediction and optimization is less than 0.5 second and the yellow signal time we set in our system is 2 seconds, so we can completely calculate and adjust the optimized signal time for next period just in yellow signal time interval.

If reaching the optimization threshold time T_{opt} we set to 5 signal periods in our system, starting the MPC-based control method. Suppose the time interval we want to optimize is T_0 as shown in Figure 4.13. First, we use four historical traffic volume data in T_{-4}, T_{-3}, T_{-2}, T_{-1} to forecast one traffic volume result $Flow_{p\_0}$.

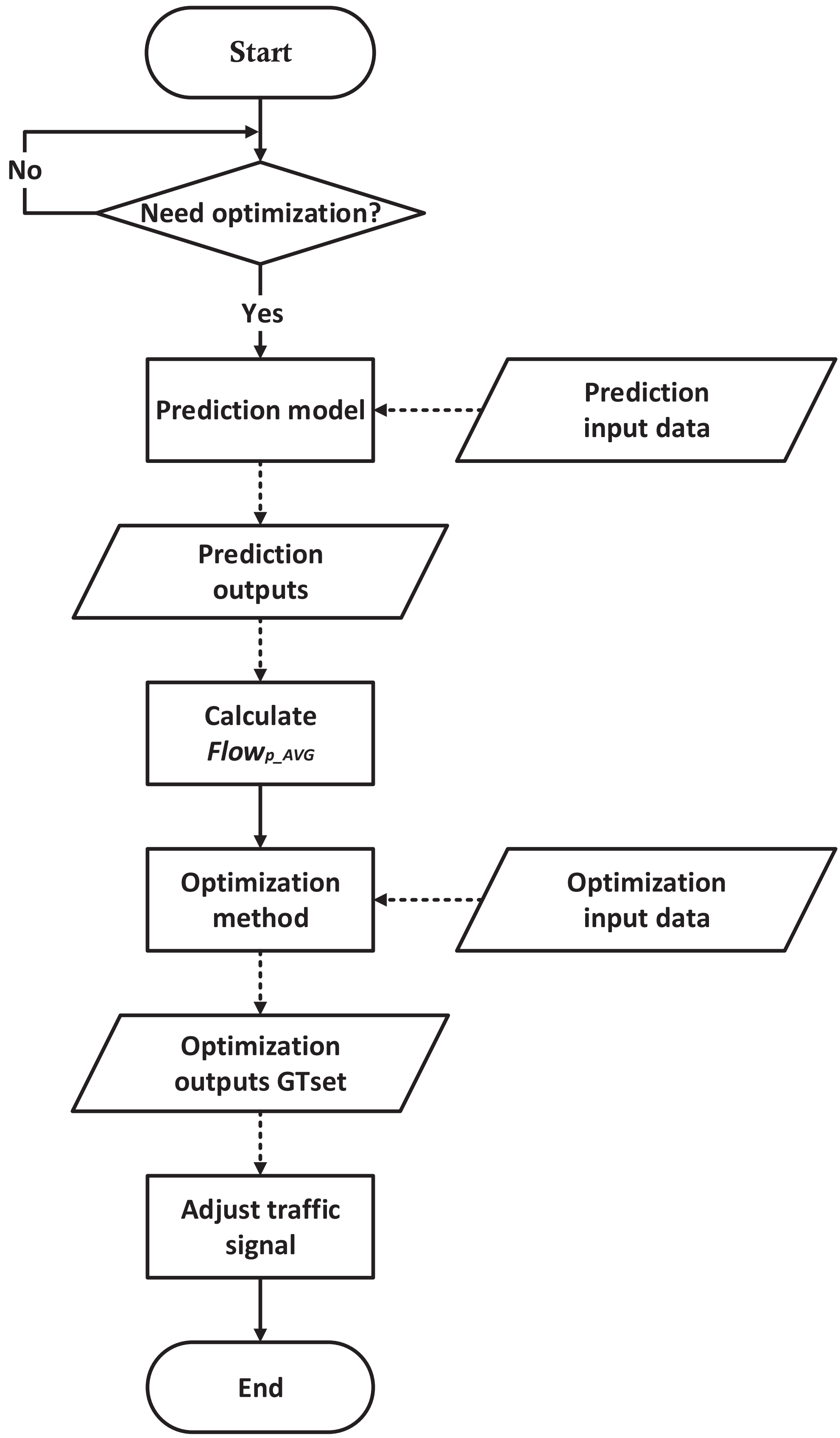

Figure 4.12: MPC-based control method framework design.

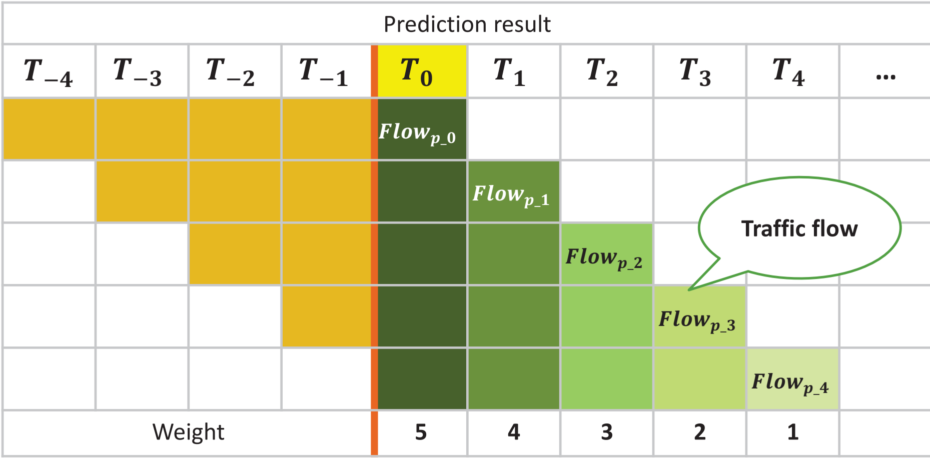

Figure 4.13: Diagram of MPC-based traffic light control method.

Then, using three history segment data in T_{-3}, T_{-2}, T_{-1} plus one prediction data in T_0 to forecast the traffic volume $Flow_{p\_1}$, and so on.

After we get all the prediction data ($Flow_{p\_0}$, $Flow_{p\_1}$, $Flow_{p\_2}$, ..., $Flow_{p\_F-1}$) we need, where F is the number of time intervals with prediction, we can use Equation 4.19 to calculate the future average traffic flow $Flow_{p\_AVG}$. It is a weighted combination of several predicted data in future time intervals, instead of just using a single predicted flow in the next time interval.

$$Flow_{p\_AVG} = \frac{\sum_{f=1}^{F}(weight_f \times Flow_{p\_f})}{\sum_{f=1}^{F} f} \tag{4.19}$$

$$weight_f = F - f + 1 \tag{4.20}$$

Finally, as shown in Equation 4.21, we input the average prediction flow values $Flow_{p\_AVG}$ and other required traffic data (queue length, past green time) into the traffic signal optimization method proposed in Section 4.2. The $Flow_{p\_AVG}$ is used as the AVG_{up} and AVG_{down} in Equation 4.10. Then we can get the optimization output as final setting time GT_{set}.

$$GT_{set} = OptimizationModel(Flow_{p\_AVG}, Data_{traffic}) \tag{4.21}$$

Chapter 5

Experiments

In this chapter, we present the experimental results of the proposed method MTLCS. We first introduce the experimental setup, including experimental environment and testing data sample. Next, we test the related parameter settings for our method and also give the predicted results and optimization results.

5.1 Experiment Setup

In this section, we explain the experimental environment and testing data sample used in our experiments.

5.1.1 Experimental Environment

Table 5.1: Experimental environment.

CPU	Intel(R) Core(TM) i3-3220 CPU @ 3.30GHz
Memory	4.00GB DDR3 @ 665MHz
Operating system (OS)	Windows 7 Ultimate (64-bits)
Development tools	Microsoft Visual Studio 2010
Programming language	C# 4.0 on .NET Framework 4

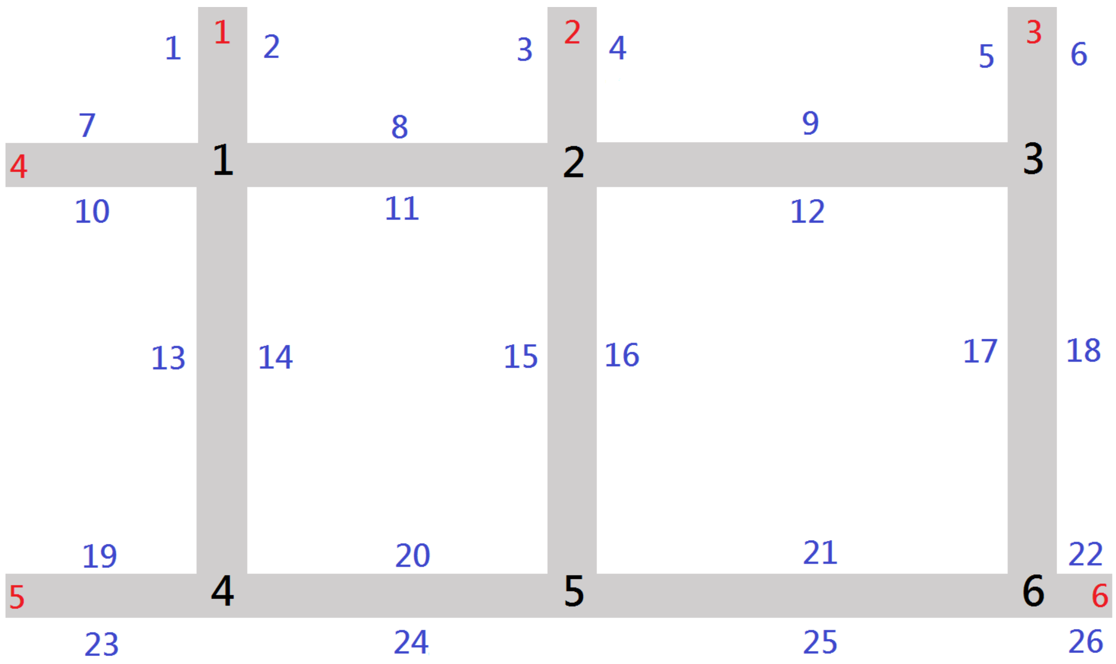

Figure 5.1: Testing map used in our experiment.

The experimental environment is shown in Table 5.1. We use a PC with an Intel(R) Core(TM) i3-3220 CPU having four cores and the frequency is 3.30 GHz. There are 4.00 GB memory. We use 64-bits Windows 7 as our operating system. Our programs were written in the programming language using C# 4.0 on Microsoft Visual Studio 2010.

5.1.2 Testing Platform and Data

The testing map used in our experiment is shown as Figure 5.1, including 6 intersections (No.1∼No.6) , 26 roads (No.1∼No.26), and 6 entry points (No. 1∼No.6). Table 5.2 shows the expected values (λ) of the Poisson distribution. The values mean the expected values of generated vehicles per minute in the entry points. Figure 5.2 shows the traffic volume sample used in our experiment to test and verify our proposed method . The characteristic is that it has two peaks during rush hours similar to the general urban traffic pattern.

Table 5.2: Vehicle expected values (λ) of Poisson distribution in the entry points.

Time (h) \ Entry points	Parameter λ in Poisson distribution					
	1	2	3	4	5	6
00:00	1	1	1	1	1	1
01:00	1	1	1	1	1	1
02:00	1	1	1	1	1	1
03:00	1	1	1	1	1	1
04:00	1	1	1	1	1	1
05:00	1	1	1	1	1	1
06:00	3	3	3	2	2	3
07:00	5	5	5	3	3	5
08:00	7	7	7	3	3	7
09:00	5	5	5	3	3	5
10:00	3	3	3	3	3	3
11:00	3	3	3	3	3	3
12:00	3	3	3	3	3	3
13:00	3	3	3	3	3	3
14:00	3	3	3	3	3	3
15:00	3	3	3	3	3	3
16:00	3	3	3	4	4	3
17:00	3	3	3	6	6	3
18:00	3	3	3	8	8	3
19:00	3	3	3	6	6	3
20:00	3	3	3	3	3	3
21:00	2	2	2	2	2	2
22:00	2	2	2	2	2	2
23:00	1	1	1	1	1	1

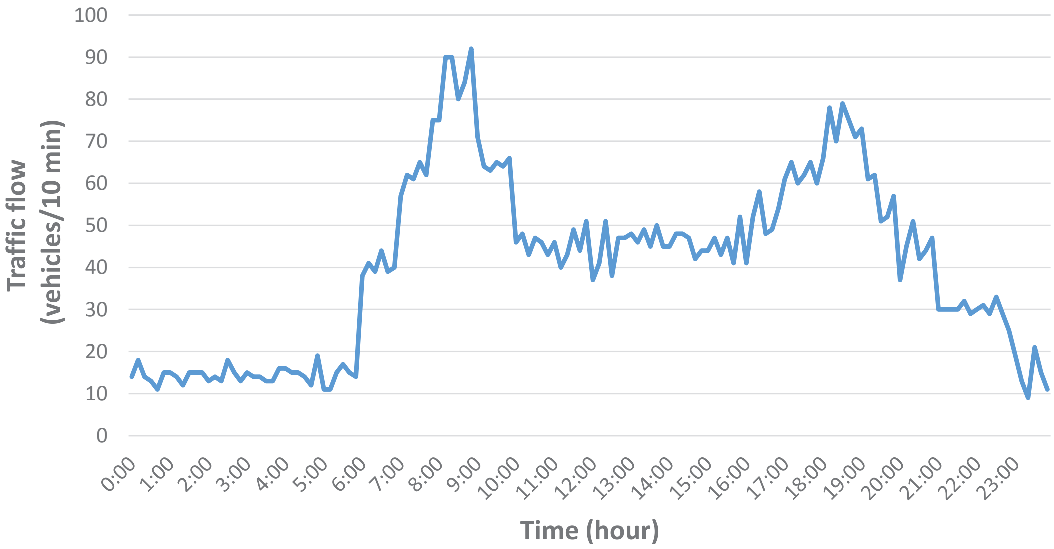

Figure 5.2: Testing traffic flow sample used in our experiment.

5.2 Experimental Results

In this section, we give the experimental results for our proposed method including the traffic flow prediction model and traffic light optimization method.

5.2.1 Traffic Flow Prediction Model

Traffic flow prediction model uses past traffic data to predict future traffic volumes. We test and compare three kinds of different variables for our prediction model and also present the complete prediction results.

Following are two target values to assess the benefits of the prediction model. Training error E shown in Equation 5.1 is used to estimate the error for training. The *Mean absolute percentage error* (MAPE) shown in Equation 5.2 is used to estimate the difference between actual values and predictive values for prediction. We use "10 minutes" traffic volume data as the training and prediction data in the following experiments.

$$E = \frac{1}{M}\sum_{m=1}^{M}(A_m - P_m)^2 \tag{5.1}$$

$$MAPE = \frac{1}{N} \sum_{n=1}^{N} \left| \frac{A_n - P_n}{A_n} \right| \tag{5.2}$$

where

- A_m: Actual traffic volume value of training sample m.
- P_m: Predicted traffic volume value of training sample m.
- M: Number of training samples.
- A_n: Actual traffic volume value of testing samples n.
- P_n: Predicted traffic volume value of testing samples n.
- N: Number of testing samples.

Table 5.3: Parameter settings of number of input nodes experiment.

Input layer node	1∼9
Hidden layer node	6
Output layer node	1
Training sample	350
Prediction sample	140
Training cycle	60
Learning rate	0.8

Number of Training Inputs

Number of training inputs means the time slots considered to train the BPNN model. For example, we use 10 minutes traffic volume data as training data, so using three training inputs means we consider the previous 30 minutes traffic volume data to train the model. This experiment is to measure the number of input nodes in BPNN. Table 5.3 shows the parameter settings for this experiment.

We measure the MAPE while varying the number of inputs from 1 to 9. Table 5.4 and Figure 5.3 show the results. We can observe that when the number of input nodes is less than 3 and when it is more than 5, the prediction results are quite inaccurate. It means that if we use too few previous traffic volume data, we cannot get an accurate prediction. On the other hand, if we consider too much previous traffic volume data, the results may be impacted by the irrelevant information from a too long historical time period. When the number of nodes are 3, 4, or 5, the proposed prediction model has a low MAPE and thus give better prediction results. Thus, the number of nodes is selected as 4 in our method.

Table 5.4: MAPE of different number of input nodes in BPNN.

Number of inputs	MAPE
1	14.42%
2	13.40%
3	13.05%
4	12.55%
5	12.80%
6	13.36%
7	13.97%
8	14.26%
9	14.92%

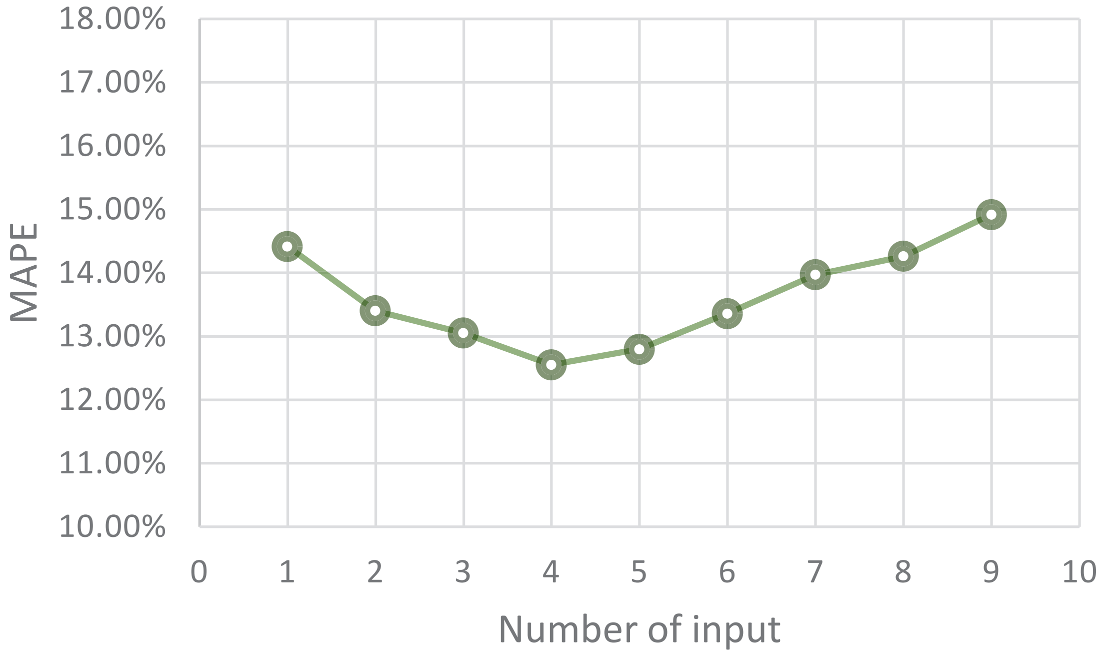

Figure 5.3: Trend chart of MAPE using different number of input nodes.

Table 5.5: Parameter settings of training sample experiment.

Input layer node	4
Hidden layer node	6
Output layer node	1
Training sample	50∼500
Prediction sample	140
Training cycle	60
Learning rate	0.8

Number of Training Samples

A training sample means a set of historical traffic volume data, including multiple training inputs and a desired output in continuous time sequence. For example, a training sample contains 5 data ($Input_a$, $Input_b$, $Input_c$, $Input_d$, $Output_a$) means it has 4 training inputs and 1 desired output. We measure the prediction efficacy using 50 to 500 samples. Table 5.5 shows the parameter settings for this experiment.

Table 5.6 and Figure 5.4 show the training errors and the MAPE results for this experiment. Observing these two data, too few training samples cause very large errors. It means the lack of training samples will lead to inaccurate prediction results. When the number of training samples increases to more than 350, the training errors approach a stable state. Thus the number of training samples is enough when the number of samples is greater than 350.

Table 5.6: Training errors and MAPE of different number of training samples.

Training Sample	Training Error	MAPE
50	0.052725	46.23%
100	0.021129	40.04%
150	0.010002	25.45%
200	0.007522	21.93%
250	0.005363	17.67%
300	0.005022	14.80%
350	0.005206	12.89%
400	0.004934	12.80%
450	0.004316	12.45%
500	0.004914	12.23%

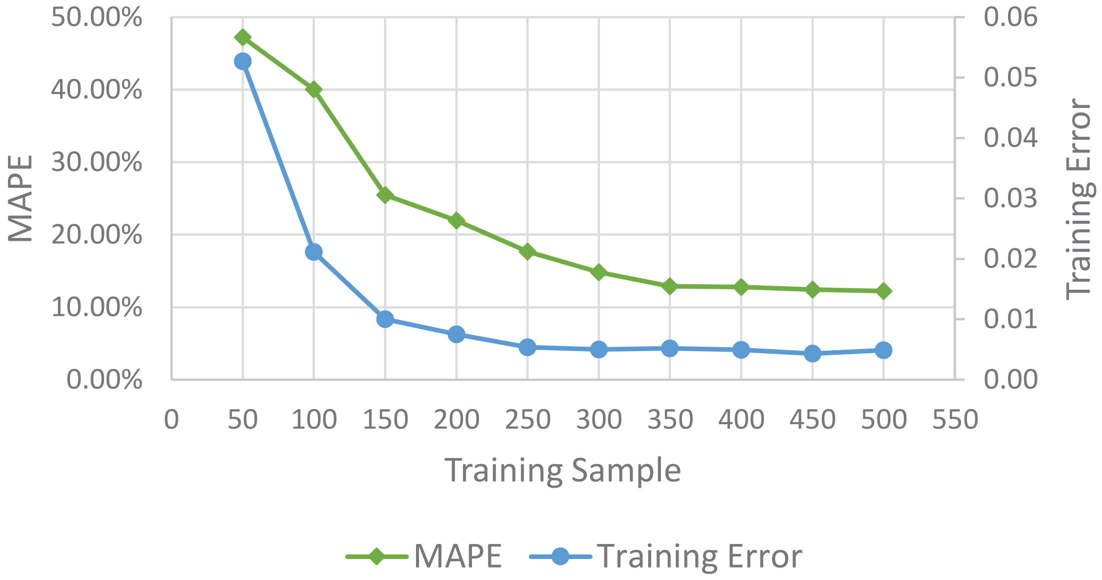

Figure 5.4: Trend chart of training errors and MAPE using different number of training samples.

Table 5.7: Parameter settings of training cycle experiment.

Input layer node	4
Hidden layer node	6
Output layer node	1
Training sample	350
Prediction sample	140
Training cycle	1∼100000
Learning rate	0.8

Number of Training Cycles

A *training cycle* of a BPNN is defined as the application of all training samples, once each, in random order. We calculate the training errors from 1 cycle to 100,000 cycles and observe the convergent condition. It can help us in determining the suitable number of training cycles setting or estimating the training completion time. Table 5.7 shows the parameter settings for this experiment.

Figure 5.5 shows the training errors of training cycles from 1 to 20. The training errors have rapid reduction from 1 to 6 cycles. Then after 6 cycles, the training errors achieve a steady state. Figure 5.6 shows the training errors of training cycles from 1000 to 100,000. We can see that the trend in the chart has a slight decline only and has some fluctuations. Thus, it means the training model has reached a convergent state. Table 5.8 shows the training errors of different training cycles from 10 to 100. We set the error threshold to 0.005 in our system and determine the training cycles more than 60 cycles is enough.

Table 5.8: Training errors of different number of training cycles from 10 to 100.

Training cycle	Training error
10	0.006465
20	0.005946
30	0.005422
40	0.005179
50	0.005022
60	0.004843
70	0.004816
80	0.004594
90	0.004679
100	0.004519

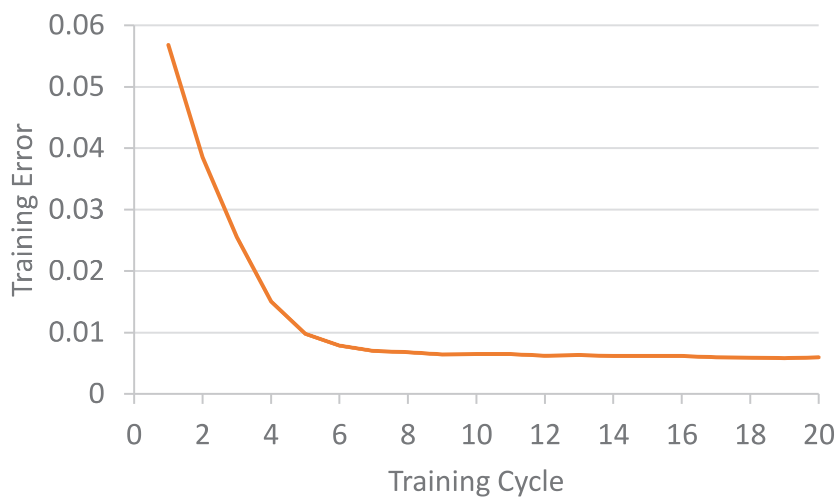

Figure 5.5: Training errors of training cycles from 1 to 20.

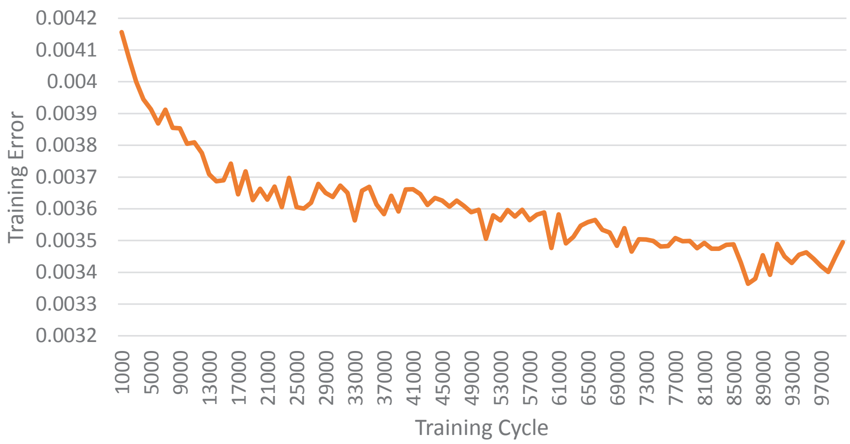

Figure 5.6: Training errors of training cycles from 1 to 100,000.

Table 5.9: Parameter settings of final prediction model experiment.

Input layer node	4
Hidden layer node	6
Output layer node	1
Training sample	350
Prediction sample	432 (3 days)
Training cycle	60
Learning rate	0.8

Prediction Results

This experiment shows the complete prediction results. Based on the above three parameter tests, our prediction model has 4 input nodes, 350 training samples, and 60 training cycles. We use 3 days traffic volume data to test and verify our model. Table 5.9 gives the parameter settings for this experiment.

In this experiment, training using above settings was performed to test the accuracy of the BPNN prediction model. The average MAPE value of experiment results is 11.59%. For the general standard of MAPE, the value less than 10% means the high accurate prediction and between 10% to 20% means the excellent prediction [40]. For the convenience of experiments, we use only one model to predict the traffic volumes during peak and off-peak time which sometimes is smooth and sometimes is varied. The average MAPE value are already close to 10% (high accurate prediction). If one day is divided into several small time sections with different prediction models which have different trained weights, then we believe the prediction results can be more accurate. Figure 5.7 shows the 3 days prediction results compared to the actual traffic volume values and we can see that the trend is very consistent.

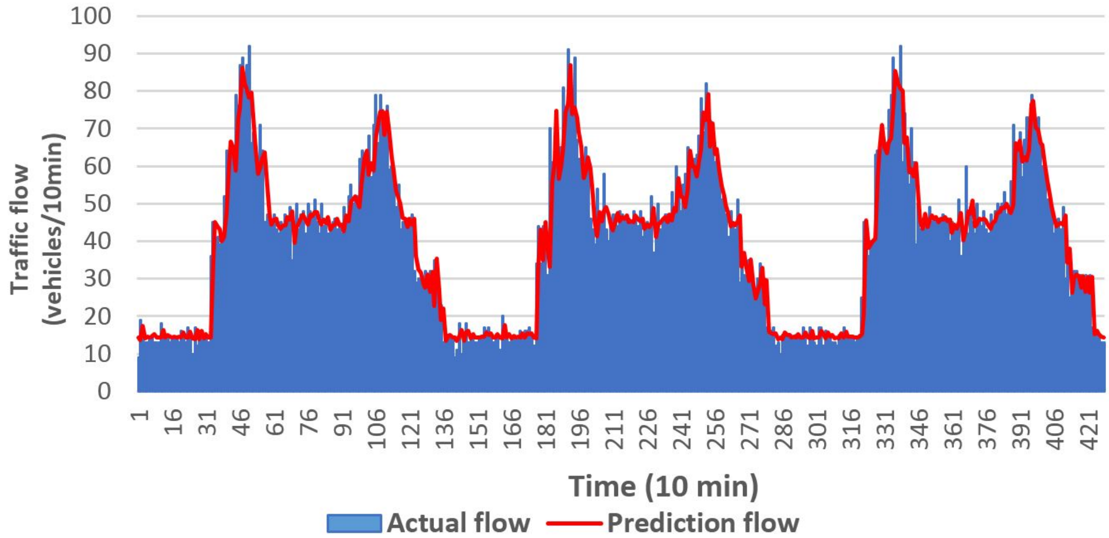

Figure 5.7: The comparison between actual flow and prediction flow.

5.2.2 Traffic Light Optimization Method

Traffic light optimization method uses the outputs of prediction model and other necessary traffic data to dynamically adjust the traffic light and improve traffic congestion.

We test the suitable parameter settings for MPC control method and also present the final optimization results compared with fixed-time system and real-time based system (RTBS) in [35] which only consider the queue length of waiting vehicles. "IAWR" and "waiting time" are used to measure the efficacy of our proposed method. IAWR was defined in Chapter 3 and waiting time means the total time of a car waits behind each red signal on the travel route.

MPC parameter experiments

This experiment tests the different number of considered future time slots for MPC method. Figure 5.8 shows the average IAWR and waiting time from 1 to 5. We can observe that when we consider more future time slots, IAWR and waiting time exhibit a greater improvement. However, 5 future time slots give a very

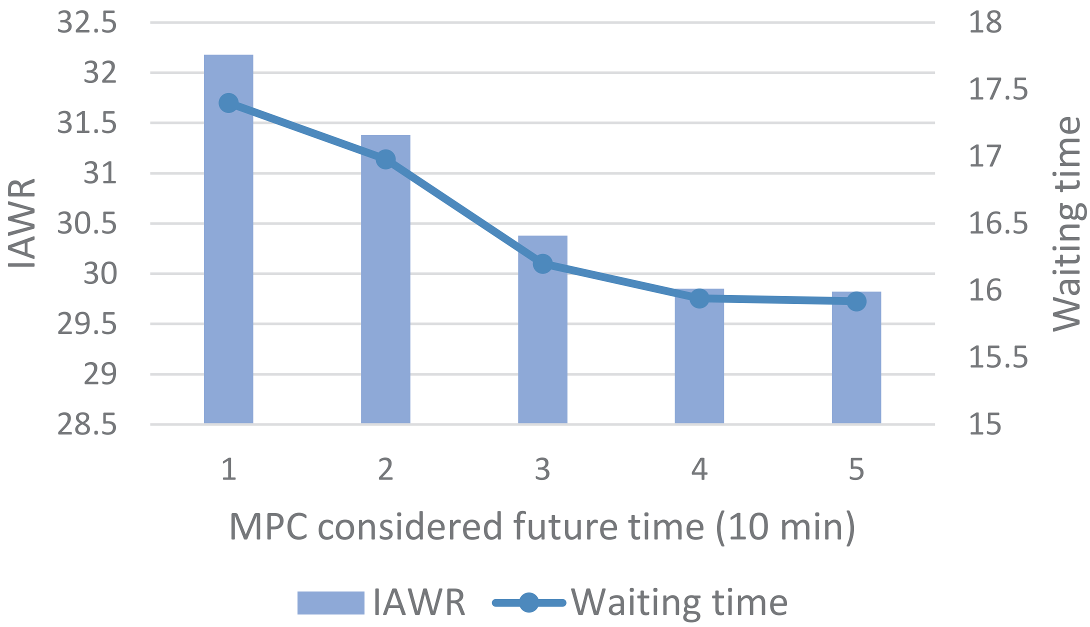

Figure 5.8: Considered future time slots for MPC method.

marginal improvement over that for 4 future time slots. Thus, we use 4 future time slots as our setting in MTLCS.

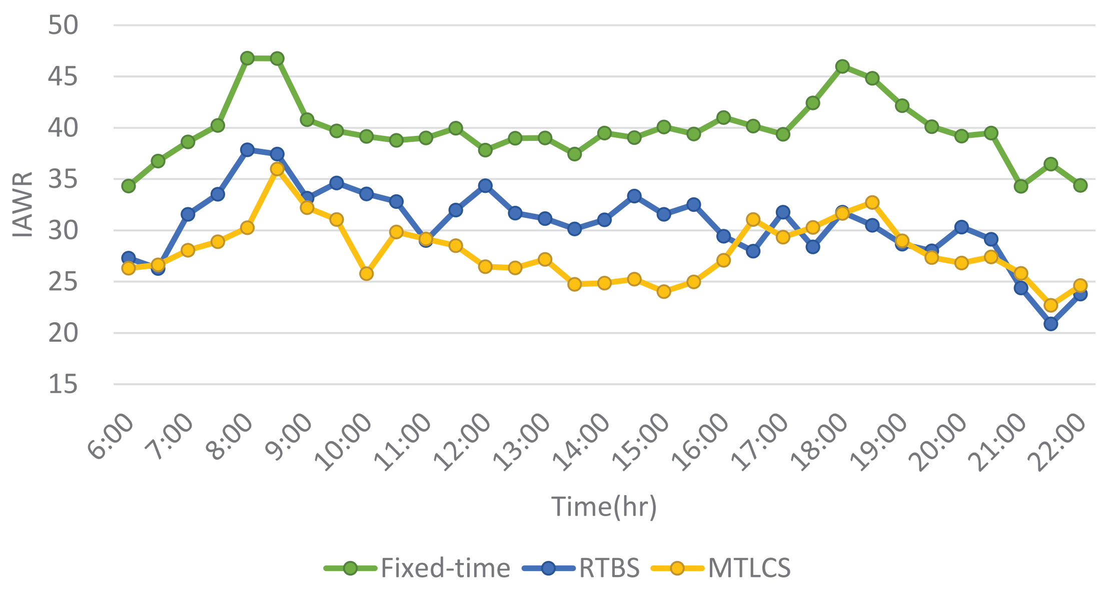

Figure 5.9: IAWR results compared with other two systems.

IAWR of Optimization Results

We measure the IAWR for our optimization method and other two systems. Figure 5.9 shows the comparison results of one day. Because of less traffic in the midnight, we only present the part from 6:00 a.m. to 10:00 p.m. The IAWR of fixed-time system is higher than RTBS and MTLCS in the whole day. Because it does not have ability to dynamically adjust the traffic signals according to the suddenly large and small traffic volumes. Most of the time, the IAWR in our system MTLCS is almost lower than that in RTBS. And we can see that our method has greater improvement in the off-peak time.

Table 5.10 and Figure 5.10 show the detailed data of this experiment. We list the IAWR of 6 intersections separately. Our method has greater improvement in the more congested intersections such as intersections 4, 5, and 6, for which reductions are more than 30%. Compared to fixed-time method, our method MTLCS gives 29.70% reduction in average IAWR, which is greater than that of RTBS (23.03%).

Table 5.10: IWAR results and reduced percentages of 6 intersections.

	1	2	3	4	5	6	Average
Fixed-time	36.72	30.35	35.98	48.56	46.19	40.75	39.76
RTBS	31.04 (-15.47%)	28.17 (-7.18%)	30.06 (-16.45%)	31.08 (-36.00%)	32.77 (-29.05%)	30.51 (-25.13%)	30.60 (-23.03%)
MTLCS	27.13 (-26.12%)	23.49 (-22.60%)	28.83 (-19.87%)	29.91 (-38.41%)	29.97 (-35.12%)	28.38 (30.36%)	27.95 (-29.70%)

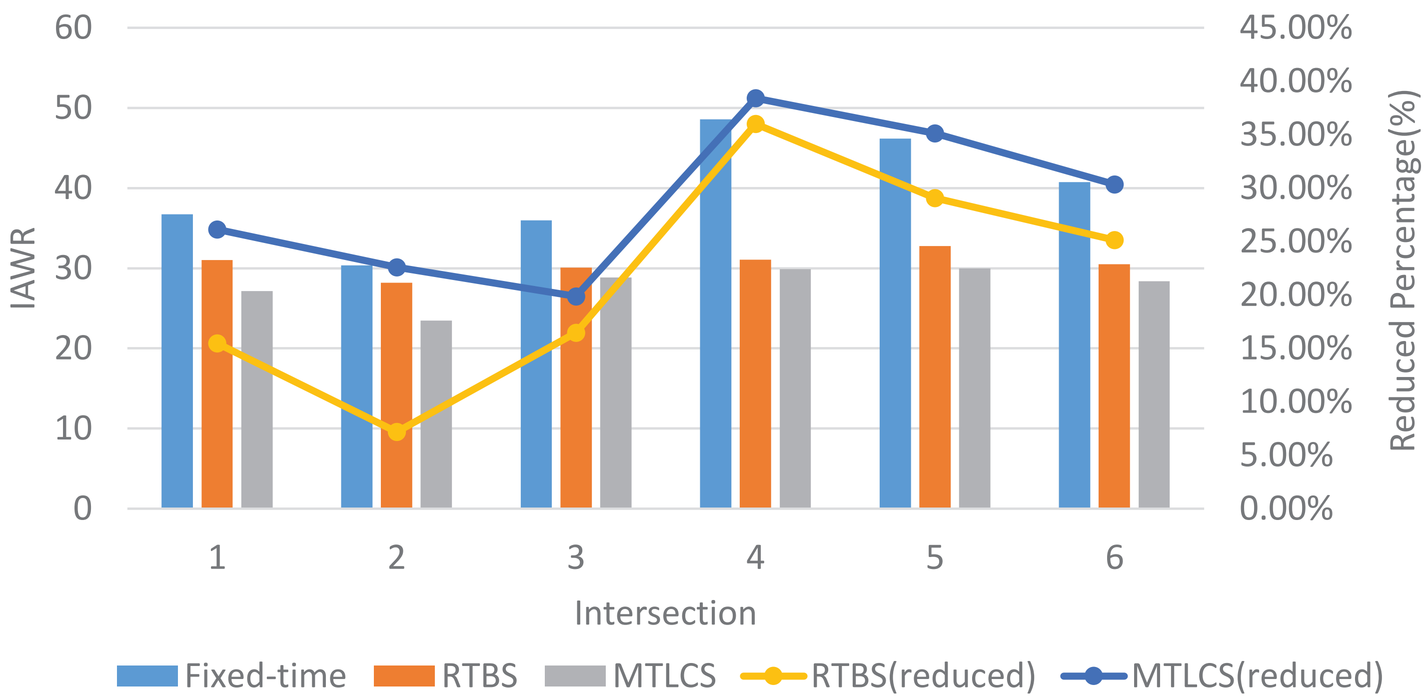

Figure 5.10: Comparison chart of IAWR results of 6 intersections.

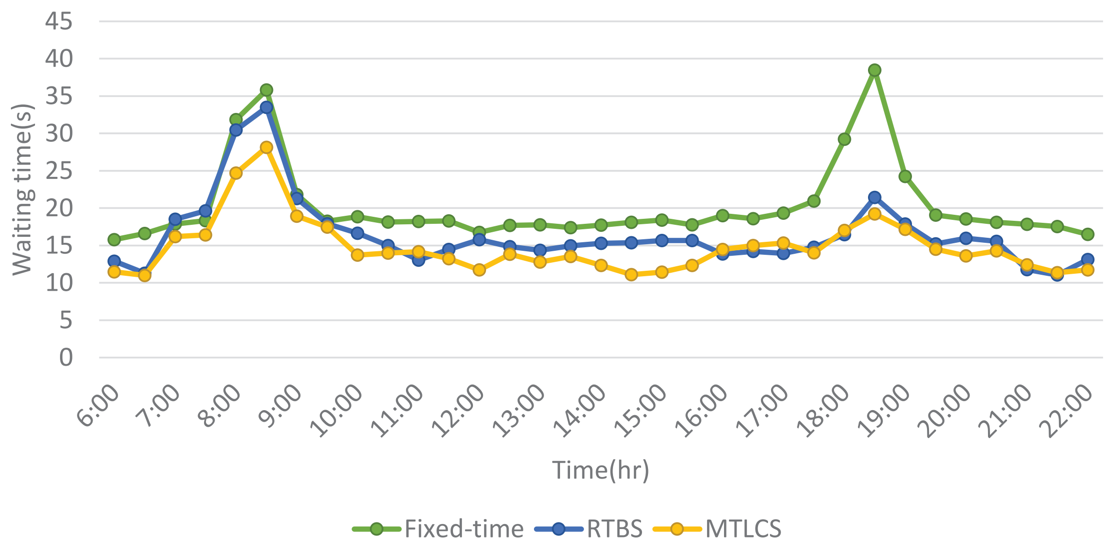

Figure 5.11: Waiting time results compared with two other systems.

Waiting Time of Optimization Results

We also calculate the average waiting time of all cars going through the map. Figure 5.11 shows the comparison results. The waiting time of our method MTLCS and RTBS is less than that in the fixed-time system. But MTLCS is slightly lower than the RTBS in most of the time.

Table 5.11 and Figure 5.12 show the detailed data of this experiment. We divide one day into several time sections including two peak sections and two off-peak sections. We can observe that our method has the greater reduction of waiting time in peak 2 (17:00∼20:00) with 35.77%. Compared with Figure 5.9, IAWR also has more improvement in this time section. Finally, the average waiting time is decreased up to 26.93% in MTLCS and greater than 19.00% in RTBS.

Table 5.11: Waiting time results and reduced percentages of 4 divided time sections.

	Peak 1 (6:00∼10:00)	Off peak 1 (10:00∼17:00)	Peak 2 (17:00∼20:00)	Off peak 2 (20:00∼22:00)	Average
Fixed-time	22.04	18.05	25.22	17.71	20.27
RTBS	20.68 (-6.17%)	14.94 (-17.23%)	16.61 (-34.14%)	13.50 (-23.77%)	16.42 (-19.00%)
MTLCS	18.04 (-18.15%)	13.12 (-27.31%)	16.20 (-35.77%)	12.69 (-28.35%)	14.81 (-26.93%)

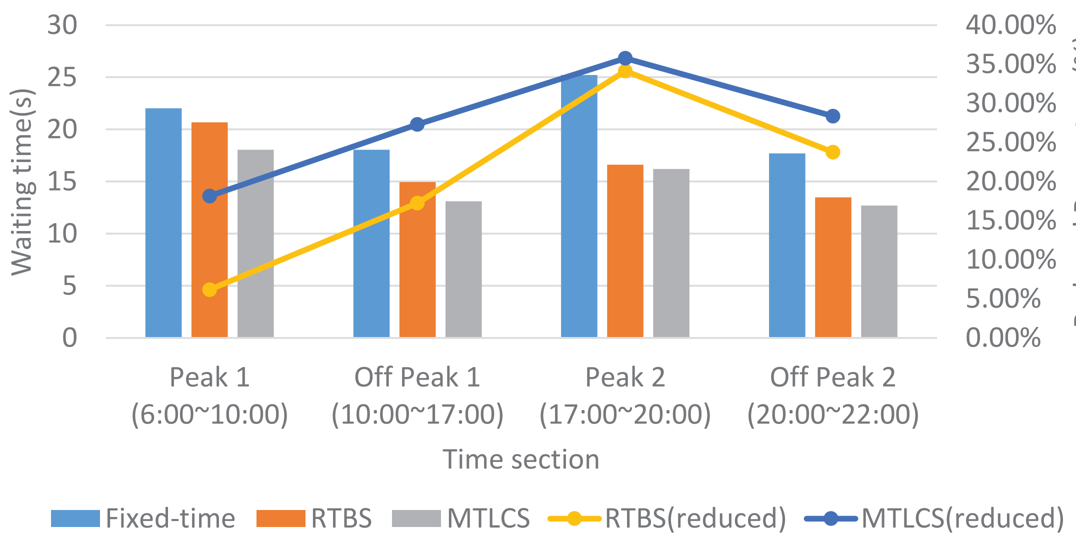

Figure 5.12: Comparison chart of waiting time of 4 divided time sections.

Chapter 6

Conclusions and Future Work

In this Thesis, we proposed a MPC-based traffic light optimization system (MTLCS) to reduce traffic congestion. Our proposed method includes two main models, which are traffic flow prediction model and traffic light optimization method.

Traffic flow prediction model is based on BPNN for predicting traffic volume in the future. Experimental results show that 4 input nodes, 350 training samples, and 60 training cycles are appropriate settings for the prediction model. Prediction results from the proposed traffic flow prediction model has an average MAPE value of 11.59%, which means it is a highly accurate prediction model. Traffic light optimization method is based on GA for optimizing and adjusting the traffic signal times. The proposed MPC-based control method gives a lower IAWR and waiting time compared to the non-MPC systems. The number of future time slots considered suitable for optimization was found to be four. MTLCS results in a reduction of 29.70% in average IAWR and a reduction of 26.93% in average waiting time. Both of these results are better than the other two compared systems.

In the future, for safety considerations, we can consider buffer time when switching the traffic signals in the vertical direction. In addition, besides adjusting traffic signals, we can also consider travel time predictions, routing plans, and navigation from drivers point of view to reduce traffic congestion. We can also consider improving the fuel consumption to reduce environmental impact.

Bibliography

[1] J. D. Carvell, E. J. Seymour, C. H. Walters, and T. R. Starr. Dallas Area-Wide Intelligent Transportation System Plan. Technical Report FHWA/TxDOT-96/591-1F, Texas Transportation Institute, July 1996.

[2] Taiwan registered vehicle amount. `http://www.thb.gov.tw/`, 1998-2014.

[3] E. F. Camacho and C. B. Alba. *Model Predictive Control*. Springer Science & Business Media, Germany, May 2007.

[4] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 593–605. IEEE, June 1989.

[5] L. D. Davis. *Handbook of Genetic Algorithms*, volume 115. Van Nostrand Reinhold, New York, January 1991.

[6] Intelligent Transportation System. `http://www.its.dot.gov/`, 2012.

[7] W. Kim, Y. Liu, and G. L. Chang. Advanced traffic management system. *Transportation Research Record: Journal of the Transportation Research Board*, 2324(1):91–100, 2012.

[8] W. L. Chang. Signal Control at Isolated Intersection with Dynamic Timing Computation Strategy. Master's thesis, National Central University, Taiwan, June 2003.

[9] K. C. Hsu. *Adaptive Traffic Signal Control Model with Self-Learning Capability*. PhD thesis, National Cheng Kung University, Taiwan, January 2003.

[10] I. Yun and B. Park. *Stochastic optimization method for coordinated actuated signal systems*. Center for Transportation Studies, University of Virginia, USA, December 2005.

[11] P. C. Hsieh, Y. R Chen, W. H. Wu, and P. A. Hsiung. Timing Optimization and Control for Smart Traffic. In *Proceedings of the IEEE International Conference on Internet of Things (iThings)*, pages 9–16. IEEE, September 2014.

[12] Kaohsiung road daily flow in 2005. `http://210.241.35.26/view.asp?func=2`, 2005.

[13] K. D. Kim and P. R. Kumar. An MPC-based Approach to Provable System-wide Safety and Liveness of Autonomous Ground Traffic. *IEEE Transactions on Automatic Control*, 59(12):3341–3356, December 2014.

[14] A. Khodayari, A. Ghaffari, M. Nouri, S. Salehinia, and F. Alimardani. Model Predictive Control system design for car-following behavior in real traffic flow. In *Proceedings of the International Conference on Vehicular Electronics and Safety (ICVES)*, pages 87–92. IEEE, July 2012.

[15] J. R. D. Frejo, A. Nunez, B. D. Schutter, and E. F. Camacho. Model predictive control for freeway traffic using discrete speed limit signals. In *Proceedings of the European Control Conference (ECC)*, pages 4033–4038. IEEE, July 2013.

[16] S. K. Zegeye, B. D. Schutter, and J. Hellendoorn. Model predictive traffic control to reduce vehicular emissions-an LPV-based approach. In *Proceedings of the American Control Conference (ACC)*, pages 2284–2289. IEEE, June 2010.

[17] T. Tettamanti, I. Varga, B. Kulcsar, and J. Bokor. Model predictive control in urban traffic network management. In *Proceedings of the 16th Mediterranean Conference on Control and Automation*, pages 1538–1543. IEEE, June 2008.

[18] X. Zhou and Y. Lu. Coordinate model predictive control with neighbourhood optimisation for a signal split in urban traffic networks. *Intelligent Transport Systems(IET)*, 6(4):372–379, December 2012.

[19] D. Chung, X. Rui, D. Min, and H. Yeo. Road traffic big data collision analysis processing framework. In *Proceedings of the 7th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–4, October 2013.

[20] S. Lin, Y. Xi, and Y. Yang. Short-term traffic flow forecasting using macroscopic urban traffic network model. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems(ITSC)*, pages 134–138. IEEE, October 2008.

[21] Q. J. Kong, Y. Xu, S. Lin, D. Wen, F. Zhu, and Y. Liu. UTN-model-based traffic flow prediction for parallel-transportation management systems. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1541–1547, September 2013.

[22] Z. Zhu and Z. Yang. A Real-time Isoparametric Traffic Volume Prediction Model Based on the Kalman Filtering Theory. In *Proceedings of the IEEE*

International Vehicle Electronics Conference (IVEC), pages 102–105, September 1999.

[23] F. Guo, R. Krishnan, and J. W. Polak. Short-term traffic prediction under normal and incident conditions using singular spectrum analysis and the k-nearest neighbour method. In *Proceedings of the IET and ITS Conference on Road Transport Information and Control (RTIC)*, pages 1–6. IET, September 2012.

[24] Ya. Wang, Y. Chen, M. Qin, and Y. Zhu. Dynamic traffic prediction based on traffic flow mining. In *Proceedings of the Sixth World Congress on Intelligent Control and Automation (WCICA)*, volume 2, pages 6078–6081. IEEE, June 2006.

[25] D. M. Li and B. Liu. Modeling and prediction of highway traffic flow based on wavelet neural network. In *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, pages 675–679. IEEE, July 2014.

[26] W. Hu, Y. Liu, L. Li, and S. Xin. The short-term traffic flow prediction based on neural network. In *Proceedings of the 2nd International Conference on Future Computer and Communication (ICFCC)*, volume 1, pages 293–296. IEEE, May 2010.

[27] J. Park, D. Li, Y. L. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips. Real time vehicle speed prediction using a neural network traffic model. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 2991–2996. IEEE, July 2011.

[28] G. Tong, C. L. Fan, F. Cui, and X. Meng. Fuzzy neural network model applied in the traffic flow prediction. In *Proceedings of the International Conference on Information Acquisition*, pages 1229–1233. IEEE, August 2006.

[29] Z. Li, Q. Lei, X. Kouying, and Z. Xinyan. A novel BP neural network model for traffic prediction of next generation network. In *Proceedings of the Fifth International Conference on Natural Computation (ICNC)*, volume 1, pages 32–38. IEEE, August 2009.

[30] K. Tawara and N. Mukai. Traffic Signal Control by Using Traffic Congestion Prediction Based on Pheromone Model. In *Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, volume 1, pages 27–30, October 2010.

[31] K. K. Tan, M. Khalid, and R. Yusof. Intelligent traffic lights control by fuzzy logic. *Malaysian Journal of Computer Science*, 9(2):29–35, December 1996.

[32] D. Srinivasan, M. C. Choy, and R. L. Cheu. Neural networks for real-time traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 7(3):261–272, September 2006.

[33] K. T. K. Teo, W. Y. Kow, and Y. K. Chin. Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. In *Proceedings of the 2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM)*, pages 172–177. IEEE, September 2010.

[34] A. M. Turky, M. S. Ahmad, and M. Z. M. Yusoff. The use of genetic algorithm for traffic light and pedestrian crossing control. *IJCSNS International Journal of Computer Science and Network Security*, 9(2):88–96, Febuary 2009.

[35] L. Singh, S. Tripathi, and H. Arora. Time optimization for traffic signal control using genetic algorithm. *International Journal of Recent Trends in Engineering*, 2(2):4–6, November 2009.

[36] B. Park, I. Yun, and K. Choi. Evaluation of microscopic simulation tools for coordinated signal system deployment. *KSCE Journal of Civil Engineering*, 8(2):239–248, March 2004.

[37] N. T. Ratrout and S. M. Rahman. A comparative analysis of currently used microscopic and macroscopic traffic simulation software. *The Arabian Journal for Science and Engineering*, 34(1B):121–133, April 2009.

[38] N. A. Chaudhary, V. G. Kovvali, and S. M. M. Alam. Guidelines for selecting signal timing software. Technical report, Texas Transportation Institute, Texas A & M University System, USA, September 2002.

[39] North Dakota State University Advanced Traffic Analysis Center(ATAC). Signal Coordination Strategies Final Report. Technical report, Grand Forks - East Grand Forks Metropolitan Planning Organization, USA, June 2003.

[40] R. J. C. Chen, P. Bloomfield, and F. W. Cubbage. Comparing forecasting models in tourism. *Journal of Hospitality & Tourism Research*, November 2007.