© Copyright 2019

Wan Li

# Traffic Signal Timing Optimization with Connected Vehicles

Wan Li

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Xuegang (Jeff) Ban, Chair

Yinhai Wang

Lillian J. Ratliff

Dominik A. Karbowski

Program Authorized to Offer Degree:

Civil and Environmental Engineering

University of Washington

## Abstract

Traffic Signal Timing Optimization with Connected Vehicles

Wan Li

Chair of the Supervisory Committee:
Dr. Xuegang Ban
Civil and Environmental Engineering

The advent and deployment of Connected vehicle (CV) and Vehicle-to-everything (V2X) communications offer the potential to significantly improve the efficiency of traffic signal control systems. The knowledge of vehicle trajectories in the network allows for optimal signal setting and significant improvements in network performance compared to existing traffic signal control systems. This research aims to develop a framework, including modeling techniques, algorithms, and testing strategies, for urban traffic signal optimization with CVs. The objective is to improve the safety, mobility, and sustainability of all vehicles in the study areas utilizing CV data, i.e., real time information on vehicles' locations and speeds, as well as communications to the signal control systems.

The proposed framework is able to optimize traffic signal timing for a single intersection and along a corridor under various market penetration of CVs. Under full penetration rate of CVs, the signal timing optimization and coordination problems are first formulated in a centralized scheme as a mixed-integer nonlinear programing (MINLP). Due to the complexity of the model, the problem is decomposed into two levels: an intersection level to optimize phase durations using dynamic programing (DP) and a corridor level to optimize the offsets of all intersections. Under medium-to-high penetration rates of CVs, Kalman filter methods are applied to estimate trajectories of unequipped vehicles given the available trajectories of CVs. The estimated trajectories combined with CV trajectories are utilized in the trajectory-based signal timing optimization process. Under relatively low penetration rates of CVs, a Deep Intersection Spatial Temporal Network (DISTN) is developed to predict short-term movement-based traffic volumes. The predicted volumes are used in a volume-based adaptive signal control method to calculate signal timing parameters.

Comprehensive testing and validation of the proposed methods are conducted in traffic simulation and with real world CV (probe vehicle) data. The testing tasks aim to validate that the developed methods are computationally manageable and have the potential to be implemented in CV-based traffic signal applications in the real world.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to my Ph.D. advisor, Professor Jeff Ban, who have provided tremendous support and assistance to me during my PhD studies, both in research and life. Without your guidance and persistent help this work would not have been possible.

I would like to thank my committee members, Professor Yinhai Wang, Professor Lillian Ratliff, and Mr. Dominik Karbowski. Thank you all for giving me all the brilliant and perceptive advice and suggestions to my dissertation.

I would like to thank my parents, Jianguo Li and Xiuxia Yuan for raising me and always give me the best. I cannot ask for more. Without your constant support and understanding, I can never go this far to pursue my dreams and make it happen.

A special thanks to my husband, Dr. Boyu Wang. Thank you for always being there for me. You are a brilliant and positive person. You can always cheer me up when I am in a bad mood. You are always respect my choices and accept me for who I am. I feel really lucky to meet you.

# Chapter 1. INTRODUCTION

## 1.1 RESEARCH MOTIVATION

As a critical infrastructure that is crucial to the economy and the daily life of everyone, transportation also creates severe congestion and consumes tremendous energy. In the United States, e.g., gasoline consumption by the transportation sector was about 143.37 billion gallons in 2016, a daily average of about 9.33 million barrels [1]. At the same time, traffic congestion on urban roads has caused extra fuel consumption as well as additional travel delays. The 2015 Urban Mobility Scorecard [2] estimated that U.S. highway congestion costs $160 billion a year, and an average American commuter loses 42 hours per year due to traffic congestion. Therefore, it is imperative to reduce traffic delay and improve transportation energy efficiency in urban areas. Over the past several decades, extensive efforts have been made to improve the efficiency of traffic signal control systems in order to alleviate the ever-growing traffic congestion. An advanced traffic signal control system is characterized by a few features: the ability to accurately collect real-time traffic information, a robust method to estimate and predict future traffic conditions, and an efficient algorithm to optimize traffic signal timing parameters based on current or future traffic conditions. However, traditional traffic signal control systems have various limitations that prevent them from effectively reducing congestion.

### 1.1.1 *Challenges of Traditional Traffic Signal Control Systems*

Previously, most traffic signal researchers assumed that infrastructure sensors (such as loop detectors or video cameras) were the major source of information on real-time traffic conditions. Traffic control systems mainly relied on traffic counts and data from infrastructure sensors. Traffic

signal plans were developed based on arrival vehicles adjusted by the time of day. However, point detectors and video detectors have many disadvantages. Detectors at stop bars have higher failure rates because of the frequent vehicle braking and accelerating behaviors [3]-[4]. Moreover, maintenance of the detectors is time-consuming and costly. Malfunctioned detectors cannot be repaired timely, which is a common issue for the traditional detector-based traffic signal control systems. Video detectors provide richer traffic information, e.g., vehicle queue lengths in front of stop bars. However, the performance of video detectors could be negatively impacted by environmental conditions, such as lighting (most cited conditions to cause video detector failure) and weather [5].

More importantly, traditional detector-based traffic signal control systems rely on partial observable information, which is incomplete and inaccurate. For example, point detectors only record information at the locations where vehicles pass by. There is no continuous trajectories information, such as speeds, positions, accelerations, and vehicle types between these locations. Such information is crucial for developing a sophisticated traffic signal control system and help better evaluate performance indices, e.g., delay, number of stops, and fuel consumption.

Most signal control systems that are currently deployed utilize historical traffic data (fixed-time signal control system) or current traffic volume collected from loop detectors (actuated traffic signal control) to adjust traffic signal timing parameters for the near-future. Intelligent traffic signal control systems should incorporate prediction schemes in their signal timing determination process. In other words, signal timing parameters should be adjusted and optimized based on future traffic conditions. However, the prediction models always suffer from higher computation burden, making it challenging to deploy in the real world. In addition, the performance of adaptive traffic control systems is heavily dependent on the prediction scheme. Under unstable traffic

conditions with fluctuated traffic demands, it is hard to make an accurate and reliable prediction for the future traffic condition. In this case, adaptive signal control systems will have bad performance compared to a fixed time or actuated plan.

The aforementioned limitations largely influence the performance of traditional traffic signal control systems. They have also motivated researchers and practitioners to apply more advanced technologies and data sources that are able to provide more complete and accurate traffic information. Such advanced data sources and sophisticated algorithms are expected to promote the transformation from traditional detector-based traffic signal control systems to more intelligent, individual vehicle-based or trajectory-based traffic signal control systems.

### 1.1.2 *CV–based Traffic Signal Control*

Instead of relying on infrastructure sensors such as loop detectors, urban traffic signal control systems can be improved by the Connected Vehicle (CV) technology. A CV is any vehicle that is able to broadcast information remotely, e.g., vehicle on-board sensor data or traffic condition data to the cloud or infrastructure center. CVs have already brought benefits in a range of different situations, from GPS navigation systems to help plan routes, geolocation systems to help track cars, to less essential but useful functions such as monitoring tire pressure, fuel/electric levels and braking systems [6]. Data generated from CVs are extreme valuable which potentially provides insights into drivers' behaviors and vehicle states. Such data include GPS location and speed, acceleration and deceleration, vehicle type, and origin/destination. There are substantial similarities between CV data and mobile sensing data (e.g., data collected using smart phones; see Herrera et al. [7]; Ban and Gruteser [8]. They can both provide large sets of high-resolution vehicle

GPS data. In this study, CV data and mobile sensing data are treated similarly for the purpose of traffic signal control.

CV enables Vehicle-to-everything (V2X) communications and leads to an intelligent transportation system where all vehicles, road users, and infrastructure systems can communicate with each other to exchange information. The idea behind this technology is that a CV with built-in electronic devices is able to communicate with its surroundings in real-time including vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), vehicle-to-roadway infrastructure (V2I), and vehicle-to-network (V2N) communications. All technologies are paving the way for future smart cities.

V2V communications show great promise in helping reduce crashes, ease traffic congestion, and improve fuel efficiency by exchanging information on the speed and positions of surrounding CVs. Imagine a CV in front of you talking to you about the intent to change lane before it even makes a change or a flat tire alert from a nearby CV. Such information allows CVs to see further and improve the level of predictability.

Significant applications of V2I technologies include smart traffic signals and smart parking. Smart traffic signals play a key role in the development of creating a smart infrastructure. They calculate traffic signal timings using different sources of information, e.g., cameras, sensors, and CV data. Moreover, it can communicate with nearby intersections so that a network of signals can work in tandem and maximize the road efficiency and traffic throughput. The smart parking system is another application of V2I technology. It allows communications from the parking lot to CVs. Vehicles are aware of which spaces are available and choose the one with the appropriate location and cost.

Vehicles today have more ways to be connected than before. Even old vehicles can be connected with the development of telematics. In the early 1990s, General Motors was the first to introduce a telematics system, named OnStar, which enabled vehicle drivers to call for roadside assistance from their vehicles. Since then the vehicle market has been exploded with CV features. Various communication technologies can be applied to CVs, such as cellular, Wi-Fi, satellite radio, or dedicated short-range communication (DSRC) [9]. They form the backbone of the CV system. A summary of the pros and cons of different communication methods is present in Table 1.1. Although cellular network covers the majority of the locations where people live and work, there are areas where cellular services are not available. Long-term evolution (LTE) is a promising technology that can help deliver data more quickly. However, the transmission rate is a major issue when users are moving or in an area with many other LTE users. Privacy and the high cost of cellular data is another concern for cellular communications. Wi-Fi technology offers higher data rates but has similar cost and security concerns to cellular communications. Satellite radios have the disadvantages of slow download time for satellite communications. DSRC is a mature communication technology that ensures reliable and secure communications when vehicles are at high speed. Traffic information can be collected into "Basic Safety Message" (BMS) and other types of messages [10]. Cost and security risks are the main concerns of the DSRC technology. A newer and faster 5G network will allow instantaneous data transmissions, which enables new technologies like CVs. Currently, many leading companies, including major automotive players such as Audi, BMW, and Daimler, the telecom services and equipment providers (Huawei, Ericsson, and Nokia), and semiconductor manufacturers (Intel and Qualcomm), are developing, testing and promoting 5G communications for CVs [11]. At the initial stages, 5G networks will be expensive for carriers and may only cover a small number of users. However, if one wireless

standard applies to all electronics and automotive industries instead of serval, and if the coverage of 5G network gets larger in cities and towns, lower costs are expected for the application of 5G for CVs.

Table 1.1. Pros and Cons of different communication methods

| Communication methods | Pros | Cons |
|---|---|---|
| Cellular | 1. Offer widespread coverage throughout the nation ;<br>2. Long-term evolution (LTE) delivers data quickly.<br>3. 5G network provides more bandwidth for everyone | 1. Dead spots exist (area cellular services are not available);<br>2. Transmission rates slow down when user is moving or in a area with many other LTE users;<br>3. Security risks<br>4. High cost of cellular data<br>5. Small coverage of 5G network. |
| Wi-Fi | 1. Offers Higher data rates | 1. Slow transmission rates if a user is moving<br>2. Security cost<br>3. Price concerns |
| Satellite Radio | 1. Provide broadcast service national wide | 1. Not covering Alaska and Hawaii<br>2. Data download time is slow<br>3. Security risks |
| DSRC | 1. Provides instantaneous network Connectivity and message transmission<br>2. Has a designated licensed bandwidth to permit secure reliable communication<br>3. Provides high data transmission rates | 1. Security risks<br>2. Cost concerns |

The information/data exchange among vehicles and between vehicles and the infrastructure has the potential to improve traffic mobility and safety, warn drivers of upcoming road conditions, and adjust traffic signal timing more efficiently at signalized intersections. There are many successful CV applications all over the world. In 2011, Japan deployed the ITS Spot system to implement V2I on both local roads and expressways by providing three services to drivers: dynamic route guidance, safe driving support, and electronic toll collection [12]. In 2013, Germany, Netherland, and Austria worked on the deployment of a European Cooperative ITS (C-ITS) corridor that incorporated V2I to provide traveler information on roadwork and upcoming traffic [13]. In the United States, USDOT, transportation agencies, academic researchers and various state

holders have been engaged in the development of technologies and systems that enable V2V and V2I applications. From 2012 to 2014, USDOT deployed V2V DSRC devices on real roads with real drivers and evaluated the functional feasibility of V2V in Model Deployment in Ann Arbor, Michigan. There are approximately 2800 equipped vehicles, including cars, trucks, and transit vehicles. Overall, the experiment was successful in creating the interactions between DSRC equipped vehicles that successfully communicated with each other. In the past decade, USDOT provided more than 600 million in funding for CV technologies. Currently, USDOT has provided about $100 million in funding for a number of pilot projects comprised of V2V and V2I technologies and applications [12].

Although the CV technology is promising, it may take a long transition time to cover the entire vehicle fleet. The National Highway Traffic Safety Administration (NHTSA) has proposed a mandate to require all new light vehicles being capable of V2V communications by 2022 so that 60% of vehicles (about 146 millions of them) may be equipped with V2X/DSRC devices by 2029 [9]. Similarly, the American Association of Highway and Transportation Office (AASHTO) predicted that 90% of light vehicles would be equipped with V2V technologies by 2040. AASHTO also estimated that by 2025, 20% of signalized intersections will be capable of V2I communications, and by 2040, 80% of the signalized intersection will be V2I capable. Figure 1.1 shows the DOT's planned CV path to deployment from 2010 to 2040 [12].

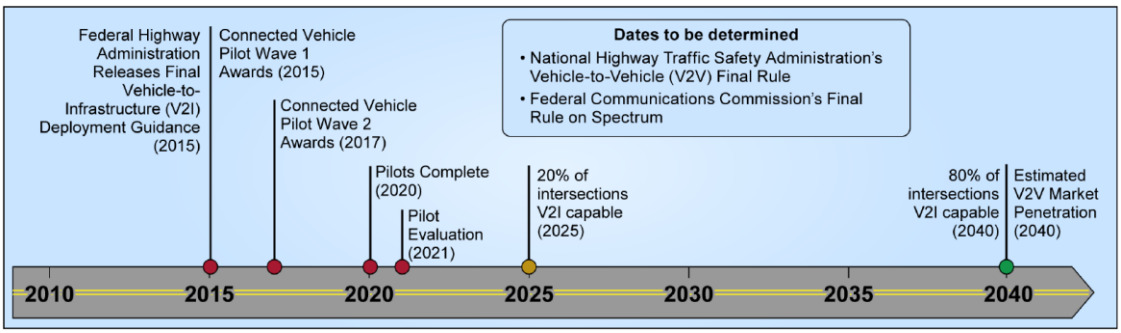



Figure 1.1. DOT's Planned connected vehicle path to deployment, 2010-2014 [12].

The advent of CV technologies offers an opportunity to significantly enhance the transportation system. Traffic signal control systems are the primary tools for urban traffic flow management on arterials with the objective to increase safety, improve traffic flow, and reduce traffic delays and fuel consumption. V2X (especially V2V and V2I) communications bring new paradigms compared to traditional traffic signal operations. Traffic controllers collect data from CVs (position, speed, fuel consumption parameters). They then process the data to optimize signal-timing plans at an intersection, along a corridor, or for a region in order to minimize delays, number of stops, and environmental impacts.

## 1.2 OBJECTIVE

This study aims to develop a framework, including modeling techniques, algorithms, and testing strategies, for urban traffic signal timing optimization with CVs. The framework should be able to optimize traffic signal timing for a single intersection and along a corridor. CV data (or mobile sensing data) are utilized, i.e., the real-time information on vehicles' locations, speeds, accelerations, and vehicle types. With such information, traffic signal timing parameters are adjusted to improve safety, mobility, and sustainability of all vehicles in the network.

In summary, the specific objectives of this project are:

1. Develop and evaluate signal timing optimization methods for isolated intersections under full penetration rates of CVs. By assuming a fixed cycle length (to facilitate signal coordination when optimizing multiple signals along a corridor or on a network), a mixed integer nonlinear programming (MINLP) that considers trajectories of individual vehicles is developed to minimize the total weighted sum of fuel consumptions and travel times of

all vehicles in the study area. The MINLP model is approximated to dynamic programming (DP) formulation to improve computational efficiency.

2. Develop coordinated signal operation schemes along corridors or for a network to optimize the phase durations and offsets of multiple intersections under full penetration rates of CVs. Signal coordination provides a smooth propagation of vehicle platoons on an arterial or network.
3. Estimate and predict traffic volumes of each movement in a signalized intersection given sampled trajectory data from CVs (under low penetration rates of CVs). The proposed model should capture the spatial and temporal dependencies of traffic data. The predicted volume can be used as the input to adjust traffic signal timings.
4. Estimate and predict trajectories for unequipped vehicles given sampled CV trajectories under medium or high penetration rates of CVs. The trajectory-based signal control method can be applied to calculate signal timing parameters.
5. Testing and validating the models and algorithms developed in this study using traffic simulation and real-world data. The methods will also be tested and compared against state-of-the-art methods.

## 1.3 CONTRIBUTIONS

This study improves the traditional detector-based traffic signal control methods to the more advanced trajectory-based signal control scheme by taking advantages of CV technologies and V2I communications. CV makes it possible to collect drivers' information, including their precise trajectory information (e.g., second-by-second vehicle speeds and locations). Such information

can be transferred to signal controllers for signal timing optimization. More specifically, the major contributions of this study include:

1. This study proposes a novel modeling framework for signal timing optimization and coordination problems using individual CV's trajectories. A Mixed Integer Nonlinear Program (MINLP) is developed to optimize phase durations for a single intersection. Due to the large dimension of the problem, directly solving the MINLP is challenging. Dynamic programming (DP) is then applied to optimize the signal timing parameters. In order to ensure a fixed cycle length (for signal coordination purposes), a two-step method is added to the traditional DP: the end stage cost and a branch and bound algorithm. The proposed method can solve the signal timing optimization problem efficiently while ensuring the fixed-cycle length.
2. For a corridor containing multiple intersections, a MINLP is first formulated to optimize phase durations and offsets considering individual CV's trajectories. Then the MINLP is reformulated from a large-size centralized optimization problem to a decentralized two-level signal control problem: the intersection level is to optimize for the timing plan (phase sequence and green times) of individual intersections, while the corridor level is to optimize for the offsets of all the intersections along the corridor. A prediction-based iterative solution method is developed for this two-level problem.
3. This study developed a new method to estimate real-time traffic volumes from vehicle trajectory data. Under low penetration rates of CVs, it is impossible to collect trajectories for all CVs. Under such condition, traffic volumes are considered as the reliable input for signal timing optimization. This study develops a Partial Least Square (PLS)-based traffic

volume estimation algorithm based on sampled CV data. This method is especially useful when missing data existed from loop detectors (a very common issue).

4. Based on the estimated traffic volumes, a Deep Intersection Spatial-Temporal Network (DISTN) is developed to predict real-time traffic volumes for each movement of a signalized intersection by considering the spatiotemporal dependencies of the traffic volume data.
5. Comprehensive testing and validation of the proposed methods are conducted in traffic simulation and using real-world data. Various combinations of travel demand levels and vehicle types of CV are tested for the proposed signal timing optimization methods. The testing tasks help validate that the developed methods are computationally manageable and have the potential to be implemented in CV-based traffic signal applications in the real world.

# Chapter 2. LITERATURE REVIEW

## 2.1 Traditional Traffic Signal Control

Traditional traffic signal control problems have been extensively investigated, with a variety of methods such as fixed-time control, actuated control, and adaptive control [14]. To ensure safety, most existing traffic signal control methods (at least in the US) are based on the dual ring design scheme, as shown in Figure 2.1. The dual ring scheme separates conflicting movements from different approaches, divide a cycle into phases, and determines the timing of each phase at an intersection. There are four movements for straight and/or right turn movements (movement number 2, 4, 6, and 8 in Figure 2) and four movements for left-turn movements (movement number 1, 3, 5, and 7 in Figure 2). The barriers or phase concurrency groups define the conflicts between movements. In this research, the dual ring scheme is applied to ensure traffic safety. This is particularly critical for scenarios when the penetration of CVs is not 100%.

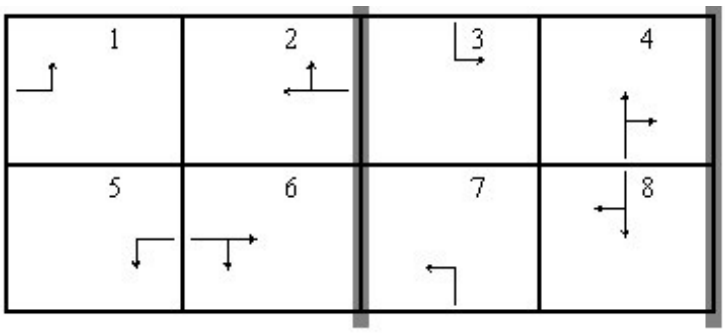



Figure 2.1. Dual ring diagram. Standard NEMA phasing [14]

Currently, there are three types of traffic controllers that are widely deployed all over the world: fixed time controller, actuated traffic controller, and adaptive traffic controller. In the case of fixed-time traffic control, the signal timing variables (e.g., cycle length, split, and offset) are

pre-determined based on historical data. Different programs can be generated to accommodate various traffic demands, such as morning and evening peak hours, according to the time of day. However, such a system is rigid to the real-time fluctuations of vehicle arrivals, leading to inefficient and unsatisfactory performances in many situations.

Actuated traffic control utilizes real-time traffic states provided by loop detectors or other infrastructure sensors at the upstream of an intersection to the stop line to detect vehicle arrivals. It usually maintains a green signal on the busiest street (i.e., the main street) until a pedestrian or a vehicle on the less traveled side street approaches the intersection. The green time will be extended if there is an incoming vehicle being detected. This system performs better than fixed time traffic control due to the following features. A phase can be skipped if there is no waiting vehicle. It can also be actuated earlier if there is no demand from the conflicting movements. The green time can be extended to the maximum green split and can also be terminated if there is a prioritized vehicle (such as fire trucks). In most cases, a signal plan in actuated traffic control is cycle-based with a pre-determined fixed signal sequence. Actuated traffic signal control systems are more appropriate for major-minor streets because it can terminate a phase immediately when there is no vehicle passing by. However, for a busy intersection that high traffic volumes are present for all directions, actuated traffic signal can be considered as fixed time because each phase may have its pre-defined maximum green time. Under such condition, actuated traffic signal control is inefficient.

Adaptive signal control is the most advanced traffic signal control method so far. It still uses detection data as actuated traffic control; however, it retrieves traffic information at the current time and applies prediction models to forecast vehicle states, vehicle arrivals, and queue length in the near future. It continuously adjusts when green lights start and terminate based on current and

predicted traffic conditions, demands, and system capacity to accommodate traffic patterns to promote smooth flow and reduce traffic congestion. The most widely known systems include SCOOT [15] and SCATS [16]. These traffic signal controllers calculate signal plans based on traffic flow information over a look-ahead horizon. The look-ahead search algorithm with short-term prediction periods (e.g., less than one cycle) often leads to side effects, such as undeserving left-turns. In addition, this type of algorithms reevaluates decisions too infrequently (e.g. 4-5s) and are unable to terminate phases immediately when queues disperse earlier than prediction [17]. The look-ahead search algorithms are the foundation for some adaptive traffic control methods, such as OPAC and COP. OPAC is a method for demand-responsive decentralized traffic signal control that requires on-line data from upstream approach detectors and from adjacent intersections. There is no coordination feature imbedded but this method has the self-coordination capabilities [18]. OPAC III implements a "rolling horizon" strategy to minimize stops and delays. They can respond to the variation of traffic flow since they emphasize on traffic prediction. OPAC IV is a network version of OPAC. With the dramatic improvement of the computational capacity and advanced information collection and communication techniques, it is possible to utilize a large volume of real-time data with OPAC IV. Recent adaptive control related research includes the swarm algorithm [19], platoon-based algorithms [20], rolling horizon approaches [21], oversaturation algorithm [22] and reinforcement-learning algorithm [23], among others. A comprehensive discussion of the signal control algorithms could be found in Goodall [24].

Traffic signal coordination can provide efficient movements of vehicle platoons through adjacent intersections and reduce travel times, delays and the number of stops. It is widely implemented for arterials, downtown areas, and closely spaced intersections. There are three parameters used for the coordination concept: cycle length, splits, and offset. Cycle length is the

total time to finish a complete sequence of all signal phases. For coordinated traffic signals, they normally need to have the same cycle length, called the common cycle length. In practice, such a common cycle length may be determined by signal design tools for coordination systems, such as SYNCHRO and TRANSYT. The split is the sum of the green, yellow and all red intervals, which is the segment of the cycle length that allocated to each phase. The offset is the time difference between a fixed point in the cycle and a system reference point.

Coordination is not beneficial for all systems. It requires that intersections are close to each other and the traffic demands between the adjacent intersections are large. A Federal Highway Administration (FHWA) report suggested that if the intersections are spaced within a certain distance (i.e., 0.75 miles), coordination can be considered [25]. If the arrival vehicles are random and unrelated to the operation of the upstream intersections, coordination may provide limited benefits.

Different signal controllers have different mechanisms to realize signal coordination. In the case of fixed time control, (i.e., downtown closely spaced intersections), traffic signal coordination is achieved by setting an appropriate offset value. It requires the same cycle length for all coordinated intersections. For actuated traffic signal control, the cycle length also needs to be the same for the coordinated actuated systems. Wardberg et al. [26] suggested that coordination cannot be realized without the common cycle time of the whole systems. Coordination for actuated traffic signal can synchronize multiple intersections using force-off mode. Force-off is a point in a cycle where a phase must end. It ensures the coordinated phases provided with a minimum amount of green time to implement the green wave. The uncoordinated phases either use the unused time of previous phases in fixed force-off mode or limit to their defined split amount of time in floating force-off mode [27]. Coordination for adaptive traffic signal can be achieved based on the common

cycle, such as the SCOOT system and the RHODES system. Some multi-agent systems (decentralized systems that focus on individual intersections) do not use a common cycle length. However, Lammer and Helbing [28] suggested that it is impossible to coordinate multiple intersections in such a system. Therefore, it is still an open question regarding whether signal coordination can be achieved without the requirement of a common cycle length.

Signal coordination models have some common Measures of Effectiveness (MOEs). Bandwidth maximization used to be a common objective function for signal coordination. It is the amount of time that a vehicle can travel through all intersections of a coordinated corridor without stopping. Bandwidth is related to the system capacity and throughput and is determined by the offsets. The literature on bandwidth optimization mostly relied on the graphical method in the early stage [28]-[32], which later focused on mixed integer linear programs (MILP) to maximize the sum of the bandwidths for the two directions of the coordinated corridor. Branch and bound algorithms were often used to solve the optimization problem. For example, Gartner et al. [33] expanded the previous signal coordination models by considering actual traffic volumes and flow capacity in the MILP formulation for bandwidth optimization. Their model is called MULTIBAND because they defined a different bandwidth for each direction of the corridor, which was individually weighted based on their contributions to the objective value. PASSER is a software tool developed to maximize the bandwidth efficiency given the pre-calculated splits [34]. Other MOEs include delays, total travel times, and the number of stops when conducting offset optimization. Coogan et al. [35] optimized the offset of the coordinated traffic signals to reduce the average queue lengths at all intersections by assuming a fixed timing plan with a common cycle length. They derived a closed-form analytical expression, which is a non-convex, quadratically constrained quadratic program (QCQP). The simulation results demonstrated a

significant reduction in queue lengths. Hu and Liu [36] developed a data-driven arterial offset optimization model to minimize the total delay for the main coordinated direction, which considered the stochastic nature of real-world traffic. They solved two problems in the proposed model: the early return to green problem for the coordinated phase and the uncertainty of the intersection queue size.

## 2.2 Traffic Signal Control with CVs

There have been various traffic signal control studies under the CV environment. Dual ring controllers have been applied to many of those studies. He et al. [37] developed the platoon-based arterial multi-modal signal control with online data (PAMSCOD) algorithm. Signal timing is updated every 30 seconds. A MINLP was solved to determine a future optimal signal plan. Simulation results in VISSIM showed that delays were significantly reduced under both non-saturated and oversaturated traffic conditions compared to traditional state-of-the-practice coordinated actuated signal control. Lee and Park [38] developed a cumulative travel-time responsive (CTR) real-time intersection control algorithm in the CV environment. They examined the different penetration rates of CV and levels of congestion. The Kalman filtering technique was utilized to estimate the cumulative of travel time under various penetration rates. They suggested that 30% of market rates of CV were needed to realize the benefits of the CTR algorithm. Feng et al. [39] presented a real-time adaptive signal control algorithm using CV data. The algorithm incorporated a two-level optimization model with two objective functions: minimizing the vehicle delay and minimizing the queue length. Dynamic programming was applied to solve the discretized signal control problem. The cycle length is assumed to be variable. Beak et al. [40] extended the work of Feng et al. [39] in two ways. First, they imposed extra constraints to the upper level to ensure a fixed cycle length. Second, the revised intersection-level model (with a

fixed cycle length) is integrated into a corridor-level model for signal coordination. They developed a two-level optimization method for adaptive coordination under the CV environment. At the intersection level, the optimal green time for each phase is determined from dynamic programming. At corridor level, the offset is optimized to obtain minimum delay. They used a platoon model to estimate the platoon length and flow rate. A platoon dispersion model was applied to identify how platoon disperses on the corridor over time. Simulation results showed that the model can reduce average delay and the number of stops for both coordinated phases and the entire network. Li and Ban [40] formulated the traffic signal optimization problem for a single intersection as a MINLP that was reformulated as a DP problem. They also developed a two-step method to make sure that the obtained optimal solution can lead to the fixed cycle length, which is often required for coordinating multiple signals on a traffic corridor or a network. Zhao et al. [42] proposed a signal timing optimization strategy to minimize the combined total energy consumption and traffic delay, considering the fuel consumption of individual vehicles. Vehicles' trajectories were predicted second by second using the Nagel-Schreckenber model. An iterative grid search algorithm was used to search for the optimized signal timing. There is also a large body of literature on traffic control with connected and automated vehicles (CAVs) for both intersection control and vehicle control; see [43]-[47].

There are also many studies that did not apply the dual-ring controllers in their CV-based signal optimization methods. These types of traffic signal designs are more flexible without considering the cycle length, the number of phases, phase transitions, or phase sequences. Although some of them still follow the phased-based signal design, they do not consider the cycle length or the offset in their designs. Priemer and Friedrich [48] proposed a decentralized adaptive traffic signal control using V2I communication data. Dynamic programming and complete

enumeration were used to optimize the signal timing in order to reduce the total queue length within a forecast horizon of 20 seconds. Various penetration rates were tested in the simulation. Cai et al. [49] presented a traffic signal control algorithm using information collected from V2I. They constructed a state-space presentation of the control problem using the speed and position as state variables and applied dynamic programming and its derivative methods to optimize signal timing. Datesh et al. [20] applied the *k*-means clustering approach to improving traffic signal efficacy. The algorithm is a platoon-based signal control method that categorizes the approaching vehicles into two groups, red or green. They demonstrated that the algorithm works properly under low penetration rate of CV. Goodall et al. [24] developed the predictive microscopic simulation algorithm (PMSA) to control traffic signal. The strategy can minimize total delays, or the combination of delays, stops, and decelerations over a 15-second time period by considering instantaneous vehicle data. The study showed that at low or mid-level traffic volume, their proposed algorithm outperformed state-of-the-practice coordinated-actuated timing plan, while the performance got worse during saturated and oversaturated conditions. The method however ignored left-turn traffic and cannot be applied to real-world intersections. Li and Qiu [50] proposed an adaptive signal control approach based on CVs to improve intersection throughput. The approach incorporated a two-step centralized responsive control for vehicles in motion and stopped vehicles. The simulation results suggested that limited benefits are achieved when the traffic demand is high. Islam and Hajbabaie [51] developed a distributed coordinated methodology for signal timing optimization on CV networks. They reduced the complexity of a network level decision problem to a single intersection level problem by deciding the termination or continuation of green times. They evaluated the influence of demand levels and penetration rates of CV on their signal optimization algorithm in several case studies.

Among the various types of methods, DP is one of the most commonly used technique to solve the discretized signal control problems. It was first applied in Sen and Head [51] to optimize traffic signal timing. The idea was later applied in Chen et al. [53] and Feng et al.[39]. In particular, Feng et al. [39] proposed a bi-level formulation for optimizing signal timing of a single intersection: the upper level is to optimize for the barrier lengths and the lower level is to optimize for the phase times. However, all these studies assumed varying cycle lengths (and thus could not apply directly to deal with the fixed cycle length constraint). Signal timing plans with variable cycle lengths may not be readily applied to multiple intersections if signal coordination is needed. Beak et al. [40] extended Feng et al. [39] to impose the fixed cycle length, albeit with a bi-level formulation. First, they imposed extra constraints to the upper level to ensure a fixed cycle length. The revised intersection-level model (with a fixed cycle length) is then integrated into a corridor-level model for signal coordination. In this dissertation research, a two-step method is developed to resolve the fixed cycle length issue at the intersection level. This avoids the use of the bi-level structure in Feng et al. [39] and Beak et al. [40], which is computationally demanding.

For signal coordination under the CV environment, most of the existing traffic signal optimization/coordination methods applied a centralized scheme that various signal timing parameters (phase durations, cycle length, and offsets) are optimized together in one mathematical problem. This can lead to several problems. First, individual vehicle-based signal control problems are often a NP-hard problem [51]. Second, for a large traffic corridor or road network, the signal timing optimization and coordination problem are hard to solve and not applicable for real-time signal control. Third, some studies tried to decentralize the signal optimization problems by decomposing the entire problem into a few manageable sub-problems. However, they mostly

assumed varying cycle lengths [38][48][51] and thus could not apply directly to traffic signal coordination.

## 2.3 Spatiotemporal Traffic State Estimation and Prediction

The CV provides real-time information on traffic conditions, which is promising and beneficial for the design and evaluation of traffic signal plan and other management strategies. However, the penetration rate of CVs can largely influence the ability to use the data (i.e., position, speed, acceleration, directions of movement, etc.). It is one of the major challenges to implement the CV-based traffic signal control system.

In the literature, it was suggested that 20%-30% penetration of CVs is needed for signal control applications [24]. Wunderlich et al. [54] suggest that even with low penetration rate (1-10%), it could still obtain an acceptable error rate of travel time estimation (less than 15%) under low and moderate congestion levels. In a series of mobile sensing based urban traffic modeling applications [55][56], it was estimated that 5% is probably the minimum penetration rate of mobile sensing data, which is very similar to CV data if only vehicle trajectories are concerned.

With the understanding that in the near future one will have to deal with mixed traffic flow (with CVs and non-CVs or *unequipped* vehicles), it is crucial to develop methods to use the available CV trajectories to estimate and predict traffic states, especially traffic volumes. Over the past several decades, many traffic state estimation and prediction models have been developed to improve traffic management and control. Autoregressive integrated moving average (ARIMA) model was first applied to predict freeway traffic flow in the 1970s [57]. Later on, time-series methods have been widely used for traffic flow prediction [58]-[60]. Although ARIMA methods have been proven to be quite effective, they do have their own limitations. For example, ARIMA

simply approximates historical patterns of time series data, which cannot explain the structures of the underlying processes that generate the data. Unlike ARIMA that requires data to be stationary, Kalman filter methods can be applied to both stationary and nonstationary data [61]-[62]. Such methods are suitable for traffic flow prediction because they can update state variables using real-time data, which can better account for the fluctuations of traffic flow [63]. Partial Least Square (PLS) Regression was applied for travel time prediction, which is similar to traffic volume prediction when data present spatiotemporal correlations [64]. PLS accounts for the spatial and temporal correlations of traffic volumes at the same time and can thus predict traffic volumes for all movements of the target intersections simultaneously. In particular, PLS can identify the correlated, low-rank representations of historical measurement data and predict future traffic volumes by determining which latent structures are dominant in traffic volume measurements. Due to the various nonlinear spatial relationships and complicated underlying temporal dependencies of traffic volume data, more studies have investigated nonparametric methods for traffic volume prediction, including the *k*-nearest neighbor (*k*-NN) [65]-[66], support vector machines (SVM) [67], and Bayesian network approaches [68]. In general, these methods were developed based on limited, site-specific data and their performances were largely determined by the embedded spatial and temporal features of the collected data.

Recently, deep learning methods have been rapidly developed and shown superior and promising performances in many areas, such as computer vision and natural language processing due to their ability to model complex nonlinear relationships and to deal with a large amount of data. First theorized in the 1980s, those methods have recently become useful due to the substantial computing power of high-performance graphics processing units (GPUs) and increasingly available data. The success of deep learning has encouraged many researchers to apply the

technique to traffic volume prediction. Lv et al. [69] developed a deep-learning-based traffic flow prediction method that considered the spatial and temporal correlations of traffic volumes. They applied a stacked autoencoder model to learn traffic volume features and trained it in a greedy layer-wise fashion. Yao et al. [70] developed a Spatial-Temporal Dynamic Network (STDN) to learn the similarity between nearby locations/regions in a network via traffic flow. They also used a periodically shifted attention mechanism to handle the long-term periodic dependency. Wu et al. [71] proposed a deep neural network-based traffic flow prediction model (DNN- BTF) to improve the prediction performance. In their model, the convolutional neural network (CNN) was used to study the spatial features and the recurrent neural network (RNN) was applied to mine the temporal features. Cui et al. [72] developed a deep stacked bidirectional and unidirectional long short-term memory (SBU-LSTM) neural network to consider both the forward and backward dependencies of the time series of traffic flow data on freeways. Polson and Sokolv [73] combined a linear model that is fitted using L1 regularization and a sequence of hyper tangent layers into their deep learning model to predict traffic flow. The first layer aimed at identifying spatiotemporal relationships among predictors and other layers were used to model nonlinear relationships. Jia et al. [74] proposed a deep belief network (DBN) LSTM to predict urban traffic flow considering the impact of rainfalls. The experimental results showed that by considering additional rainfall features, traffic flow predictions have better accuracy than the model without the rainfall input. Furthermore, LSTM outperformed DBN to capture the temporal features of traffic flow data.

Table 2.1 summarizes both the pros and cons of several state-of-the-art volume prediction methods and the proposed DISTN model. In the numerical experiment, the proposed model is evaluated and compared with all the methods in Table 2.1.

Table 2.1. Comparisons of the state-of-the-art volume prediction/estimation methods

| METHODS | Pros | Cons |
|---|---|---|
| **ARIMA** (Williams et al. [59]; Lee & Fambro, [60]) | 1. More interpretable, less flexible;<br>2. Work best on long and stable series | 1. Cannot capture the spatial features<br>2. Not appropriate for nonstationary data<br>3. Cannot model the nonlinear spatial and temporal relationships |
| **Kalman filter** (Ojeda et al. [61]; Guo et al. [62]) | 1. Robust to noise data;<br>2. Work for both stationary and nonstationary data<br>3. Computationally efficient, suitable for real-time traffic flow prediction | 1. Cannot capture the spatial features<br>2. Cannot model the nonlinear spatial and temporal relationships. |
| **DMD** | 1. Robust to redundant data; dimensionality reduction algorithm.<br>2. Easy algorithm, nonparametric technique | 1. Cannot capture the spatial features<br>2. Inability to capture the transient and intermittent phenomenon. |
| **PLS** (Coogan et al. [35]) | 1. Robust to noise and correlated data.<br>2. Makes few assumptions on how the data are distributed | 1. Difficult to interpret loadings of independent latent variables<br>2. Limited capabilities to apply on large scale data |
| **LSTM** (Cui et al. [72]) | 1. Account for long term dependencies<br>2. Account for the spatial and temporal correlations | 1. Results (parameters) are always hard to interpret and visualize<br>2. Cannot capture the unique spatial features of the lane-based traffic volume at signalized intersection |
| **CNN** (Ma et al. [93]) | 1. Account for both spatial and temporal correlations by using a time-space matrix. | 1. The unique spatial features of movement-based volume are not captured |
| **DISTN** | 1. Account for long term dependencies<br>2. Account for the spatial and temporal correlations<br>3. Capture the unique spatial features of lane-based traffic volume data at signalized intersections | 1. Results are always hard to interpret and visualize.<br>2. More periodical and seasonal trends need to be further investigated.<br>3. More periodical and seasonal trends need to be further investigated. |

## 2.4 SUMMARY

Most of the existing traffic signal optimization/coordination methods applied a centralized scheme that various signal timing parameters (phase durations, cycle length, and offsets) are optimized together in one mathematical problem. This can lead to several problems. First, individual vehicle-based traffic signal control/coordination problems often have large dimensions and are usually NP-Complete (a class of computational problems for which no efficient solution algorithm can be found) [51]. Second, for a large traffic corridor or road network, the signal timing optimization and coordination problem are hard to solve and not applicable for real-time signal control. Some studies did try to decentralize the signal optimization problems by decomposing the entire problem into a few manageable sub-problems. However, they mostly assumed varying cycle lengths and thus could not apply directly to traffic signal coordination.

Under low penetration rates of CVs, the aggregate level of performance need to be estimated and predicted. Existing methods applying deep learning techniques for traffic volume prediction have two main limitations. First, so far, the focus has been mainly on freeway traffic volume prediction and volume prediction on urban streets is sparse. Compared with the bidirectional continuous traffic flow at freeways, movement-based traffic volumes at signalized intersections have larger fluctuations and more complex underlying relationships in both spatial and temporal domains. For example, the traffic volume of a certain movement is not only correlated with other movements of the same intersection but also related to the arrival vehicles from upstream intersections and departure vehicles to the downstream intersection. In addition, movement-based traffic volumes are largely influenced by traffic signal operations. The common stop-and-go conditions increase the difficulties of volume prediction. The second limitation is that although

many methods accounted for the spatial dependencies of traffic flow on freeways to some extent, the unique spatial features of movement-based traffic volumes at urban intersections have not been fully investigated. For example, how traffic volume of a certain movement correlates with other movements of the same intersection and relates to the volumes at the upstream and downstream intersections has not been fully understood and captured in deep learning methods for volume prediction.

# Chapter 3. OVERALL MODELING FRAMEWORK

This chapter provides an overview of the methods developed in this research, including signal optimization methods for both a single intersection and multiple intersections along a traffic corridor. More detailed discussions of the methods are presented in subsequent chapters. The overall research framework is shown in Figure 3.1. It is noted that pedestrians are not considered in this study. Different methodologies are developed with respect to full, medium-to-high, or low penetration rates of CV. Here medium-to-high penetration represents that at least one vehicle per platoon for a given cycle. The minimum value of medium-to-high penetration rates of CVs is one vehicle per cycle. Table 3.1. shows some examples of the possible penetration values considering traffic volume and cycle length. Under heavily traveled intersections (high volumes and longer cycle lengths), the required minimum penetration is not very high (from a few percentages to 20%). If there are two platoons per cycle, the values need to be doubled. For less traveled intersections, (low volume and shorter cycle lengths), the minimum penetration requirements for medium penetration can be quite high.

Table 3.1. The minimum value of medium-to-high penetration rates of CVs

| **Traffic volume (vphpl)** | **Cycle Length (s)** | | | |
|---|---|---|---|---|
| | **30** | **60** | **90** | **120** |
| **250** | 48.00 | 24.00 | 16.00 | 12.00 |
| **500** | 24.00 | 12.00 | 8.00 | 6.00 |
| **750** | 16.00 | 8.00 | 5.33 | 4.00 |
| **1000** | 12.00 | 6.00 | 4.00 | 3.00 |

Under full penetration rates of CVs, Intelligent Driver Model (IDM) is applied to predict vehicle trajectories in the short term, based on which to optimize signal timing parameters at the isolated intersection and along a corridor. Medium to high penetration represents (roughly) that at least one sampled vehicle exists per cycle. Under this condition, individual vehicle trajectories of

the unequipped vehicle can be estimated and predicted using Kalman filter methods. Under low penetration, the aggregate performance measures, e.g., volumes and queue lengths can be estimated and predicted for signal optimization. If CV trajectory data and detector/camera data are both available for certain sites/intersections, data fusion-based PLS model can be trained given those data. For other similar intersections with only CV trajectory data, traffic volumes of those intersections can be obtained or estimated using the trained PLS model. If only detector/camera data available, deep learning-based methods can be developed to predict future traffic volumes. This can be done on a cycle-by-cycle basis or for a short time period, i.e., 15 min or 30 min, whichever has the best prediction performance. The predicted volumes can be used for fixed signal control to optimize green splits.

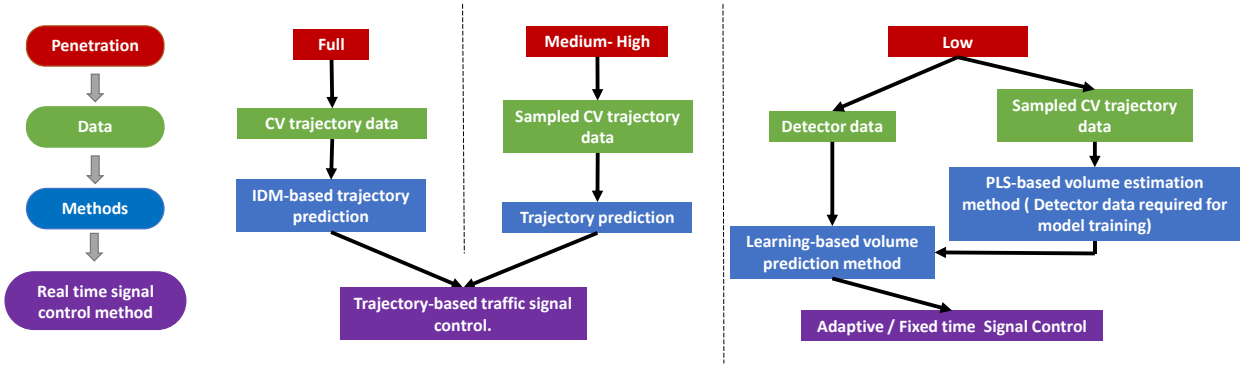



Figure 3.1. Traffic signal timing optimization

Practical traffic signal control systems have different priorities. It is commonly agreed that the priorities of traffic signal control systems are (from high priorities to low priorities): safety, efficiency, and other objectives (such as fuel consumption and emissions). Safety can be ensured by well-established traffic control design methods that can separate conflicting traffic flows in time in order to reduce collisions. For example, a dual-ring controller can be applied to allocate the phase duration of each phase group, as shown in Figure 3.2. Due to its actuation features, and safety concerns (minimal conflicts between different movements), dual ring phase controllers have

become the dominant traffic signal type especially in the US [75]. Dual ring controllers can also properly balance the safety and efficiency of traffic signal control [14][39]. This is important since the primary objective of traffic signal control is to ensure safety, i.e., to minimize movement conflicts [14], while mobility is also important as long as safety is ensured. Moreover, it can be easily set up and applied for signal coordination by setting a fixed cycle length constraint.

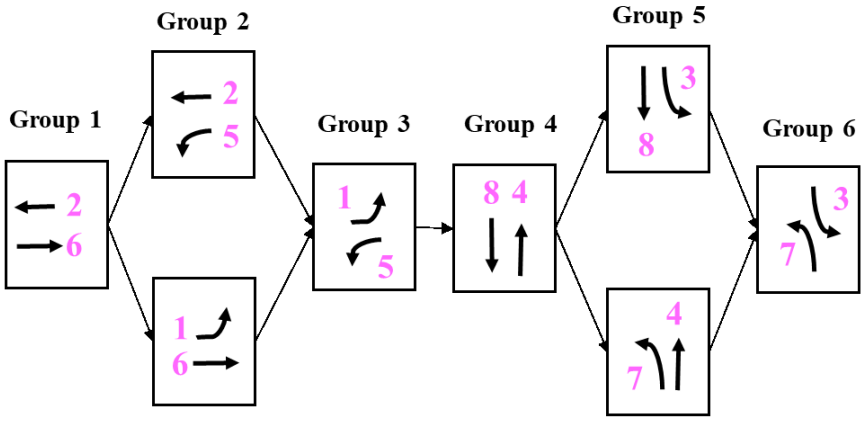



Figure 3.2. Traffic signal configuration [76]

Second, mobility and other objectives, such as fuel consumptions and emissions can be optimized by considering trajectories of all vehicles. For multiple intersections along a corridor or in a network, this can be done in a two-level traffic signal optimization method: the intersection level to optimize the green time and corridor level to optimize the offset. As shown in Figure 3.3, we can treat the bottom intersection as the reference signal and coordinate the other intersections based on the signal operations of the reference signal. Usually, the offset value is maintained for a period of time (e.g., 10 minutes) and may be changed based on the real traffic conditions.

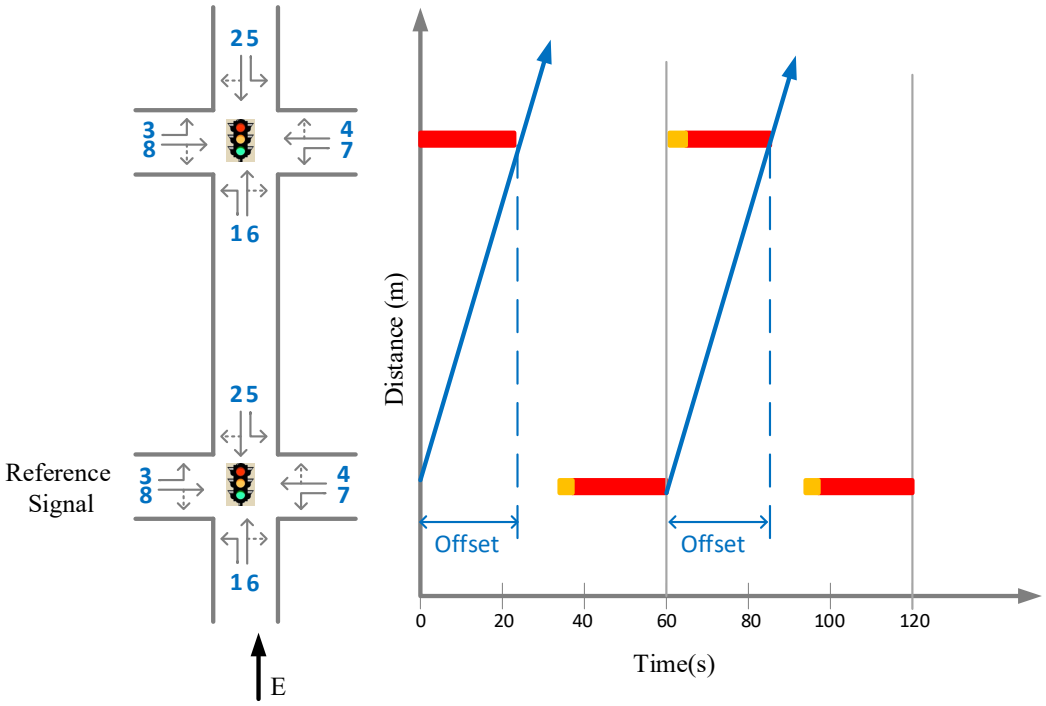



Figure 3.3. Coordination of multiple intersections

# Chapter 4. TRAFFIC SIGNAL TIMING OPTIMIZATION UNDER FULL PENETRATION RATE OF CVS

## 4.1 INTERSECTION LEVEL OPTIMIZATION[1]

This section presents signal optimization modeling framework and numerical results for single intersection optimization. The signal control problem for a single intersection with a fixed cycle length constraint is formulated as a MINLP. Here we adopt the dual-ring method for signal design as shown in Figure 3.2. Without loss of generality, we assume the eastbound/ westbound (EB/WB) through movements (2 and 6 in Figure 3.2) are the major movements and thus cannot be skipped (i.e., for coordination purposes). Other phases may be skipped by setting the corresponding phase durations as zero. We also assume a cycle always starts with movements 2 and 6. Such a signal timing plan can be considered as 6 groups with a sequence of 8 phases in Figure 4.1. Noted that phase 2 and 3 in group 2 cannot be realized at the same time, indicating that at least one of the two phases need to be skipped. The same situation happens for phase 6 and 7 in group 5. In this paper, the continuous time is discretized into 1s intervals.

Based on the literature [77], the maximum/minimum green parameters are defined for a movement. In this study, we define maximum/minimum green based on phases, which is consistent to [77] and Sen and Head [51] when phase overlaps are not allowed, i.e., only phase 1, 4, 5, and 8 exist, as shown in Figure 4.1. A phase overlap refers to a pair of phases that contain one common movement, e.g., phase 1 contains movement 2 and 6, and phase 2 contains movement 2 and 5, occurring in sequence would allow movement 2 to "overlap". If phase overlaps exist, minimum greens for the overlap phases (phase 2, 3, 6, and 7) are set to be zero in order to allow

[1]This subchapter is an amended version of Li and Ban [41]. © [2018] IEEE. Reprinted, with permission, from [41].

them to be skipped, the minimum greens for phases 1,4,5,8 remain the same. Maximum greens for the overlap phases (2, 3, 6, and 7) are set to be a small value, e.g., 10s, while maximum green for the non-overlap phases needs to be deducted by a corresponding value, e.g., the maximum green for phase 1 needs to subtract 10s. In the numerical test, we set the maximum green for non-overlap phases (1, 4, 5, and 8) as the cycle length, minimum green for non-overlap phases as 5s. For overlap phases (2, 3, 6, 7), the maximum green is half of the cycle length and minimum green is zero. This is designed to allow DP to search for sufficiently large state spaces to find an optimal solution, with a reasonable computational effort. The numerical tests suggest that selecting different values as maximum/minimum greens does not have a significant influence on the total cost.

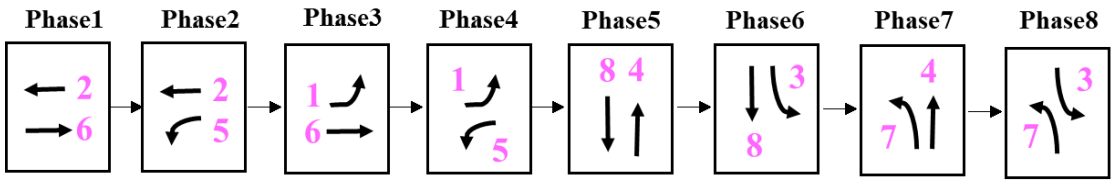



Figure 4.1. Traffic signal configuration

4.1.1 *Mixed-Integer Nonlinear Program*

**Parameters**

| | |
|---|---|
| C | Cycle length (s). |
| $m_F$ | Monetary value of fuel ($/gal), e.g., $3/gal. |
| $m_{TT}$ | Monetary value of travel time ($/s). e.g., $12/h ($0.005/s). |
| $e$ | Idle fuel consumption rate (gal/h). |
| $l_n$ | Length of vehicle $n$ (m). |
| $\delta$ | Acceleration exponent in IDM. It usually set at 4. |
| $H$ | Desired time headway (s), e.g., 1.5s. |

$a$ Maximum acceleration rate ($\mathrm{m/s^2}$), e.g., 1 $\mathrm{m/s^2}$.

$b$ Maximum deceleration rate ($\mathrm{m/s^2}$), e.g., 3 $\mathrm{m/s^2}$.

$s_{on}$ The gap between vehicles in complete standstill traffic jams (m), e.g., 2m.

$v_0$ Vehicle desired speed (m/s).

$g_k^{min}$ Minimum effective green time of phase *k* (s).

$g_k^{max}$ Maximum effective green time of phase *k* (s).

$k$ Signal phase, $k = 1, 2 \ldots 8$.

$\bar{d}_n^0, \bar{d}_n^1$ Entrance location and exit location of the incoming approach of vehicle *n* (m).

$d_{signal,n}$ The location of the nearest front signal of vehicle $n$ (m).

**Variables**

$FC_{n,t}$ Fuel consumption for vehicle $n$ at time *t* (gal/s).

$TT_{n,t}$ Travel time of vehicle n at time t (s).

$FC_{I,n}$ Fuel consumption for vehicle $n$ at idle status (gal/s).

$FC_{s,n,t}$ Fuel consumption of vehicle *n* at time *t* at the moving status (gal/meter).

$g_k^i$ Effective green time allocated to phase *k* of cycle *i* (s).

$v_{n,t}$ Speed of vehicle $n$ at time $t$ (m/s).

$d_{n,t}$ Location of vehicle $n$ at time $t$ (m).

$I_{n,t}$ Idle status indicator for vehicle *n* at time *t*.

$S_{k,t}$ Traffic signal status of phase group *k* at time *t*.

$\bar{k}$ Current phase index at time t. It represents the phase that is currently given the green light.

$Z_{n,t}$ Traffic signal status for vehicle *n* at time *t*.

$y_{n,t}$ Traffic signal indicator. It takes 1 if the preceding vehicle is a traffic signal.

$s_{n,t}$ Vehicle gap (m).

$\Delta v_{n,t}$ The speed difference between vehicle $n$ and $n-1$ at time $t$ (m/s).

$a_{n,t}$ Acceleration rate for vehicle *n* at time *t* ($\mathrm{m/s^2}$).

$y_{t,1}, y_{t,2}$
$y_{t,3}, y_{t,4}$ Binary variables (auxiliary).

The objective of the CV-based signal optimization problem can be formulated as minimizing the weighted sum of total fuel consumption and travel time of all vehicles approaching the intersection:

$$min\, F = \sum_{t=1}^{T} \sum_{n=1}^{N} \left( m_F FC_{n,t} + m_T T_{n,t} \right) \tag{4.1}$$

$FC_{n,t}$ and $T_{n,t}$ are the fuel consumption and travel time for vehicle $n$ at time *t*. The corresponding parameters $m_F$ and $m_T$ are the "value of fuel" and "value of time" respectively. Eq. (4.1) indicates that the objective function here considers the travel time and energy consumption of individual vehicles. Eq. (4.2) calculates the fuel consumption of vehicle *n* at time *t*, which is determined by the vehicle status. If vehicle *n* is idling at time *t*, the indicator variable for idle status, $I_n(t)$ takes one and the fuel consumption model $FC_{I,n}$ is applied, as shown in Eq. (4.4). Otherwise, $FC_{s,n,t}$ will be used, as shown in Eq.(4.5), which calculates the fuel consumption of vehicle n at the moving status ($I_n(t) = 0$). Eq. (4.3b) reformulate (4.3) using the "big *M*" method by establishing a relationship between speed $v_{n,t}$ and idle status indicator $I_{n,t}$. M here is a very large number. The model could be used to calculate fuel consumption for different vehicle types, including sedan,

SUV, bus, electric vehicle (EV), and hybrid electric vehicle (HEV). Zhao et al.[42] provided the calibrated parameters in Eq. (4.4) and (4.5) for different vehicle types, as shown in Table 4.1.

$$FC_{n,t} = FC_{s,n,t} * v_{n,t} * \left(1 - I_{n,t}\right) + FC_{I,n} * I_{n,t} \tag{4.2}$$

$$I_{n,t} = \begin{cases} 1, & if\ v_{n,t} < 5, \quad idle\ at\ time\ t, \\ 0, & if\ v_{n,t} \geq 5, moving\ at\ time\ t \end{cases} \tag{4.3}$$

$$\begin{cases} v_{n,t} - 5 < (1 - I_{n,t})M \\ v_{n,t} - 5 + I_{n,t}M \geq 0 \end{cases} \tag{4.3b}$$

$$FC_{I,n} = e \tag{4.4}$$

$$FC_{s,n,t} = \frac{a}{v_{n,t}} + b + cv_{n,t} + dv_{n,t}^2 \tag{4.5}$$

Table 4.1. Parameter identification for fuel consumption models

| **Vehicle Type** | **Parameters** | | | | |
|---|---|---|---|---|---|
| | ***a*** | ***b*** | ***c*** | ***d*** | ***e*** |
| 1 EV | 4.74e-2 | 2.66e-3 | 6.37e-5 | 1.49e-6 | 0 |
| 2 HEV (SOC0=0.7) | 1.83e-1 | 3.67e-3 | 1.27e-4 | 2.39e-6 | 0 |
| 3 HEV (SOC0=0.6) | 1.83e-1 | 3.67e-3 | 1.27e-4 | 2.39e-6 | 0 |
| 4 HEV (SOC0=0.5) | 1.82e-1 | 1.51e-3 | 5.67e-4 | -4.35e-6 | 0 |
| 5 Sedan | 4.75e-1 | -8.50e-3 | 5.41e-4 | 1.04e-7 | 0.211 |
| 6 SUV | 7.44e-1 | -1.23e-2 | 6.78e-4 | 5.29e-6 | 0.491 |
| 7 Bus | 2.51e+0 | 3.03e-2 | 4.18e-3 | -1.26e-5 | 1.184 |

As aforementioned, this study assumes the cycle length be fixed for the total time span *T* (e.g., a few hours). The effective green time for each phase *k* of cycle *i*, $g_k^i$, must sum up to the (fixed) cycle length *C,* as shown in Eq. (4.6). There are eight phases in Figure 4.1, so *K* = 8. Eq. (4.7) indicates the bounds of the green time $g_k^i$. For phases that can be skipped, $g_k^{min} = 0$. Eq. (4.8) (4.9) indicate phase 2 and 3 (and phase 6 and phase 7) cannot be realized for the same cycle *i*. At least one of the two variables, e.g., $g_2^i$ ($g_6^i$) and $g_3^i$ ($g_7^i$), need to be zero.

$$\sum_{k=1}^{K} g_k^i = C \qquad \forall\, i \in 1,2,\dots I \tag{4.6}$$

$$g_k^{min} \le g_k^i \le g_k^{max} \tag{4.7}$$

$$g_2^i * g_3^i = 0 \tag{4.8}$$

$$g_6^i * g_7^i = 0 \tag{4.9}$$

Each intersection contains multiple movements with each movement served by different phases $k$. Variable $S_{k,t}$ denotes the signal status at time $t$ for phase k, as shown in Eq. (4.10) . It takes one if the signal status at the current time is red and zero if it is green. The variable $\bar{k}$ is the current phase index at time t (i.e., 1, 2… 8). It represents the phase that is currently given the green light. Eq. (4.10b) reformulates (4.10) using two binary variables $y_{t,1}$and $y_{t,2}$ based on the big *M* concept.

$$S_{k,t} = \begin{cases} 0, & if\ \sum_{k=1}^{\bar{k}-1} g_k^i \le (t\ mod\ C) < \sum_{k=1}^{\bar{k}} g_k^i \\ 1, & otherwise \end{cases} \tag{4.10}$$

$$\begin{cases} \sum_{k=1}^{\bar{k}} g_k^i - (t\ mod\ C) + y_{t,1}M > 0 \\ \sum_{k=1}^{\bar{k}} g_k^i - (t\ mod\ C) \le (1 - y_{t,1})M \\ (t\ mod\ C) - \sum_{k=1}^{\bar{k}-1} g_k^i + y_{t,2}M \ge 0 \\ (t\ mod\ C) - \sum_{k=1}^{\bar{k}-1} g_k^i < (1 - y_{t,2})M \\ S_{k,t} = y_{t,1} + y_{t,2} \end{cases} \tag{4.10b}$$

Eq. (4.11) use indicator variables $y_{t,3}$and $y_{t,4}$ together to identify whether vehicle *n* is within the boundaries of the incoming approach: $\bar{d}_n^0$ and $\bar{d}_n^1$. Furthermore, the signal status $Z_{n,t}$ at time *t* for vehicle *n*, could be determined as long as the incoming approach of vehicle *n* is identified, as shown in Eq. (4.12). Noted that signal status $Z_{n,t}$ and $S_{k,t}$ are different. Vehicles coming from different approaches may encounter different signal status (red or green).

$$\begin{cases} d_{n,t} - \bar{d}_n^1 \le y_{t,3}M \\ d_{n,t} - \bar{d}_{n,}^1 + (1 - y_{t,3})M \ge 0 \\ d_{n,t} - \bar{d}_n^0 + y_{t,4}M \ge 0 \\ d_{n,t} - \bar{d}_n^0 \le (1 - y_{t,4})M \end{cases} \tag{4.11}$$

$$Z_{n,t} = S_{k,t} - \frac{S_{k,t}+(y_{t,3}+y_{t,4})-|S_{k,t}-(y_{t,3}+y_{t,4})|}{2} \quad (4.12)$$

The CV-based signal timing strategies in this paper require information on real-time vehicle trajectories. This project assumes the 100% penetration rate of connected vehicles. Vehicle trajectories can be transmitted when a vehicle enters the boundary of an intersection. Furthermore, to optimize signal timing for the current and future cycles, future vehicle trajectories are needed. For this, the Intelligent Driver Model (IDM) [78] is applied to simulate the vehicle trajectories. IDM is a car-following model that fits better with CV. We assume that there is only one lane per incoming approach, so there is no lane changing behavior involved. It is necessary to account for the signal status in the prediction of traffic flow propagation when applying IDM. For this, we model the red signal as a "standing vehicle" with speed equal to zero. It would disappear if the signal turns green. Eq. (4.13) indicates whether the front object of vehicle *n* is a real vehicle or a standing vehicle (traffic signal) by comparing the relative location of the front vehicle $n-1$, vehicle $n$, and the nearest traffic signal in front of vehicle *n*. The binary variable $y_{n,t}$ takes one if the front "vehicle" is the traffic signal (could be red or green) at location $d_{signal,n}$ with speed zero. If $y_{n,t}$is zero, the front vehicle $n-1$ is a real vehicle with location $d_{n-1,t}$ and speed $v_{n-1,t}$. This helps update the vehicle trajectories in IDM as shown later. Eq. (4.13b) reformulate (4.13) using the big *M* method and two binary variables $y_{n,t,1}$ and $y_{n,t,2}$.

$$y_{n,t} = \begin{cases} 1, if\ x_{n,t} < d_{signal,n} < x_{n-1,t} \\ 0, \qquad\qquad otherwise \end{cases} \quad (4.13)$$

$$\begin{cases} d_{signal,n} - d_{n-1,t} < y_{n,t,1}M \\ d_{signal,n} - d_{n-1,t} + (1-y_{n,t,1})M \geq 0 \\ d_{n,t} - d_{signal,n} < y_{n,t,2}M \\ d_{n,t} - d_{signal,n} + (1-y_{n,t,2})M \geq 0 \\ y_{n,t} = 1-(y_{n,t,1}+y_{n,t,2}) \end{cases} \quad (4.13b)$$

Eq. (4.14) - (4.15) identify the vehicle location and speed of the preceding "vehicle" $n-1$, which could be a real vehicle or the nearest front signal.

$$f_{n-1,t}^{d} = d_{n-1,t} * [1 - \frac{y_{n,t}+S_{n,t}-|y_{n,t}-S_{n,t}|}{2}] + d_{signal,n,t} * \frac{y_{n,t}+S_{n,t}-|y_{n,t}-S_{n,t}|}{2} \quad (4.14)$$

$$f_{n-1,t}^{v} = v_{n-1,t} * [1 - \frac{y_{n,t}+S_{n,t}-|y_{n,t}-S_{n,t}|}{2}] \quad (4.15)$$

Eq. (4.16) - (4.19)shows how IDM estimates the acceleration rate for vehicle *n* at each time interval, given the location and speed of vehicle $n-1$.

$$s_{n,t} = f_{n-1,t}^{d} - d_{n,t} - l_{n-1} \quad (4.16)$$

$$d_{n,t} = f_{n-1,t}^{v} - v_{n,t} \quad (4.17)$$

$$a_{n,t} = a[1 - \left(\frac{v_{n,t}}{v_0}\right)^{\delta} - \left(\frac{s^*(v_{n,t},d_{n,t})}{s_{n,t}}\right)^2] \quad (4.18)$$

$$s^*(v_{n,t}, d_{n,t}) = s_{on} + v_{n,t}H + \frac{v_{n,t}d_{n,t}}{2\sqrt{ab}} \quad (4.19)$$

Eq. (4.20) - (4.21) are applied to update the trajectories for vehicle *n* at next time interval $t+1$. More details of IDM can be found in Treiber *et al.* [78]. In this paper, the values of the parameters in IDM are chosen as $a = 1 m/s^2$, $b = 3\ m/s^2$, $s_{on} = 2$m, $H = 1.5$s, and $\delta = 4$, according to Khondaker and Kattan [79].

$$v_{n,t+1} = max(0, v_{n,t} + a_{n,t}) \quad (4.20)$$

$$d_{n,t+1} = d_{n,t} + \frac{v_{n,t}+v_{n,t+1}}{2} \quad (4.21)$$

Eq. (4.1) - (4.21) is a MINLP for the CV-based signal control problem. It clearly shows that when individual vehicle status is considered for signal control, e.g., under the CV environment, the problem can be formulated as a very complex MINLP. This is mainly due to the different status of vehicles and signal phases, as well as the various if-then-else types of conditions (e.g., equations (4.3), (4.10), (4.13)and others) inherent to this coupled signal-vehicle optimization problem. In

addition, since the variables of the model include the location and speed of each vehicle at each time interval, the dimension of the problem can be quite large. Furthermore, the IDM-based car-following model is also very complex. Thus, solving the model directly is quite challenging, and more tractable and efficient methods are needed. We next present one of such methods based on DP.

### 4.1.2 *Dynamic programming formulation*

DP provides a general framework to divide an optimization problem into multiple stages (under certain conditions), which could be solved sequentially one stage at a time. Here we divide the signal timing decisions into stages, one stage for a phase. We then approximate the total fuel consumption and travel time of a stage as functions of the state and decision variables. The notation is summarized as follows:

| | |
|---|---|
| $x_p$ | Decision variable, phase duration of stage $p$ (s). |
| $s_p$ | State variable, the total time from the beginning of the cycle to the end of stage $p$ (s). |
| $X_p(s_p)$ | The set of feasible control variable given stage variable $s_p$ at stage $p$ (s). |
| $V_p(s_p)$ | Value function, the cumulative value of the objective function from stage 1 up to stage $p$ ($). |
| $x^{min}$ | The minimum value of the decision variable (s). |
| $f_p(s_p, x_p)$ | Total cost at stage $p$, given state variable $s_p$, and decision variable $x_p$ ($). |
| $N_p$ | A total number of vehicles in phase $p$ (veh). |
| $FC_{n,t}(s_p, x_p)$ | Fuel consumption of the vehicle $n$ at time $t$ given stage variable $s_p$ and decision variable $x_p$ (gal/s). |

| | |
|---|---|
| $TT_{n,t}(s_p, x_p)$ | Travel time of vehicle $n$ at time $t$ given stage variable $s_p$ and decision variable $x_p$ (s). |
| $FC_{s,n,t}(s_p, x_p)$ | Fuel consumption of vehicle $n$ at moving status at time $t$ given stage variable $s_p$ and decision variable $x_p$ (gal/meter). |
| $FC_{I,n}(s_p, x_p)$ | Fuel consumption of vehicle $n$ at idle status at time $t$ given stage variable $s_p$ and decision variable $x_p$(gal/s). |
| $A_p(s_{p-1}, s_p)$ | The number of arriving vehicles in the time interval $[s_{p-1}, s_p]$. |
| $M_p(x_p)$ | The maximum number of vehicles that can be discharged during phase duration $x_p$. |
| $v_{n,t}(s_p, x_p)$ | Approximated speed of vehicle *n* at time *t* given the stage variable $s_p$ and decision variable $x_p$ (m/s). |
| $t_a$ | Arrival time at the predefined intersection boundary (distance *L* upstream of intersection) (s). |
| $t_d$ | The time when vehicle joins the queue (started to slow down) (s). |
| $t_0$ | The time when the vehicle fully stops (s). |
| $t_{ac}$ | The time when the vehicle starts to be discharged (s). |
| $t_l$ | The time when the vehicle achieves the free flow speed $v_0$(s). |
| $l_d$ | Distance upstream of the end of the queue or the stop line (if no queue) (m), e.g., 100m. |
| $V_p(s_p)$ | Value function at phase p given state variable $s_p$. |
| $\sigma$ | Tolerance of the fixed cycle length (s), e.g., 5s. |

As shown in Figure 3.2 and Figure 4.1, there are eight stages in total. The state variable $s_p$ is defined as the total number of time intervals from the beginning of the cycle to the end of stage *p*,

while the decision variable $x_p$ is the phase duration. Eq. (4.22) - (4.23) illustrate the relationship between the state variable and decision variable; see Sen and Head [51] for more details.

$$s_p = s_{p-1} + h(x_p) \tag{4.22}$$

$$h(x_p) = \begin{cases} 0, \ if \ x_p = 0 \\ x_p + r, other \ wise \end{cases} \tag{4.23}$$

Given the state variable $s_p$, the feasible set of decision variables could be determined based on Eq. (4.24):

$$X_p(s_p) = \begin{cases} 0, & if s_p < x^{min} \\ \{0, x^{min}, x^{min} + 1, \dots, s_p\}, & otherwise \end{cases} \tag{4.24}$$

To formulate the DP, we first assign the initial value function $V_0 = 0$. The DP starts from stage (phase) $p$ = 1, and proceed recursively to $p$ = 2, 3 … 8. At each stage, the method calculates the optimal decision variable $x_p^*(s_p)$ by minimizing the value function for each possible value of the state variable $s_p$ in the forward recursion. Note that after stage 1, phase 2 or phase 3 may be chosen for the optimal decision variable, but not both; the same rule applies to phase 6 and phase 7. After the decision variables are estimated for all stages, the optimal decision of each stage can be retrieved in the backward recursion. The DP calculation process can be equivalently represented by an acyclic graph, as shown in Figure 4.2.

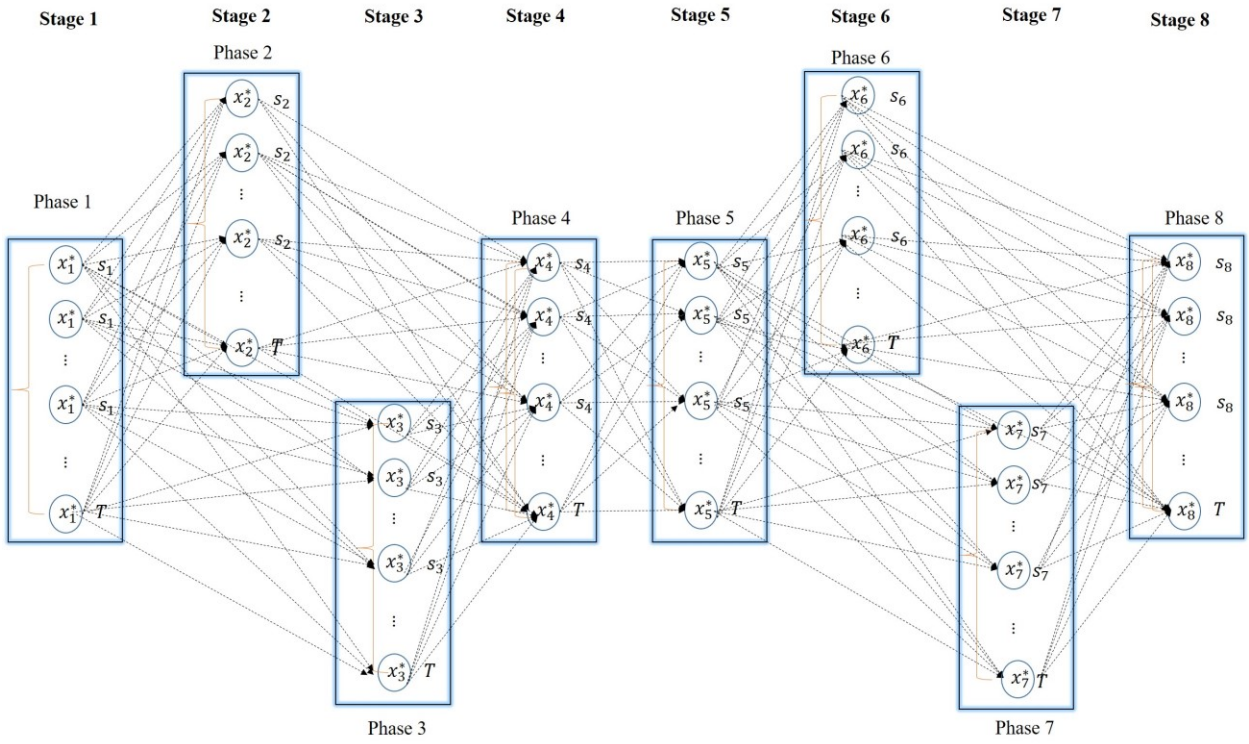



Figure 4.2. Acyclic graph of DP calculation process

However, in order to reformulate the signal control problem (4.1) - (4.21)as a DP, a critical condition is that the objective function in (4.1) can be expressed as the summation of the objective function of each stage. Furthermore, the stage-specific objective function (i.e., the sum of the vehicle fuel consumption and travel time of all vehicles in the stage) can be expressed as a function of the state and decision variables of that stage only [51]. This, however, is not true in general for most of the objectives we consider here, i.e., travel time or fuel consumption. It is especially so when we consider the data/information of individual vehicle (such as trajectories, speeds, delays, etc.). In the next subsection, we approximate the objective function of each stage so that it can be expressed as a function of the state and decision variables of the stage.

#### 4.1.2.1 Objective function approximation

Eq. (4.25) expresses the total fuel consumption and travel time of all the vehicles for phase $p$ (i.e., it is from time $s_{p-1}$ to $s_p$), where $N_p$ is the total number of vehicles in phase $p$. In this paper, the

travel time of a vehicle is estimated by the summation of free flow travel time of the vehicle and the delay it encountered. As shown previously [51], the total delay (and thus travel time) of a stage can be approximated as a function of the state and decision variables. We show in this subsection how the fuel consumption can be approximated as a function of the state and decision variables. Fuel consumption is a function of vehicle speed. Thus, vehicle speed should be approximated as a function of the state and decision variables.

$$Min \sum_{n=1}^{N_p} \sum_{s_{p-1}}^{s_p} f_p(s_p, x_p) \tag{4.25}$$

$$f_p(s_p, x_p) \triangleq m_F FC_{n,t}(s_p, x_p) + m_t TT_{n,t}(s_p, x_p) \tag{4.25b}$$

$$FC_{n,t}(s_p, x_p) = FC_{s,n,t}(s_p, x_p) * v_{n,t}(s_p, x_p) * (1 - I_{n,t}) + FC_{I,n} * I_{n,t} \tag{4.26}$$

The speed of a vehicle is estimated based on the queue discharging process. Let $A_p(s_{p-1}, s_p)$ denote the number of arriving vehicles for phase *p*, i.e., during the time interval $[s_{p-1}, s_p]$. $M_p(x_p)$ denotes the maximum number of vehicles that can be discharged during the time interval of green time $x_p$. As shown in Sen and Head [51], $A_p$ and $M_p$ can be expressed as functions of the state and decision variables of stage *p* only. We next show how the *speed* of a vehicle can be approximated as those variables. There are four possible cases for the speed of a vehicle arriving in stage *p*, as shown in Figure 4.3.

(1) The vehicle arrives during the green signal at stage p and can pass freely through the intersection.

$$\bar{v} = v_o \tag{4.27}$$

(2) The vehicle arrives during the green signal at stage p but a queue already exists. The queue includes vehicles that were not discharged in the previous stage *p*-1 and the newly arriving vehicles at current stage *p* before the current vehicle. Denote $t_a$ the time when the current vehicle arrives

at the predefined intersection boundary (distance $l$ upstream of intersection), $t_d$the time when the vehicle joins the queue (started to slow down), $t_0$ the time when the vehicle fully stops, $t_{ac}$ the time when the vehicle starts to be discharged, and $t_l$ the time when the vehicle achieves the free flow speed $v_0$ again after being discharged. We assume the vehicle starts to decelerate with a constant rate at the time $t_d$ if the queue exists. We can then approximate the average speed between $t_d$ and $t_0$ as $\frac{v_0}{2}$. The same assumption applies to the acceleration process from $t_{ac}$ to $t_l$. When the vehicle starts to decelerate at $t_d$, the distance between the vehicle and the stop line is denoted as $l_d$.

If the vehicle could pass the intersection within the current phase $p$ (trajectory ① in Figure 4.3 (b)), speed could be approximated using Eq. (4.28), otherwise (trajectory ②), the vehicle has to wait until the next phase. In this case, we only apply the first three conditions in Eq. (4.28) since $t_1 \geq s_p$. We could consider ② as a special case of ①.

$$v_{n,t}(s_p, x_p) = \begin{cases} v_0, & if\ t_a \leq t \leq t_d \\ \frac{v_0}{2}, & if\ t_d < t \leq t_0 \\ 0, & if\ t_0 < t \leq t_{ac} \\ \frac{v_0}{2}, & if\ t_{ac} < t \leq t_l \\ v_0, & if\ t_l < t \leq s_p \end{cases} \tag{4.28}$$

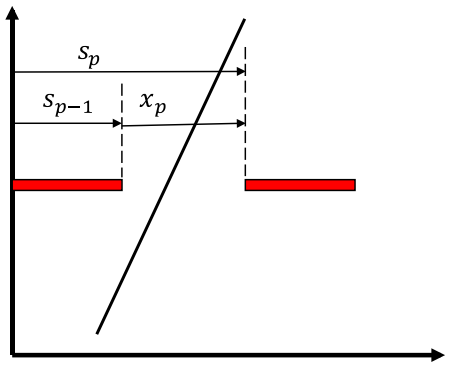
(3) The vehicle arrives during a red signal at stage *p*. Figure 4.3 (c) and (d) are differentiated by whether queue exists at the end of stage p. If there is no queue, vehicle n will start to leave at $t_1(t_1 = s_p)$, otherwise, it will have to wait for the queue to dissipate ($t_1 \geq s_p$). Here we only care about the vehicle status from time $s_{p-1}$ to $s_p$. The speed of vehicle $n$ arriving during red can be summarized as:

$$v_{n,t}(s_p, x_p) = \begin{cases} v_0, & if\ t_a \le t \le\ t_d \\ \frac{v_0}{2}, & if\ \ t_d < t \le t_0 \\ 0, & if\ t_0 < t \le s_p \end{cases} \tag{4.29}$$

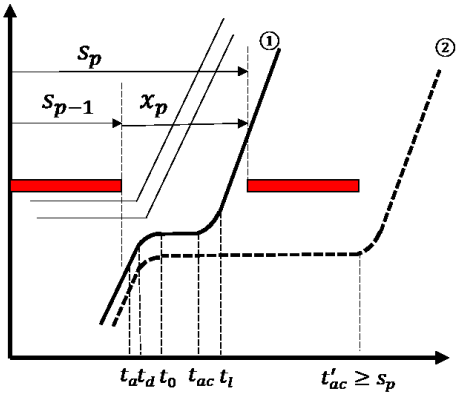
After approximating the vehicle speed $v_{n,t}(s_p, x_p)$ in the objective function for stage $p$ based on the above four cases, the objective function can be approximated as a function of $s_p$ and $x_p$ only, as shown below:

$$V_p(s_p) = min\{f_p(s_p, x_p) + V_{p-1}(s_{p-1}) | x_p \in X_p(s_p), p \in P\} \tag{4.30}$$

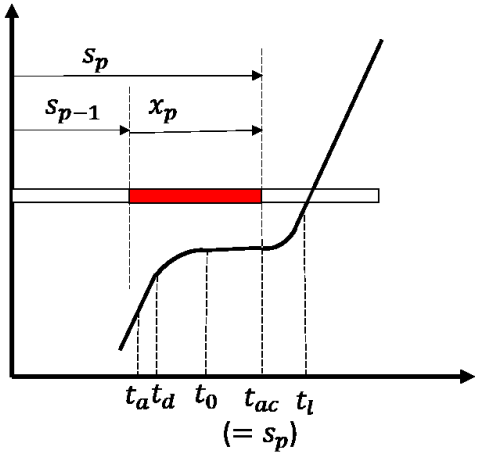
This means that Eq. (4.25), and the objective function of the MINLP in Section II as well, can be expressed as a function of $s_p$ and $x_p$ only. Thus, the problem can be reformulated as a DP. The detailed proof is straightforward and omitted here. Furthermore, after obtaining the optimal decision variable $x_p^*$ at stage p, IDM is applied to update the trajectories of all the vehicles from $s_{p-1}$ to $s_p = s_{p-1} + x_p^*$.



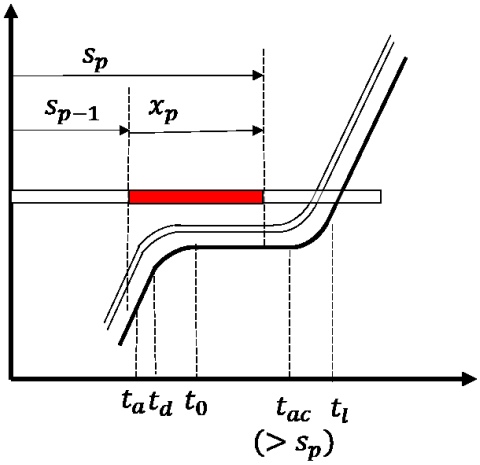

(a) Vehicle arrives during green signal (no queue)




(b) Vehicle arrives during green signal and queue exists




(c) Vehicle arrives during red signal (no queue)




(d) Vehicle arrives during red signal and queue exists

Figure 4.3. Four cases for approximating the vehicle average speed

#### 4.1.2.2 End Stage Cost

The DP formulation so far does not impose any constraint on the cycle length. As shown in the next section, imposing such a constraint will invalidate the DP formulation. We propose two steps to address this issue: adding the end-stage cost to the DP formulation and a branch-and-bound

method to refine the solution. First, in the last stage *P*, we modify the value function by adding the end stage cost $f_e$ if the difference between the estimated cycle length $C_e$ by DP and the predefined cycle length $C$ is large. If $\sigma$ denotes the tolerance, e.g., 5 seconds, and $w$ denotes the weight, the revised value function for the last stage *P* can be expressed as:

$$V_p(s_p) = min\{f_p(s_p, x_p) + V_{p-1}(s_{p-1}) + f_e | x_p \in X_p(s_p), p = 8\} \tag{4.31}$$

The end-stage cost $f_e$ can be expressed as in Eq. (4.32)while the estimated cycle length is the summation of the green time of all phases.

$$f_e = \begin{cases} 0, & if\ |C_e - C| \le \sigma \\ w(C_e - C)^2, & otherwise \end{cases} \tag{4.32}$$

$$C_e = \sum_{p=1}^{P} x_p \tag{4.33}$$

The end-stage cost in Eq. (4.32)with a proper weight works as a penalty function, ensuring that the obtained solution from the DP can result in a cycle length Ce close enough to the predefined cycle length C. To obtain a solution that can lead to C exactly, the second step applies a branch and bound method to refine the DP solution.

#### 4.1.2.3 Branch and bound algorithm

To produce a signal timing solution that leads to exactly the predefined cycle length *C*, we need to add the following constraint to the decision variables in the DP formulation.

$$\sum_{p=1}^{8} x_p = C \tag{4.34}$$

This constraint, however, will invalidate the DP model as decisions cannot be made stage by stage when Eq. (4.34) is considered. In other words, DP cannot guarantee that the optimal phase duration sums up to a fixed cycle length. A branch and bound method is applied here to resolve the issue, which was used in the past to solve the Resources Constrained Shortest Path (RCSP) problem [80]. The method creates a tree by selecting one variable each time from an initial

solution. Here the initial solution is produced by solving the DP formulation with the end-stage cost, as discussed above. The maximum level of the tree is the number of stages because the variable in a given set can only be used for branching once. Moreover, all branches at a given level of the tree have to be computed and analyzed before advancing to the next level. The numerical example and more explanations of the branch and bound method will be provided in the numerical results section. The branch and bound method can be summarized as below:

(1) DP with the end-stage cost is solved first to produce an initial solution.

(2) Define the error gain for stage/phase $p$:

$$ER_p(s_p, x_p) = f_p(s_p, x_p) \tag{4.35}$$

The error gain is the objective function in DP (total fuel consumption and travel time) for all vehicles traveling in stage p with stage variable $s_p$ and decision variable $x_p$.

(3) If the estimated cycle length happens to be exactly the predefined cycle length C, the DP solution is an optimal solution. The algorithm stops.

(4) If the cycle length from the DP solution is larger than the predefined fixed cycle length (decision variable should be decreased), the selected phase for branching at each level is the one that has the minimum error gain. There could be a multiple numbers of branching depending on the difference between the predefined cycle length and the cycle length from the DP solution. If the produced cycle length is less than the fixed cycle length (decision variable should be increased), the selected phase for branching would be the maximum error gain.

(5) The algorithm stops when the results of all branching (i.e., leaves) are feasible. The optimal solution is selected from the feasible solutions that satisfy the cycle length constraint while producing the minimum objective value.

## 4.2 CORRIDOR LEVEL OPTIMIZATION[2]

This section aims to optimize the signal timing and coordination parameters of multiple intersections in a traffic corridor by taking advantages of V2I and V2V communications in the CV environment. All the vehicles in the corridor are assumed to be connected to the infrastructure so that all of their trajectories (second-by-second speed and location) are available in real time to transfer to signal controllers for signal timing optimization. In order to account for the coordination of vehicles, the fixed cycle length constraint is applied to the problem. The overall CV-based signal control and coordination problem are first formulated as a MINLP. The objective is to minimize the total weighted sum of fuel consumptions and travel times of all vehicles in the corridor by calculating the optimal phase durations and offsets at the same time. Since the MINLP formulation has a large dimension and contains complex constraints, directly solve the problem is challenging. We then decompose the problem into a CV-based two-level traffic signal optimization and coordination scheme that contains an intersection level and a corridor level. Under intersection level, a fixed cycle length is assumed for individual intersections so that the coordination can be achieved for multiple intersections. We first formulate the CV-based signal control problem as a MINLP (optimize phase duration for a single intersection). Due to the large dimension of the problem and the complexity of the nonlinear car-following model, solving the nonlinear program directly can be challenging. Secondly, we reformulate the problem as a DP model. We note that imposing the fixed cycle length constraint will invalidate the DP formulation. We then apply a two-step method to resolve this issue: end-stage cost and branch and bound algorithm. At the corridor level, the "temporary" optimal offsets are calculated iteratively in order to find the optimal

[2]This chapter is an amended version of Li and Ban [105]. © [2017] IEEE. Reprinted, with permission, from [105].

offsets when the total cost converges to its minimum. In order to solve such a two-level model, we develop a prediction-based approach that collects the arrival vehicle information at the beginning of each cycle and calculates the optimal phase durations for each intersection using a DP method. During the calculation process, each intersection is aware of other intersections' decisions, traffic conditions, and the "temporary" optimal offsets. This ensures that the traffic flows on the main street are coordinated among adjacent intersections.

The proposed model is different from the two-level optimization method developed by Beak et al [40] in several important aspects. First, at the intersection level, Beak et al. [40] applied a fixed force-off option and added more coordination constraints to the DP method to ensure a fixed cycle length. Such a method is itself in a two-level structure, making the overall framework in Beak et al. [40] essentially a three-level model. The proposed two-level model in this paper uses DP with a two-step method to find the optimal phase durations that satisfy the fixed cycle length constraint for a single intersection, as shown in Li and Ban [41]. At the corridor level, Beak et al [40] developed a MILP to minimize the delay of a vehicle platoon. This paper accounts for the individual vehicle's trajectories when determining the optimal offsets, and thus considers more detailed information than the corridor-level model in Beak et al. [40].

### 4.2.1 *Formulating signal coordination as a MINLP*

As shown in section Mixed-Integer Nonlinear Program, the signal control problem for a single intersection can be formulated as a MINLP. It can produce the optimal phase durations that satisfy fixed cycle length constraint. The previous model can be extended to a corridor level by introducing the offset variable $O_j$ for intersection $j$, when considering the coordination of multiple intersections on the corridor. The objective is to minimize the weighted sum of fuel consumption

and travel times for all vehicles (the total number is *N*) on the main street for the total time span *T*, as shown in Eq. (4.36). $FC_{n,t}$ and $TT_{n,t}$ are the fuel consumption and travel time for vehicle $n$ at time *t*. The corresponding parameters $m_F$ and $m_{TT}$ are the "value of fuel" and "value of time" respectively. The fuel consumption $FC_{n,t}$ can be estimated using Eq.(4.2) - (4.5).

$$min\, F = \sum_{t=1}^{T} \sum_{n=1}^{N} \left( m_F FC_{n,t} + m_{TT} TT_{n,t} \right) \tag{4.36}$$

In order to coordinate multiple intersections, we need to establish connections between coordinated phases for different traffic signals, e.g., the start of the green time of phase 1 (movement 2 and 6 in Figure 4.1) for each intersection. These connections are represented by two types of time stamps: global time *t* and local time $t_j'$. Global time index t refers to the master clock to which each signal is referenced during coordinated operations. It starts from zero up to the total time span *T*. Local time $t_j'$ refers to the time stamp for a local intersection j starting from zero up to cycle length C. The relationship between t and $t_j'$ can be present by Eq. (4.37):

$$t_j' = mod\left(t - O_j,\ C\right), 0 \le t_j' \le C \tag{4.37}$$

where *mod* refers to modulo operation. Offset $O_j$ is the time difference of the starting times of the coordinated phase between the reference signal and signal j. The boundary condition is shown in Eq. (4.38), where $O^{max}$ denotes the parameter of maximum offset.

$$0 \le O_j \le O^{max} \tag{4.38}$$

The fixed cycle length constraint is necessary in order to conduct signal coordination. It requires the effective green time for each phase *k* of cycle *i* at intersection *j* sums up to the fixed cycle length *C*, as shown in Eq. (4.39). Eq.(4.40) shows the boundary of green time at phase k of cycle i at intersection j. Eq. (4.41) - (4.42) indicates that phase 2 and 3 (phase 6 and 7) cannot be

realized in the same cycle at intersection j. In other words, one of the two variables, e.g., $g_{j,2}^{i}$ ($g_{j,6}^{i}$) and $g_{j,3}^{i}$ ($g_{j,7}^{i}$), need to be zero.

$$\sum_{k=1}^{K} g_{j,k}^{i} = C \qquad \forall\, i_j \in 1,2,\dots I \tag{4.39}$$

$$g_{j,k}^{min} \le g_{j,k}^{i} \le g_{j,k}^{max} \tag{4.40}$$

$$g_{j,2}^{i} * g_{j,3}^{i} = 0 \tag{4.41}$$

$$g_{j,6}^{i} * g_{j,7}^{i} = 0 \tag{4.42}$$

In order to estimate the fuel consumption and travel time of all vehicles in the network, we need to know the signal status at each intersection at each time stamp because it will influence vehicle trajectories. Variable $S_{j,k,t}$ denotes the signal status at time $t$ for phase $k$ at intersection $j$, as shown in Eq. (4.43). It takes one if the signal status at time $t_j'$ at intersection $j$ is red and zero if it is green. The variable $\bar{k}$ is the current phase index at time $t_j'$ (i.e., 1, 2… 8). It represents the phase that is currently given the green light. Eq. (4.43b) reformulates (4.43) using two binary variables $y_{t,1}$and $y_{t,2}$ based on the big $M$ concept.

$$S_{j,k,t} = \begin{cases} 0, & if\ \sum_{k=1}^{\bar{k}-1} g_{j,k}^{i} \le t_j' < \sum_{k=1}^{\bar{k}} g_{j,k}^{i} \\ 1, & otherwise \end{cases} \tag{4.43}$$

$$\begin{cases} \sum_{k=1}^{\bar{k}} g_{j,k}^{i} - t_j' + y_{t,1} M > 0 \\ \sum_{k=1}^{\bar{k}} g_{j,k}^{i} - t_j' \le (1 - y_{t,1}) M \\ t_j' - \sum_{k=1}^{\bar{k}-1} g_{j,k}^{i} + y_{t,2} M \ge 0 \\ t_j' - \sum_{k=1}^{\bar{k}-1} g_{j,k}^{i} < (1 - y_{t,2}) M \\ S_{j,k,t} = y_{t,1} + y_{t,2} \end{cases} \tag{4.43b}$$

After identifying the signal status at each intersection at each time stamp, we need to find the nearest front signal of vehicle n based on the relative positions of signals and itself. Eq. (4.44) use indicator variables $y_{t,3,j}$and $y_{t,4,j}$ together to identify whether vehicle n is within the boundaries of the intersection $j$: $\bar{d}_{j,n}^{0}$ and $\bar{d}_{j,n}^{1}$. It helps identify the nearest traffic signal in front of

vehicle *n* at time *t*. Furthermore, the signal status $Z_{n,t}$ at time *t* for vehicle *n* could be determined as long as the nearest traffic signal $\bar{J}$ in front of vehicle *n* is identified, as shown in Eq. (4.46).

$$\begin{cases} d_{n,t} - \bar{d}^1_{j,n} \le y_{t,3,j} M \\ d_{n,t} - \bar{d}^1_{j,n} + (1 - y_{t,3,j}) M \ge 0 \\ d_{n,t} - \bar{d}^0_{j,n} + y_{t,4,j} M \ge 0 \\ d_{n,t} - \bar{d}^0_{j,n} \le (1 - y_{t,4,j}) M \end{cases} \tag{4.44}$$

$$\bar{J} = \sum_{j=1}^{J} j(1 - (y_{t,3,j} + y_{t,4,j})) \tag{4.45}$$

$$Z_{n,t} = S_{\bar{J},k,t} - \frac{S_{\bar{J},k,t} + (y_{t,3,\bar{J}} + y_{t,4,\bar{J}}) - |S_{\bar{J},k,t} - (y_{t,3,\bar{J}} + y_{t,4,})|}{2} \tag{4.46}$$

As long as we identify the signal status of the nearest signal in front of vehicle *n*, its trajectories can be estimated and predicted based on the intelligent driving model, i.e., IDM, which follows the same principals in the single intersection model in Eq. (4.14) - (4.21). Eq. (4.47) - (4.55) are identical to Eq. (4.14) - (4.21). They are listed in this section in order to show the complete MINLP.

$$y_{n,t} = \begin{cases} 1, if\ x_{n,t} < d_{signal,n} < x_{n-1,t} \\ 0, \qquad\qquad otherwise \end{cases} \tag{4.47}$$

$$\begin{cases} d_{signal,n} - d_{n-1,t} < y_{n,t,1} M \\ d_{signal,n} - d_{n-1,t} + (1 - y_{n,t,1}) M \ge 0 \\ d_{n,t} - d_{signal,n} < y_{n,t,2} M \\ d_{n,t} - d_{signal,n} + (1 - y_{n,t,2}) M \ge 0 \\ y_{n,t} = 1 - (y_{n,t,1} + y_{n,t,2}) \end{cases} \tag{4.47b}$$

$$f^d_{n-1,t} = d_{n-1,t} * [1 - \frac{y_{n,t} + Z_{n,t} - |y_{n,t} - Z_{n,t}|}{2}] + d_{signal,n,t} * \frac{y_{n,t} + Z_{n,t} - |y_{n,t} - Z_{n,t}|}{2} \tag{4.48}$$

$$f^v_{n-1,t} = v_{n-1,t} * [1 - \frac{y_{n,t} + Z_{n,t} - |y_{n,t} - Z_{n,t}|}{2}] \tag{4.49}$$

$$s_{n,t} = f^d_{n-1,t} - d_{n,t} - l_{n-1} \tag{4.50}$$

$$d_{n,t} = f^v_{n-1,t} - v_{n,t} \tag{4.51}$$

$$a_{n,t} = a[1 - \left(\frac{v_{n,t}}{v_0}\right)^{\delta} - \left(\frac{s^*(v_{n,t}, d_{n,t})}{s_{n,t}}\right)^2] \tag{4.52}$$

$$s^*(v_{n,t}, d_{n,t}) = s_{on} + v_{n,t}H + \frac{v_{n,t}d_{n,t}}{2\sqrt{ab}} \tag{4.53}$$

$$v_{n,t+1} = max(0, v_{n,t} + a_{n,t}) \tag{4.54}$$

$$d_{n,t+1} = d_{n,t} + \frac{v_{n,t}+v_{n,t+1}}{2} \tag{4.55}$$

Eq. (4.36) - (4.55) is a MINLP for the CV-based signal optimization and coordination for multiple intersections on a traffic corridor. When trajectories of the individual vehicle are considered, it clearly shows that the problem can be formulated as a MINLP. Compared to the single intersection model in section 4.1.1, model (4.36) - (4.55) is even more complex due to the relationships between different statuses of vehicles and multiple traffic signals, various if-then-else conditions, and the large dimension of the problem (considering the location and speed of each CV). As a result, for a large corridor or a road network, the MINLP formulation is hard to solve and not applicable for real-time signal control. When solving the MINLP directly, a local optimum is often produced instead of the global optimal solution. In the next section, the traffic signal optimization and coordination problem (4.36) - (4.55) is decomposed into a decentralized two-level model that is able to solve the problem more efficiently.

### 4.2.2 *Reformulating signal coordination as a two-level model*

Due to the complex of the MINLP formulation (4.36) - (4.55), the MINLP is reformulated as a decentralized two-level problem. In this two-level model, instead of solving the phase durations and offsets optimization for multiple intersections as one mathematical program, we decompose the overall problem into two levels: an intersection level to optimize the phase durations for every single intersection and a corridor level to optimize the offsets. As a result, the complexity of the original mathematical problem is significantly reduced.

Figure 4.4 shows the overall framework of the two-level model. It is assumed that all intersections and vehicles are connected, and they can send messages to each other. It further assumes some initial offsets and phase durations are available, e.g., those generated by SYNCHRO, a widely used traffic signal design and optimization software tool. The arrival vehicles to each intersection can then be estimated and predicted using IDM. Such information is a critical input to the DP method as detailed in Section 4.1.2 to optimize the timing plans for single intersections. In particular, the DP method is applied to compute the optimal phase durations in every cycle of each intersection by minimizing the total weighted sum of fuel consumption and travel time of all CVs within the boundary of the intersection, e.g., 400m upstream and 400m downstream of the intersection. Section 4.1.2 shows that DP with a two-step method (end-stage cost and branch and bound algorithm) is able to generate the phase durations that guarantee the given fixed cycle length for a single intersection. Figure 4.2 illustrates the DP calculation process, which can be equivalently represented as an acyclic graph. Solving the DP is to find the shortest path in the graph. The state variable $s_p$ here is defined as the total number of time intervals from the beginning of the cycle to the end of stage $p$, while the decision variable $x_p$ is the phase duration. The variable offset provides the connection between the intersection and corridor level optimization. In order to consider the influence of offsets on DP implementation for an intersection, we follow the rules in Eq.(4.37) that DP for intersection $j$ will be implemented using its local time $t_j'$. This essentially reduces the model (4.36) - (4.55) for the entire corridor to each individual intersection.

After the optimal signal plans for all intersections on a corridor are generated for a few cycles, such information is transferred to a central controller where offset optimization is conducted at the corridor level. Usually, the offset values are maintained for a period of time and not updated in

every cycle. Here the offsets are assumed to be updated in every *Nc* cycles in Figure 4.4, e.g., in every 10 cycles. The offset optimization problems are further reformulated as a MINLP in this section, but with a much fewer number of variables since the phase durations for each intersection are given from solving the intersection-level model (using the DP method in Section 4.1.2) above. The objective of the corridor-level optimization is to minimize the total weighted sum of fuel consumption and travel time of all CVs on the main street of the corridor:

$$min\, F = \sum_{t=1}^{T} \sum_{n=1}^{\bar{N}} \left( m_F FC_{n,t} + m_{TT} TT_{n,t} \right) \tag{4.56}$$

where $\bar{N}$ is the total number of vehicles on the main street. The constraints on the offsets are identical to those in Eq. (4.37) - (4.38). Since the optimal green time for phase $k$ of cycle $i$ at intersection $j$ is generated from the intersection-level model, the variable $g_{j,k}^{i}$ in the original MINLP model (4.36) - (4.55) becomes a parameter $\bar{g}_{j,k}^{i}$ in the corridor-level model. The signal status at intersection $j$ for phase $k$ at time $t$ can be estimated from Eq. (4.57). Eq. (4.57b) reformulate Eq. (4.57) using big *M* constraint.

$$S_{j,k,t} = \begin{cases} 0, & if\ \sum_{k=1}^{\bar{k}-1} \bar{g}_{j,k}^{i} \le t_j' < \sum_{k=1}^{\bar{k}} \bar{g}_{j,k}^{i} \\ 1, & otherwise \end{cases} \tag{4.57}$$

$$\begin{cases} \sum_{k=1}^{\bar{k}} \bar{g}_{j,k}^{i} - t_j' + y_{t,1} M > 0 \\ \sum_{k=1}^{\bar{k}} \bar{g}_{j,k}^{i} - t_j' \le (1 - y_{t,1}) M \\ t_j' - \sum_{k=1}^{\bar{k}-1} \bar{g}_{j,k}^{i} + y_{t,2} M \ge 0 \\ t_j' - \sum_{k=1}^{\bar{k}-1} \bar{g}_{j,k}^{i} < (1 - y_{t,2}) M \\ S_{j,k,t} = y_{t,1} + y_{t,2} \end{cases} \tag{4.57b}$$

After the signal status is identified, vehicle trajectories can be estimated from IDM using Eq. (4.47) - (4.55). This constitutes the corridor-level model that is also a MINLP but much simpler than the original MINLP (4.36) - (4.55). The corridor-level model has much fewer numbers of variables (only the offsets) and can be solved by the NOMAD solver in Matlab. NOMAD uses a

Mesh Adaptive Direct Search algorithm to solve non-differentiable and global nonlinear programs. It computes/updates vehicle speed and location at each time step using IDM. As shown in Figure 4.4, in order to find the optimal offsets, "temporary" optimal offsets generated from the corridor-level MINLP are calculated iteratively until the total cost converges. The convergence criterial is defined as the difference between the total costs estimated from two consecutive iterations, i.e., when $\Delta TC$ is less than a small tolerance, e.g., $10^{-5}$.

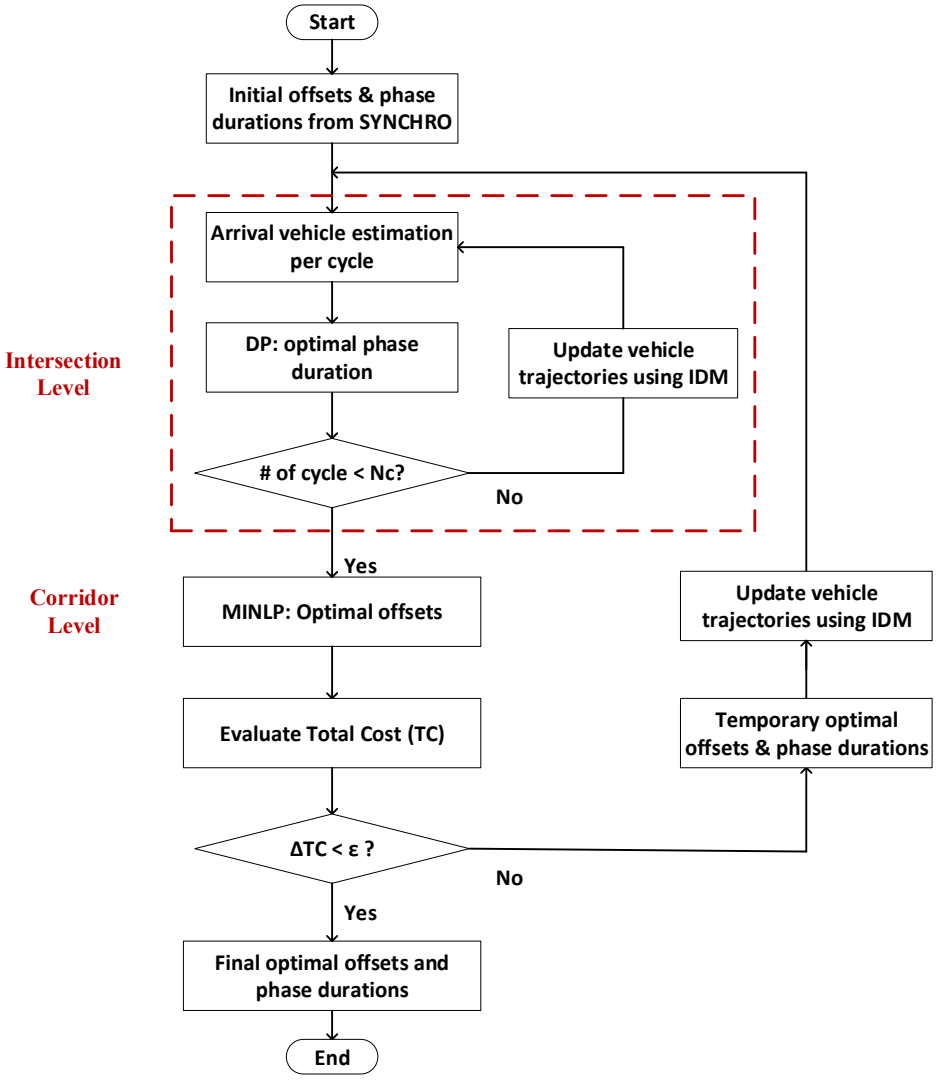



Figure 4.4. Solution technique of the two-level traffic signal optimization model

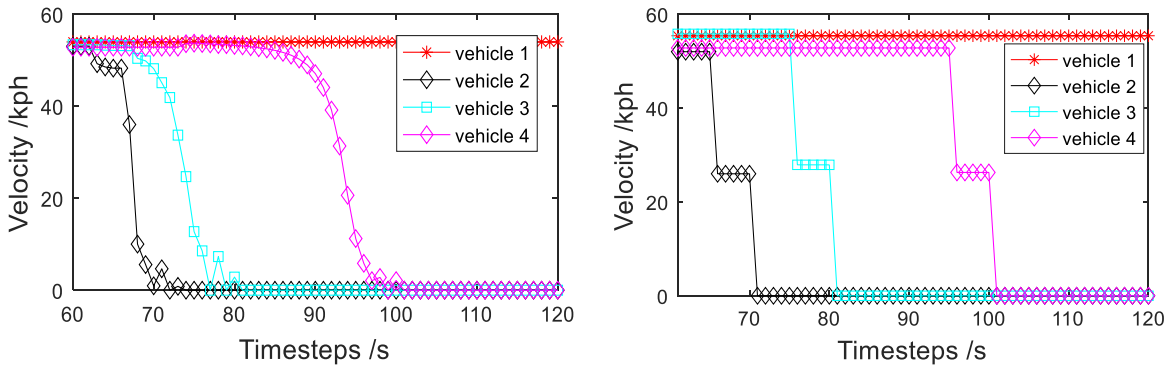
## 4.3 NUMERICAL EXPERIMENT

### 4.3.1 *Single Intersection*

The proposed CV-based signal timing optimization model and the DP method are tested using data generated via traffic simulation in VISSIM. The testing network contains a single intersection with a boundary of 300 meters upstream and downstream of the intersection to mimic the communication range of V2I. The $\sigma$ parameter in Eq. (4.32) is set as 5. After comparing the approximated vehicle speed using our approximation method and the speed generated from VISSIM, we evaluate the proposed signal timing optimization method in three steps. First, we estimate the optimal cycle length using SYNCHRO, a widely used traffic signal design and optimization software tool, for different traffic demand cases. Next, for a given case, we apply different methods to optimize the signal timing plans (phases and green splits), one for each method. We then evaluate the performance of each signal timing plan (i.e., each method) by applying the plan to the intersection and using IDM to generate vehicle trajectories, based on which to calculate the total cost. We also illustrate the procedure of the branch and bound algorithm and test the impact of the tolerance in the end-stage cost Eq. (4.32) on the performance of the proposed model.

#### 4.3.1.1 Speed approximation

In the proposed method, we approximate the vehicle speed as a function of the state variables and decision variables, as shown in section 4.1.2.1. Here we compare the vehicle speed between our method in Eq. (4.27) - (4.30) and IDM simulation for the major streets (link 2 and link 4) in Figure 4.5. The platoon contains four vehicles. It is observed that the approximate speeds have the similar trend but not as smooth as the speed profile in IDM.




(a) IDM speed (b) Approximated speed

Figure 4.5. Speed Comparisons

4.3.1.2 Signal timing optimization

Different combinations of traffic demand levels and vehicle types are tested in order to evaluate the proposed signal optimization method. Vehicle arrivals at the boundaries of the intersection are generated from VISSIM. For each direction, 80% of the vehicles use through lanes and the others use left lanes. Six cases are tested to identify the influence of traffic demands and vehicle types on the model performance. In Case I – III, vehicle demand is set to be 250 vph, 500 vph, and 800 vph, respectively. All vehicles are sedans (Type 5 in Table 4.1). In Case IV - VI, traffic demands are identical as in Case I - III, but the vehicle types are assigned differently. In the N-S directions, vehicles are assigned as EV, i.e., Vehicle type 1, while in the W-E directions, vehicles are assigned as buses (Vehicle type 7).

We test all models for 10 cycles. There are vehicles randomly generated to enter the network during the first 8 cycles, while during the last 2 cycles, there is no traffic demand. This guarantees the network is cleared by the end of the simulation. We test three signal control methods, as shown in Table 4.2 and Table 4.3. The first method is the actuated signal timing plan produced by SYNCHRO. The timing plan from SYNCHRO is then applied in IDM to update vehicle trajectories and estimate the objective function value, i.e., the total cost of fuel consumption and

travel time using Eq. (4.1) - (4.5). The same procedure also applies to the other two models: the NOMAD solver in MATLAB and the DP method proposed here. The solver "NOMAD" uses a Mesh Adaptive Direct Search algorithm to solve non-differentiable and global nonlinear programs. It can solve non-convex MINLPs while it may not produce the global optimal solution. Since the starting point will directly influence the optimization results in NOMAD, we set the solution from SYNCHRO and DP respectively as the starting point in NOMAD to further reduce the cost. Dimensionality is another key factor affecting the performance of NOMAD. We test the model by updating the signal plan in various updating intervals (e.g., in every 1, 2, 5 or 10 cycles). The number of variables in NOMAD increases as the update frequency increases. For example, if we update the signal plan every cycle, there will be 80 variables in the total time span (10 cycles). The third model is the DP with the proposed end-stage cost and the branch and bound method. DP update the signal plan in every cycle. The results from DP without fixed cycle length constraints are also shown in the parentheses in the table.

Table 4.2 and Table 4.3 summarize the results, in terms of the objective values, by different models for the six cases respectively, considering the influence of demand levels only (Table 2) and the combined demand levels and vehicle types (Table 4.2). The cycle lengths for different demand levels are determined by SYNCHRO: they are 60s for low traffic demand (250 vph), 65s for medium traffic demand (500 vph), and 85s for high demand (800vph), which are also the maximum value for the state variables in DP. Table 4.2 shows the cost from different models under various demand levels. NOMAD produces different solutions for different signal plan updating intervals and the starting points (from SYNCHRO in first four rows and from DP in the last row). The best solutions from NOMAD are highlighted for each Case. Table 4.2 shows that if the signal plan is updated every cycle or every two cycles, the network performance is not improved since

the total costs keep the same as its initial evaluation from the initial guess (SYNCHRO plan). However, a better solution may be obtained as the updating frequency decreases to every 5 or 10 cycles. This indicates that NOMAD has difficulties finding optimal solutions when the number of variables is relatively large. Table 4.2 also shows that different starting points in NOMAD can affect the optimal solutions. When compared with different models, DP outperforms SYNCHRO and achieves the same solution as NOMAD under low and medium demands in case I and case II. In case III, as the demand increases, DP results are still better than SYNCHRO but slightly worse than NOMAD. Table 4.3 shows the costs from different models under various demand levels and vehicle types. By comparing Case IV to Case I, we can see that the performance improvements of DP and NOMAD over SYNCHRO are more dramatic, which is also shown in Figure 4.6. Case IV incorporates buses in W-E direction, which produces much more fuel consumption than the sedan in Case I. The cost generated from DP and NOMAD were both lower than SYNCHRO because SYNCHRO does not consider the influence of vehicle types on the fuel consumption when optimizing the signal timing. As the demand increases while still maintaining different vehicle types on NS and WE directions, the performance of DP is still better than SYNCHRO but not as good as NOMAD in medium (500vph) and high (800 vph) demand levels. Since NOMAD uses the optimal phase plan from SYNCHRO and DP as the starting point, it makes sense that the best NOMAD results are always better than or equal to SYNCHRO and DP for all cases. It is also observed that the results from DP without the fixed C constraint always have lower objectives compared with the one with the fixed C constraint. This indicates that enforcing the fixed cycle length, i.e., for signal coordination purposes (and thus to the benefit of the entire system), may likely worsen the performance of individual intersections.

Table 4.2. Cost of Different Models under Various Demand Levels

| Model | Case I | Case II | Case III | # of variables |
|---|---|---|---|---|
| | 250 vph, sedan | 500 vph, sedan | 800 vph, sedan | |
| 1. SYNCHRO | 49.64 | 162.39 | 368.99 | / |
| 2. NOMAD update every cycle | 49.64 | 162.39 | 368.99 | 80 |
| NOMAD update every 2 cycles | 49.64 | 162.39 | 368.99 | 40 |
| NOMAD update every 5 cycles | 49.64 | 153.25 | 362.5 | 16 |
| NOMAD update every 10 cycle | 48.12 | 143.52 | 358.05 | 8 |
| NOMAD update every 10 cycles & initial points from DP solution) | 47.74 | 140.65 | 359.96 | 8 |
| 3. DP with fixed C constraint | 47.74 | 140.65 | 360.73 | / |
| (DP without fixed C constraint) | 46.97 | 138.55 | 354.74 | |

Table 4.3 Cost of Different Models under Various Demand Levels and Vehicle Types

| Model | Case IV | Case V | Case VI | # of variables |
|---|---|---|---|---|
| | 250 vph; NS: Evs; WE: Bus | 500 vph; NS: Evs; WE: Bus | 800 vph; NS: Evs; WE: Bus | |
| 1. SYNCHRO | 68.42 | 209.54 | 453.81 | / |
| 2. NOMAD update every cycle | 68.42 | 209.54 | 453.81 | 80 |
| NOMAD update every 2 cycles | 68.42 | 209.54 | 453.81 | 40 |
| NOMAD update every 5 cycles | 68.42 | 168.29 | 453.81 | 16 |
| NOMAD update every 10 cycle | 65.78 | 180.67 | 434.37 | 8 |
| NOMAD update every 10 cycles & initial points from DP solution) | 63.37 | 169.54 | 436.65 | 8 |
| 3. DP with fixed C constraint | 63.37 | 172.43 | 436.65 | / |
| (DP without fixed C constraint) | 61.25 | 163.54 | 433.36 | |

Figure 4.6 shows the performance improvements of NOMAD, DP with fixed cycle length and DP without fixed cycle length from SYNCHRO. As shown in the dashed line, the model improvements of low and high demand levels are not as significant as the middle demand levels. This may be because under unsaturated but relatively heavy traffic conditions, there are more opportunities to optimize the splits and reduce the total cost of fuel consumption and travel time. Such opportunities tend to diminish when traffic is very light (all methods can work well) or very heavy (no method can work well). Furthermore, the performance improvements are more obvious

if considering different vehicle types, as shown in Case IV – VI in Figure 4.6.

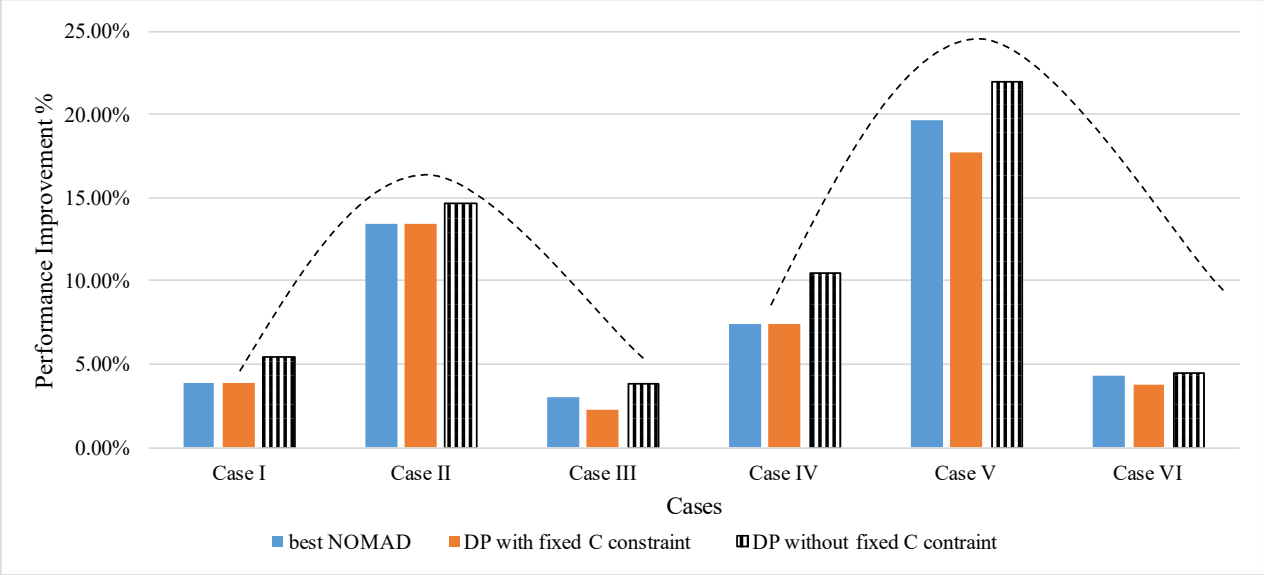



Figure 4.6. Improvement of Model Performance over SYNCHRO Results

Figure 4.7 shows the cost of fuel consumption and travel time separately for the four methods and six cases. For all cases, the cost of travel time is much larger than the cost of fuel consumption. Compared cases IV, V, VI to cases I, II, III, it is observed that the influence of vehicle types is more significant on the cost of fuel consumption than travel time. As shown in Figure 4.7, considering the same level of travel demand (e.g., case I and IV), the cost of fuel consumption is larger for the cases considering different vehicle types while the costs of travel time are similar.

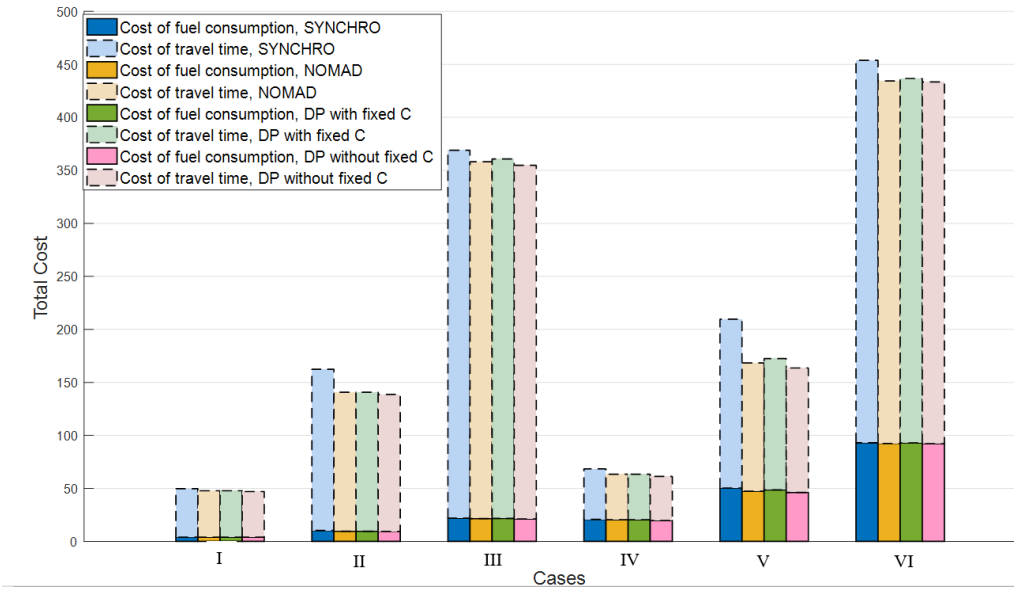
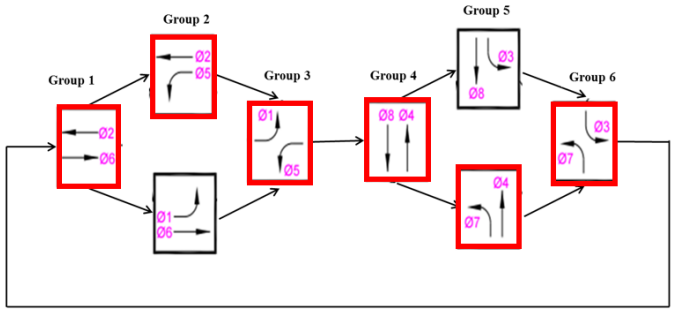



Figure 4.7. Total cost comparisons

4.3.1.3 Branch and bound algorithm

This section illustrates how the branch and bound algorithm can be applied to the DP results (without the fixed cycle length constraint) to guarantee a solution with the fixed cycle length. Here we use the DP result of Case I in Table 4.1. Figure 4.8 shows the result from the DP by considering the end stage cost, which leads to a cycle length of 57s, 3 seconds lower than the predefined and fixed cycle length of 60s.




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $x_p$ | 5 | 11 | 10 | 9 | 11 | 11 | 57s < C = 60s |
| EG (from DP) | 0.58 | 0.93 | 0.88 | 0.84 | 0.94 | 0.92 | TC= 4.76 (using IDM) |

Figure 4.8. The estimated solution from DP

With the information of the DP solutions, the estimated Error Gain (EG) from DP, the total cost of fuel consumption and travel time, and the fixed cycle length, the branching could be selected based on the value of EG. In this case, since the phase group 5 (phase 7) has the maximum EG, it is selected as the first level of branching, guided by the algorithm presented in the previous section. The nodes 1 – 3 enumerate each possible value of the decision variable for phase 7 in phase group 5, with the green time for phase 7 being 12, 13, 14, respectively. Only node 3 is feasible since its phase durations add up to the fixed cycle length 60s. For node 1 – 2, the next level of branching is generated by the same rule. Figure 4.9 is the tree for branch and bound that lists all the feasible solutions with the total cost estimated using IDM. The feasible solution with the minimum objective value (TC) represents the optimal solution. It is observed that the total cost for the optimal solution in Figure 4.9 is larger than the total cost of the initial solution from DP (C=57s) because we sacrifice the signal performance by adding a fixed cycle length constraint using branch and bound method. The signal timing parameter estimation algorithm as follows:

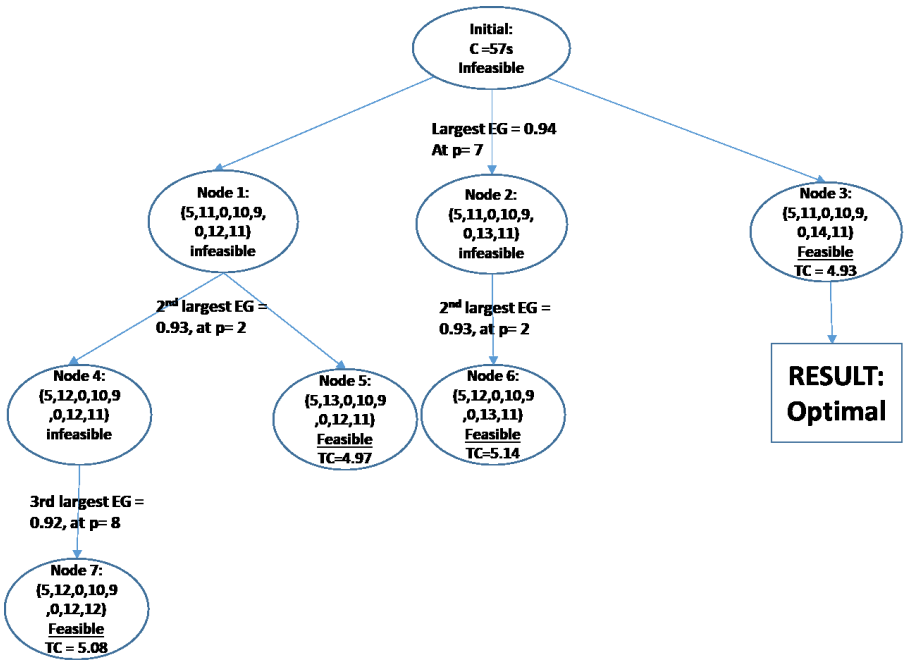



Figure 4.9. Branch and Bound Tree

4.3.1.4 Tolerance parameter of a branch and bound method

In Eq. (4.32), we define a tolerance $\sigma$ to calculate the end stage cost to find the DP solution. Different values of $\sigma$ may produce difference intial solutions and influence the number of evaluations in the Branch and Bound algorithm. Here an evaluation means that for a given tentative signal timing plan, we need to estimate the objective in Eq. (4.32) using IDM, which may be time-consuming. As shown in Figure 4.9, one node in the graph corresponds to one signal timing plan that needs to be evaluated. We test the tolerance $\sigma$ from 0 to 10 and test the performance of the algorithm for Case I. In Figure 4.10, as $\sigma$ increases, the total cost decreases slightly and attain its minimum value at $\sigma = 8$, but the number of evalutations increases dramatically. A very larger value of $\sigma$ (e.g., 10) will need to evalutate more timing plans, but may not help much to minimize the objective. Similar trends can be found forother cases. Seting $\sigma = 5$, as we use in this paper, seems a good balance between solution quality and the computational effort of the algorithm.

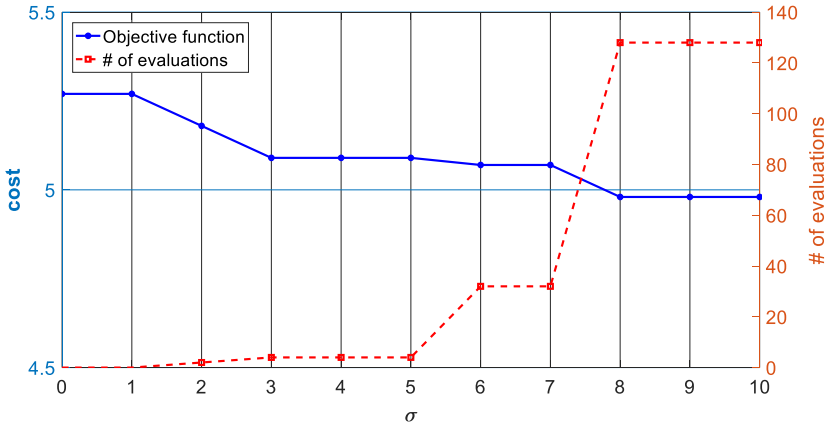



Figure 4.10. Influence of sigma on the total cost for Case I

4.3.2 *Multiple intersections on a corridor*

The proposed CV-based traffic signal optimization and coordination models are tested on a corridor that contains five signalized intersections. The distance between two adjacent

intersections is 800 meters (0.5mile). The West-East direction is the coordinated direction. Each intersection has a boundary of 400m upstream and 400m downstream of the intersection, as shown in Figure 4.11. It is assumed that upstream infrastructure-based detector data are available for each intersection to provide information on vehicle arrival. In this study, the vehicles are randomly generated at the boundary of the entire network with known arrival times, initial speeds and turning movements. At each intersection, 80% of the vehicle will use the through movement and others will turn left. We also assume that there is only one lane per incoming approach so that no lane changing behavior is modeled. The penetration rate for CV is 100% in this study. We evaluate the proposed signal timing optimization/coordination methods in three steps. First, we estimate the optimal signal timing parameters, including cycle length, phase duration, and offset in SYNCHRO, for different traffic demand levels. Second, for each given scenario that considers different combinations of traffic demands and vehicle types, we apply different methods to optimize signal timing plans (including the phase sequence and durations of each intersection and its offset). Third, we evaluate the performance of each signal timing plan generated from different methods by implementing the plans to the corridor and using IDM to generate vehicle trajectories, based on which to calculate the total cost of fuel consumption and travel time.

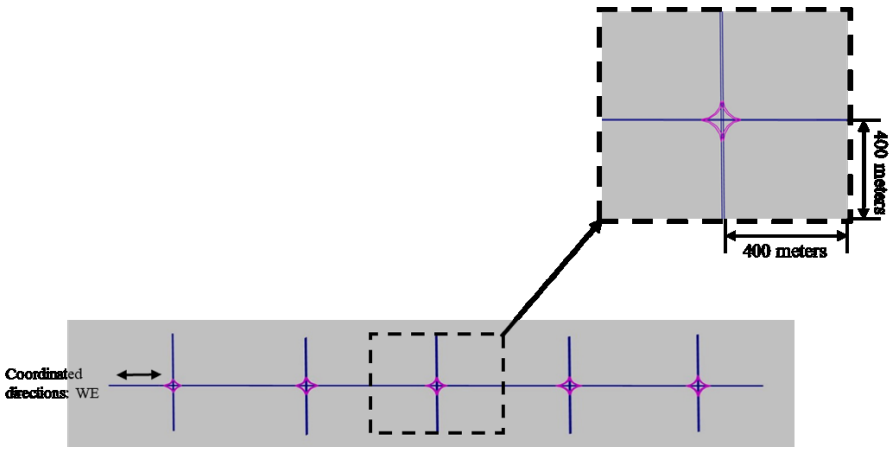



Figure 4.11. Simulation network containing five intersections

Six cases are tested in order to evaluate the proposed signal optimization methods considering different combinations of traffic demand levels and vehicle types. In Case I – Case III, vehicle demands for the main street (W - E direction) are 250 vph, 500 vph, and 800 vph. For minor street (N - S direction), the demands are set to be 125 vph, 250 vph, and 250 vph. All vehicles are assumed to be sedans. In Case IV – Case VI, vehicle demand levels are identical to Case I – Case III, but vehicle types are assigned differently. In the N-S directions, vehicles are assigned as Electric Vehicles (EVs), while in the main directions, vehicles are assigned as buses.

Table 4.4 shows the total costs estimated using different signal plans generated from three methods in 10 cycles. The first method is the actuated signal plan produced from SYNCHRO. Given the geometric information of the corridors and volumes on each movement, SYNCHRO can calculate the optimal signal parameters, including phase durations, cycle length, and offsets. The optimal signal plan, together with the information of randomly generated arrival vehicles and updated trajectories using IDM, are implemented in simulation network to estimate the objective function value, i.e., the total cost of fuel consumption and travel time using Eq. (4.36). The same procedures are applied to the other two methods to calculate the total cost. Noted that we consider a fixed cycle length constraint in this study in order to conduct signal coordination. The fixed cycle length for different demand levels for all methods are determined by SYNCHRO: they are 60s for low traffic demand (Case I and IV), 80s for medium traffic demand (Case II and V), and 120 for high demand (Case III and VI). The second method, MINLP in Eq. (4.36) - (4.55), is solved by the "NOMAD" solver in Matlab. There are eight phases for each signal, as shown in Figure 4.1. Considering we update the signal plan in every 10 cycles, there are 40 variables of phase durations and 4 variables of offsets for the simulation network containing five signals. The third method is the decentralized two-level model that can be solved by the proposed prediction-based approach.

At the intersection level, the phase durations for each intersection is solved by the DP method and can be updated every cycle. At the corridor level, the offsets are updated every 10 cycles.

In Table 4.4, the minimal total cost for each case is highlighted. For all cases, the results from MINLP and the two-level model are better than those from SYNCHRO. For Case III and IV, the results from the two-level model perform better than MINLP. Figure 4.12 shows the improvement of model performance for each case. For Case I and Case IV with relatively low demand levels, the improvements are smaller than other cases, which may suggest that coordination will bring limited benefits when traffic volume is low.

Table 4.4. Total cost from different methods

| **Method** | **Case I** NS: 125vph, WE: 250 vph; All Sedan | **Case II** NS: 250 vph, WE: 500 vph All Sedan | **Case III** NS: 250vph,WE: 800 vph All Sedan | **Case IV** NS: 125vph, WE: 250 vph; NS:Evs, WE:Bus | **Case V** NS: 250 vph, WE: 500 vph; NS:Evs, WE:Bus | **Case VI** NS: 250vph,WE: 800 vph; NS:Evs, WE:Bus |
|---|---|---|---|---|---|---|
| 1. SYNCHRO | 232.07 | 804.34 | 2047.8 | 320.44 | 1088.5 | 2778.5 |
| 2. MINLP | 227.44 | 737.53 | 1998.03 | 316.86 | 1036.23 | 2623.95 |
| 3. Two-level model | 229.5 | 764.42 | 1884.63 | 314.81 | 1065.7 | 2675.47 |

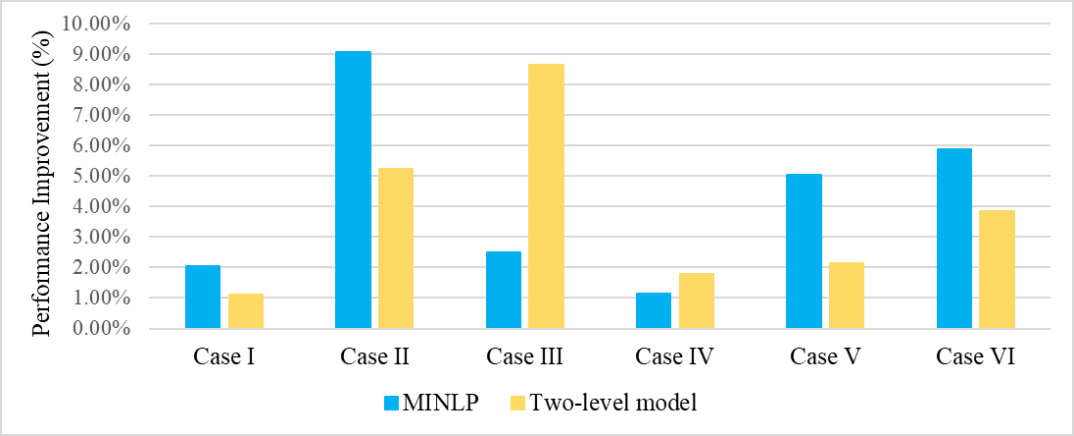



Figure 4.12. Improvement of model performance over SYNCHRO

To further evaluate whether coordination can benefit the road network, including both main street and minor street under different scenarios, we compare the improvement of model performance between two scenarios: consider coordination and without coordination (offset = 0). For example, in MINLP, we first solve the model that contains only 40 variables of phase durations

and all offset values are set to be zero. We then implement the estimated signal plan in IDM and estimate the total cost for the main street and the minor street separately. These costs are compared with the results from solving the entire model, i.e., Eq. (4.36) - (4.55), which optimizes both phase durations and offsets. Same procedures apply to the two-level model. Table 4.5 shows the comparison results. The negative values are highlighted in the table, which suggests that by applying coordination, the performance for the main street or the minor street gets worse. For the main street (left number in each cell), MINLP and the two-level model are both underperformed under low demand levels (Case I or IV) if applying coordination, while the improvements are more significant for higher demand levels (Cases II, III, V, VI). The results suggest that coordination schemes may not be beneficial to a corridor with low traffic volumes and random arrival vehicles because those vehicles are less likely to form a platoon to be influenced significantly by the operation of adjacent intersections. For the minor street (right number in each cell), the improvements are relatively small no matter if they are positive or negative. Coordination on the main street seems to have little impact on vehicles on the minor street.

Table 4.5. Model performance improvement from coordination

| **Method** | **Case I** | **Case II** | **Case III** | **Case IV** | **Case V** | **Case VI** |
|---|---|---|---|---|---|---|
| | NS: 125vph, WE: 250 vph; All Sedan | NS: 250 vph, WE: 500 vph All Sedan | NS: 250vph,WE: 800 vph All Sedan | NS: 125vph, WE: 250 vph; NS:Evs, WE:Bus | NS: 250 vph, WE: 500 vph; NS:Evs, WE:Bus | NS: 250vph,WE: 800 vph; NS:Evs, WE:Bus |
| | main street \| minor street | | | | | |
| MINLP | **-2.5%** \| **-1.5%** | 9.1% \| **-0.6%** | 4.5% \| 1.2% | **-2.7%** \| **-2.9%** | 4.6% \| 1.6% | 5.9% \| 0.3% |
| Two-level model | **-1.1%** \| 0.5% | 8.7% \| 0.7% | 3.7% \| **-0.7%** | 0.5% \| 0.6% | 7.8% \| **-0.8%** | 3.8% \| 0.2% |

The previous six cases test the influences of various combinations of demand levels and vehicle types on signal coordination. It is still worthwhile to test how the proposed methods performed under various traffic demands on opposite directions or different movements (turning and through movement). Hence, three more cases are tested. First of all, the traffic demands for right-turn movements are added to the simulation network. The vehicles traveling on left-turn and

right-turn movements are randomly sampled between 10%-20% of the total demands for each approach in case VII-IX. In addition, traffic demands of the opposite directions are not the same, e.g., for the major street, volumes on E→W and W→E directions are different. For cases VII and VIII, the traffic demand of each direction is given value as shown in Table 4.6. In case IX, traffic demand in each direction is randomly sampled within a certain range. For example, in the E→W direction, initial demand is randomly (uniformly) selected from 250 vph to 500 vph. In all cases, the performance of MINLP and the two-level model are all better than the baseline SYNCHRO model. The model improvements are shown in Figure 4.13.

Table 4.6.Total cost under various demand levels of opposite directions

| **Method** | **Case VII** | **Case VIII** | **Case IX** |
|---|---|---|---|
| | W-->E: 250 vph;<br>E-->W: 500 vph;<br>N-->S: 100 vph;<br>S-->W: 150 vph. | W-->E: 500 vph;<br>E-->W: 250 vph;<br>N-->S: 150 vph;<br>S-->W: 100 vph. | E-->W: 250-500 vph;<br>W-->E: 250-500 vph;<br>N-->S: 100-150 vph;<br>S-->W: 100-150 vph. |
| 1. SYNCHRO | 316.57 | 318.32 | 349.56 |
| 2. MINLP | 307.91 | 305.43 | 323.73 |
| 3. Two-level model | 304.75 | 306.9 | 330.21 |

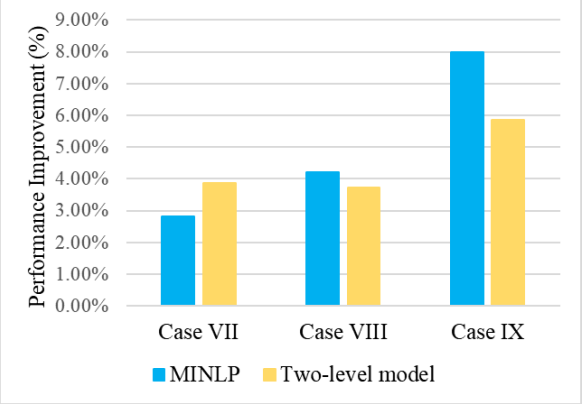
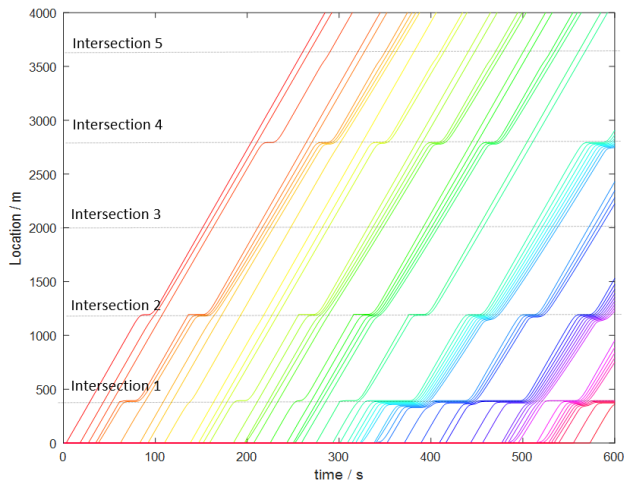


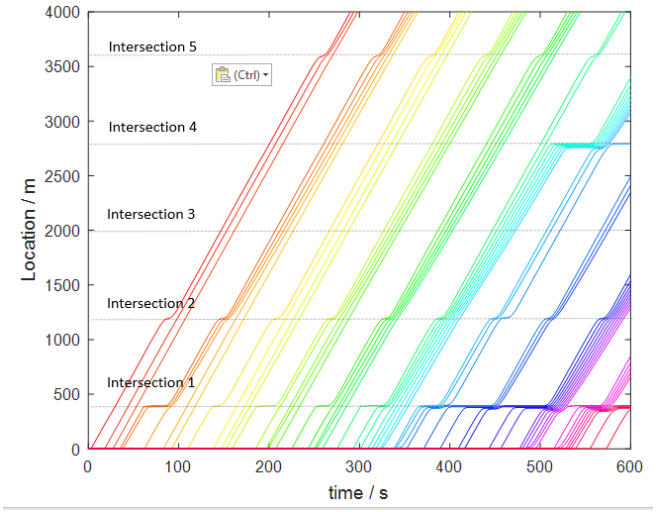

Figure 4.13. Improvement of model performance over SYNCHRO for cases VII-IX

Figure 4.14 shows vehicle trajectories updated based on different signal plans for 10 cycles (600s) along a 4000m corridor. The vehicles in the figure are on the coordinated movement (W→E direction in Figure 4.11) and randomly generated at the boundary of the corridor. Compared with the trajectories generated using Synchro plan in Figure 4.14(a), the delays and number of stops in other

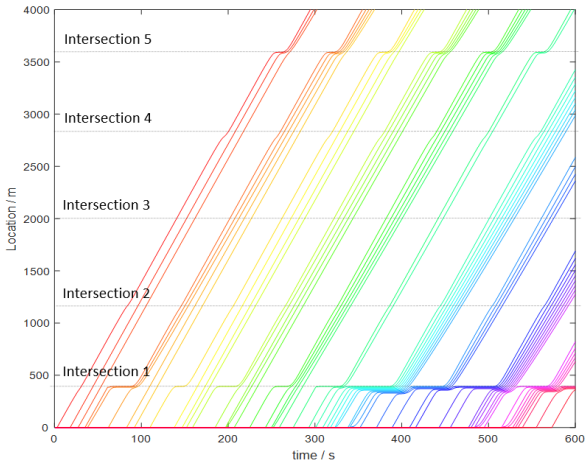
two figures (Figure 4.14(b) and (c)) are significantly reduced by applying the MINLP method and the two-level model. Through the optimization of signal plans, the randomly generated vehicles form vehicle platoons to pass the intersections smoothly.




(a) Trajectories from SYNCHRO signal plan




(b) Trajectories from MINLP signal plan




(c) Trajectories from the two-level model signal plan

Figure 4.14. Vehicle trajectories from different signal plans

The numerical experiments test different methods in 10 cycles ranging from 10 min to 20 min based on the cycle length. The simulation period can be extended to a longer time but requiring a significant longer computation time. It is influenced by several factors, e.g., update intervals of signal plans, whether considering various vehicle types and the level of traffic demand in the

network. For the NOMAD solver, it may fail to find the feasible solutions under high vehicle demand levels and short signal plan updating intervals (meaning a larger number of variables). The two-level model can generally ensure convergence, but the computation time also varies based on the aforementioned factors. Figure 4.15 shows the number of iterations for different offset updating intervals, for Case I. Usually, the program will converge within 10 iterations for different cases. Similar patterns are found for other cases, which are omitted here.

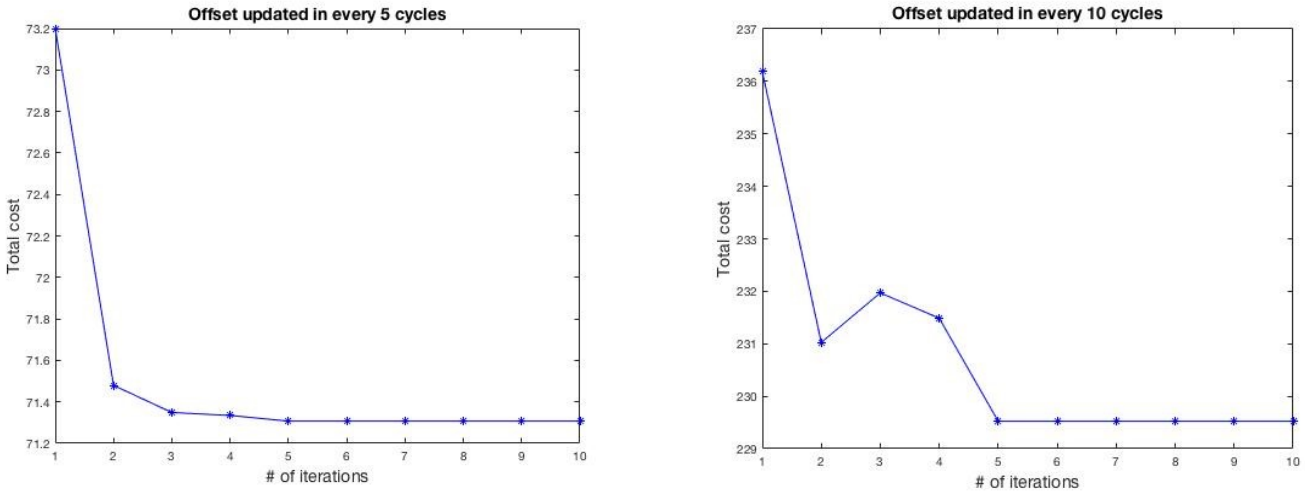



Figure 4.15. Optimization results of the two-level model for case 1

## 4.4 SUMMARY

This chapter illustrated the modeling framework and numerical experiments on the signal timing optimization problem under full penetration rates of CVs. First, a signal timing optimization model for a single intersection was presented with a fixed cycle length. The algorithm utilized arrival information (speeds, locations, etc.) from CV as the input to optimize the green times by considering individual vehicles' fuel consumption and travel time. The problem was first formulated as a MINLP by applying the IDM to predict vehicle trajectories. Such a formulation has a large dimension and a complex car-following model (the IDM). A DP formulation was then developed to approximate the MINLP. The overall problem was divided into stages (one stage for

each signal phase). The objective is the summation of the objective of each stage. The objective function of a stage was approximated as a function of the state and decision variables of the stage only, by approximating the vehicle speeds and delays. Imposing the fixed cycle length constraint would invalidate the DP formulation. Then a two-step method was applied to address this issue. In the first step, an end-stage cost was added to the DP formulation, defined by how much the DP solution violates the fixed cycle length constraint. This step forced the DP to produce a solution with a cycle length that is close to the given fixed cycle length. The second step was a branch and bound method to further refine the DP results to obtain a solution that produces the given cycle length exactly. The performance of the algorithm was tested using data generated from traffic simulation. The proposed DP model was compared with two other models. The first one was the traditional actuated signal timing plan generated by SYNCHRO. The second was to solve the MINLP formulation directly using the NOMAD solver in MATLAB. The results showed that the proposed DP method is always superior to SYNCHRO under all cases and can generate similar (slightly worse) solutions compared with NOMAD. However, NOMAD had difficulties finding optimal solutions when the number of variables is relatively large, and the computational times of NOMAD were much larger than DP. This makes the proposed DP method more favorable when dealing with large scale problems.

Second, a signal timing optimization and coordination framework were presented for a signalized corridor under the CV environment considering individual vehicle's trajectories. The problem was formulated in a centralized scheme as a MINLP considering the fixed cycle length constraint to optimize the phase durations and offsets in one mathematical program. Due to the complexity of the model, we decentralized the problem into two levels: an intersection level to generate optimal phase durations using the DP method the authors developed previously for each

intersection and a corridor level to update the optimal offsets for all the intersections. The two-level model reduced the complexity of the MINLP. In order to solve the two-level model, a prediction-based solution technique was further developed that can solve the problem iteratively. We tested the proposed models and solution technique in a corridor containing five intersections through traffic simulation. The results from MINLP and the two-level model both outperformed the signal optimization and coordination plan generated by SYNCHRO. This was tested for nine cases that consider various combinations of traffic volumes and vehicle types. The results also suggested that signal coordination may bring limited benefits to intersections with low traffic volumes or to the vehicles on the minor street.

# Chapter 5. TRAFFIC SIGNAL CONTROL UNDER LOW PENETRATION RATE OF CVS

It is generally agreed that CV technology has the potential to transform urban traffic modeling and control practices. However, the penetration rate of CVs is the key factor that largely influences the ability to deploy CV technologies for traffic signal control. With the emergence of new technologies and systems in transportation, such as new sharing mobility services (e.g., Didi or Uber), accessibility of more data sources, e.g., mobile sensing-based data, offers an opportunity to significantly enhance traffic signal operations. Because currently CV data are limited and we do not have public access to those data, mobile sensing data are used in this chapter to illustrate how the proposed models work for volume estimation and prediction, as well as traffic signal control using the predicted volumes, when the penetration rate of mobile sensing data is low.

Figure 5.1 shows the overall research framework of traffic signal control method under a low penetration rate of mobile sensing data. Here, the low penetration rate refers to 2%-5% penetration rate of mobile sensing data, as shown in Table 3.1. Trajectory-based traffic signal control method can hardly apply to this scenario because it is really challenging to estimate trajectories for unequipped vehicles given sampled trajectory of CVs. So, the aggregated performance measurements, e.g., volumes or queue lengths can be estimated and predicted for signal timing optimization. Two cases are discussed in this chapter. If only loop detector/camera data are available, traffic volumes of each movement for a given intersection can be obtained directly. A deep learning approach is developed to predict short-term traffic volume, which is further utilized for traffic signal timing optimization. If the loop detector/camera data are not available, the partial least regression (PLS) method is applied to estimate traffic volume first given sampled vehicle trajectory data. Then traffic volumes are predicted based on the estimated volumes using the

proposed deep learning model. The predicted volumes are used for fixed time signal control to optimize green splits.

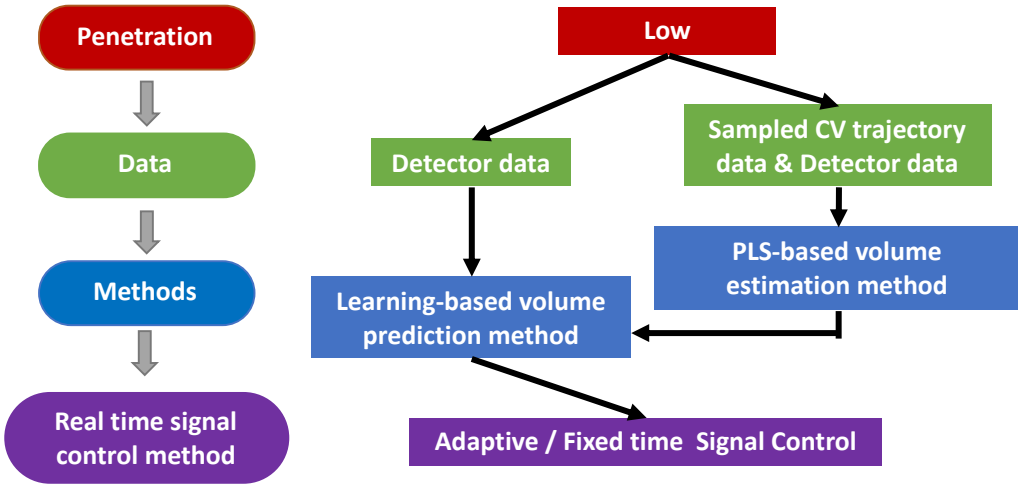



Figure 5.1. Traffic signal control framework under the low penetration rate of mobile sensing data

## 5.1 Traffic Volume Estimation Using PLS

At signalized intersections, traffic volume is considered as an important indicator for traffic signal control systems. Traffic volume data are always collected from loop detectors or camera sensors. However, not all intersections are equipped with sensors/detectors. Moreover, detector failures are common problems due to pavement cracking or increment weather. Hence, it becomes inevitable to use sampled CV data/mobile sensing data to estimate traffic volume for each movement at signalized intersections.

Traffic data under low penetration rate contains multiple issues. First, they only present partial observable traffic information, which is incomplete and bias. Second, the features extracted from traffic data are highly correlated in the spatial and temporal domain. For example, speed profiles are similar for two closely located road segments. They are also almost identical in a few continuous time intervals. The missing data and redundant information make volume estimation

task very challenging to solve. This study develops a traffic volume estimation method based on PLS. PLS is capable of extracting the dominant and relevant information from noisy data sets and reducing the complex data set to a lower dimension for effective and efficient estimation problem. The objective of traffic volume estimation is to utilize sampled mobile sensing data/CV data, e.g., sampled taxi GPS data, to estimate traffic volume of each movement for signalized intersections is in real time. Taxi trajectory data provide many features that can represent the traffic conditions for a certain road segment, e.g., the number of sampled vehicles/trajectories, number of stops, and average speed on each movement. These features are considered as predictor variables in PLS. They are used to regress the ground-truth value of traffic volume by accounting for day-to-day variations and spatiotemporal dependencies of traffic data.

### 5.1.1 *Problem formulation*

We denote traffic features, e.g., # of vehicle trajectories, average speed, and number of stops, extracted from sampled mobile sensing data/CV data at day *d* as a vector $x_m^d \in \mathbb{R}^T$:

$$x_m^d = [x_m^d(1) \quad x_m^d(2) \quad \dots \quad x_m^d(T)]^T, \quad d = 1, \dots, D, m = 1, \dots, M \tag{5.1}$$

where $m$ denotes the location that traffic states are measured and $T$ denotes the total time intervals. The $[\cdot]^T$ denotes vector transpose.

For a certain day *d*, traffic features can be represented as a vector $x^d \in \mathbb{R}^{TM}$:

$$x^d = [x_1^d \ x_2^d \quad \dots \quad x_M^d]^T, \qquad d = 1, \dots, D \tag{5.2}$$

The aggregate Traffic Feature Matrix (TFM) is shown in Eq. (5.3) where each row represents traffic features at time $t = 1, \dots, T$ on location $m = 1, \dots, M$ on a certain day *d*. Each column represents the traffic features at a certain time $t$ on a certain location $m$ on different days $d = 1, \dots, D$.

$$X_D^{T,M} = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \dots \\ (x^D)^T \end{bmatrix} \in \mathbb{R}^{D\times(TM)} \tag{5.3}$$

Denote $T_p$ ($T_p < T$) the current time. The traffic volume estimation problem aims to learn a function F(·) that maps historical traffic features before $T_p$ ($T_p < T$) to real traffic volume $V_{\bar{d}}^{T_p,M}$ at $T_p$ on day $\bar{d}$ accounting for day-to-day variations of traffic states learned from previous *D* days.

$$\mathrm{F}\left(X_D^{T_p,M},\right) = V_{\bar{d}}^{T_p,M} \tag{5.4}$$

### 5.1.2 *PLS modeling*

PLS regression model is capable of conducting real-time traffic volume estimation based on sampled CV/mobile sensing data. The PLS algorithm constructs components as a linear combination of the original predictor variables while considering the correlation between predictors and response variables. It is noted that traffic feature before $T_p$ and after $T_p$ are highly correlated due to the temporal dependencies of traffic flow. In practice, it is often more efficient to use the traffic features from the recent past of $T_p$ (e.g., 2 hours) instead of all the estimates from the starting of the day. In the following, the PLS method I described in general which can be readily modified to restrict the past data or the estimation horizon to any pre-defined time window; see the numerical section 5.4.1 for more details.

In PLS, the objective is to a collection of predictor components $p^1, p^2, \dots p^N$ with each $p^i \in \mathbb{R}^{T_p M}$; predicted components $q^1, q^2, \dots, q^N$ with each $q^i \in \mathbb{R}^{(T-T_p)M}$; and a vector of weights $\omega(d) \in \mathbb{R}^N$ with

$$\omega(d) = [w_1(d) \quad w_2(d) \quad \dots \quad w_N(d)]^T \tag{5.5}$$

such that the traffic states before $T_p$, and after $T_p$, can be express as:

$$x_b = \overline{x}_b + \sum_{i=1}^{N} w_i(d)\, p^i \tag{5.6}$$

$$x_a = \overline{x}_a + \sum_{i=1}^{N} w_i(d)\, q^i \tag{5.7}$$

where $\overline{x}_b$ and $\overline{x}_a$ are the average traffic states before and after $T_p$.

After the predictor and predicted component estimated from PLS using historical sampled CV/mobile sensing data, the ground truth traffic volumes are estimated for a sample day at $T_p$ using the calculated historical predictor component $p^i$ and predicted component $q^i$. Denote $x_b^{\bar{d}} \in \mathbb{R}^{T_p M}$ denotes the traffic features before $T_p$ on the sample day $\bar{d}$. The weights are estimated as:

$$\widehat{w} = \left(\left(x_b^{\bar{d}} - \overline{x_b}\right)^T (P^T)^{\dagger}\right)^T \tag{5.8}$$

where $P$ is the vector of predicted component and $(P^T)^{\dagger}$ is the Moore-Penrose pseudoinverse of $P^T$. Then the predicted ground truth value of travel time after $T_p$ can be estimated:

$$x_a^{\bar{d}} = \widehat{w}^T Q^T + \overline{tt}_a, \;\; x_a^{\bar{d}} \in \mathbb{R}^{(T-T_p)M} \tag{5.9}$$

where $Q^T$ is the vector of predicted components. $P^T$ and $Q^T$ are estimated from PLS.

The PLS prediction algorithm is present:

---

Algorithm to predict traffic states using PLS regression

---

Set current time interval $t$ (e.g., at 10:00AM)
**for** time step *t* between *0* to *T* in every $\Delta t$ min:
    Prepare training and validation sets for the predictor variable ($X_D^{T,M}$)and response variable Y($X_{\bar{d}}^{T_p,M}$) respectively.
    **for** # of PLS component *ncomp* from 1 to *Ncomp* (e.g.,40):
        Run PLS regression on X_train and Y_train.
        Test the model using X_validate and Y_validate.
        Record the mean square error of validation.
    **end**
    Find the optimal *ncomp** leading to the minimum mean square error.
    Estimate traffic volume using PLS regression with optimal *ncomp**
**end**

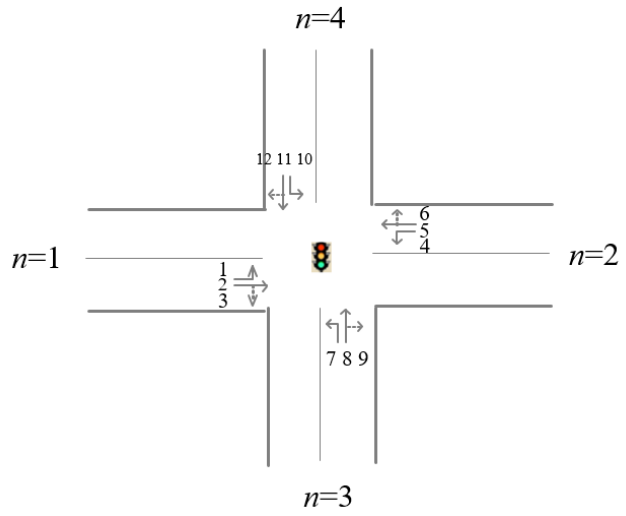
## 5.2 Traffic Volume Prediction using Convolutional LSTM Network

This section presents a Deep Intersection Spatial-Temporal Network (DISTN) learning approach for real-time movement-based traffic volume prediction at signalized intersections given either loop detector data (considered as ground truth traffic volume data) or the estimated traffic volume based on sampled CV/mobile sensing GPS data, which considers both spatial and temporal features by the CNN and LSTM, respectively. Real-time traffic information is collected through a loop detector, i.e., traffic volume for each movement in every 15 min, which is the input to the DISTN to predict future traffic volume. In particular, CNN is applied first to learn the spatial dependencies. Movement-based traffic volume data for each intersection are transferred to a 2D image. Each image is an origin-destination matrix representing the volume from an incoming approach to an outgoing approach of an intersection. Then multiple images from nearby intersections are stacked together as the input to CNN. This is inspired by the intuition that traffic volumes of some movements of the same intersection are correlated, which may also be correlated with volumes at upstream and downstream intersections. The LSTM then takes the output from the CNN at each time step as input to learn the temporal dependencies. Traffic volumes have temporal correlations due to the dynamic nature of transportation systems, which may present within a day, daily and weekly periodic patterns. Each of those complex nonlinear relationships can be captured by constructing a separate pair of CNN and LSTM. CNN and LSTM have been widely investigated and applied in traffic flow predictions. However, most applications are conducted for traffic flow prediction on freeways. Compared with freeways, traffic flow at intersections has large fluctuations and uncertainties. This section starts by describing the requirements of data type and data range for the proposed model.

### 5.2.1 *Data Description*

Historical traffic counts collected from loop detectors or estimated traffic volume from sampled mobile sensing data in each movement can be used in DISTN directly. Camera detector can provide similar traffic volume data by identifying the number of vehicles in each time step (e.g., 5 min). In addition, the proposed model is developed to predict traffic volumes of all movements simultaneously for multiple intersections in a corridor or network by accounting for the spatial and temporal correlations of traffic volumes. Hence, movement-based traffic volumes for multiple adjacent intersections are required, e.g., three intersections in a corridor. In the numerical experiment section, the influences of network size and network structure are also tested using real-world data and in simulation.

For an intersection, traffic volume data are measured at its boundaries, e.g., 300m upstream of the intersection. It is assumed that there are three movements for each approach, i.e., the left-turn movement, through movement, and right-turn movement, resulting in 12 movements per intersection. Figure 5.2 defines how these 12 movements are numbered. For example, if we measure the volume on movement #1 at time $t$, $x_t^1$, we count the number of vehicles coming from approach 1 ($n$ = 1 in Figure 5.2) and heading to approach 4 ($n$ = 4) by following the left-turn movement using movement 1. Notice that this study does not distinguish different lanes for the same movement. Also if an intersection is not 4-leg (e.g., 3-leg or 5-leg), its movements (9 or 15 movements respectively) can be numbered in a similar way.




| Origin \ Destination | n=1 | n=2 | n=3 | n=4 |
|---|---|---|---|---|
| n=1 | Σ | $x_t^2$ | $x_t^3$ | $x_t^1$ |
| n=2 | $x_t^5$ | Σ | $x_t^4$ | $x_t^6$ |
| n=3 | $x_t^7$ | $x_t^9$ | Σ | $x_t^8$ |
| n=4 | $x_t^{12}$ | $x_t^{10}$ | $x_t^{11}$ | Σ |

Figure 5.2. Traffic volume data are represented by an OD matrix.

The volume prediction problem aims at predicting the volumes for all movements of target intersections in a corridor or network at time step $t + 1$ given the historical data until time step $t$. Assume there are $I$ target intersections consisting of $12 \times I$ movements totally. The historical volume data available at time $t$ can thus be characterized as a matrix:

$$X_t = \begin{bmatrix} x_{t-H}^1 & x_{t-H+1}^1 & \cdots & x_t^1 \\ x_{t-H}^2 & x_{t-H+1}^2 & \cdots & x_t^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{t-H}^{12\times I} & x_{t-H+1}^{12\times I} & \cdots & x_t^{12\times I} \end{bmatrix} \tag{5.10}$$

Here for each element in the matrix, the superscript indicates the movement (see Figure 5.2), while the subscript indicates the historical time step. Each column of the matrix represents the traffic volumes at a certain time step for all $12 * I$ movements of $I$ target intersections. Each row of the matrix represents the traffic volumes for a certain movement from time step $t - H$ to time step $t$, with $H$ the number of historical time steps (e.g., if 2-hour historical data are considered and the time step is 15 min, $H$ will be 8). In other words, starting from the first row, every 12 rows of the matrix consist of the historical volume data for one intersection.

Based on the way how the movements are numbered in Figure 5.2, the traffic volume data in Eq. (5.10) can be first transferred into a number of matrices, one for each intersection at a given time step, which can be readily used as the input to CNN. As shown in Figure 5.2, at each time *t*, we consider the 12 movements of an intersection (12 consecutive rows of the column in the matrix in equation (5.10)) that corresponds to time *t* as a $4 \times 4$ image. For example, as shown in Figure 5.2, the volume $x_t^1$ is inserted into row 1 and column 4 in the right matrix in Figure 5.2 since it represents the left turn volume from approach 1 to approach 4. The traffic volume at the diagonal of the matrix is the summation of the volumes of each row. As a result, movement-based traffic volumes at an intersection for any given time step can be represented as an origin-destination matrix of that time step. As we will see later, data organized this way to help explore and take advantage of the structure of an intersection. At the same time, interpreting the matrix as an image allows us to apply CNN-type image processing methods to model/capture the correlations among all volumes of the same intersection and downstream and upstream intersections. This is an important feature and one of the key contributions of the volume prediction model (DISTN) proposed in this paper.

### 5.2.2 *Spatial features extracted from CNN*

CNN can help exploit the fundamental spatial features of image or videos. In transportation studies, CNN is also considered as an effective model to extract spatial features of traffic volume data. As mentioned earlier, upstream and downstream intersections may influence the traffic volume prediction for a target intersection. For the same intersection, traffic volumes from nearby movements may also correlate with each other. This section explains how these unique spatial features of movement-based traffic volumes can be captured by CNN.

As shown in Figure 5.2, a 2D ($4 \times 4$) image can represent the traffic volume of a single intersection at each time step *t*. For multiple consecutive intersections in a corridor or network, the $4 \times 4$ image of each intersection at the same time step needs to be stacked together, as shown in Figure 5.3. As a result, the 2D input data at time step *t*, $Y_t$, is a matrix with dimension $4I_x \times 4I_y$, where $I_x$ and $I_y$ indicates the number of intersections in the W-E direction and N-S direction respectively. We denote $Y_t^k$ as the input to the $(k+1)^{th}$convolutional layer which is also the output of the $k^{th}$ convolutional layer ($k = 1, 2 \ldots K$). When $k = 0$, $Y_t^0 = Y_t$ (i.e., the input data). Thus, $Y_t^k$ at time step *t* is characterized as:

$$Y_t^k = \sigma_k(Y_t^{k-1} * W_t^k + b_t^k) \tag{5.11}$$

where $\sigma_k$is the activation function for layer *k*, * denotes the convolutional operation, and $W_t^k$ and $b_t^k$ are the weight and bias parameters in the $k^{th}$ convolution layer. The outputs of each convolutional layer *k* use the same parameters $W_t^k$ and $b_t^k$ in many different positions in the inputs $X_t^{k-1}$ in order to detect spatial features. As a result, the number of parameters in the CNN is largely reduced compared with the fully connected neural network. The output in the $K^{th}$ convolution layer, $Y_t^K$ is flatten to a feature vector $y_t$ at time *t* and will be the input into the LSTM network that is used in the next section to capture temporal correlations.

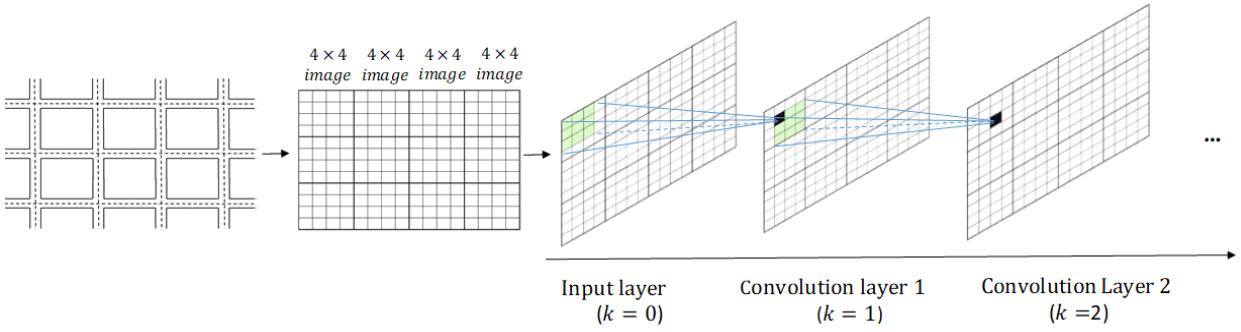



Figure 5.3. Spatial features from multiple intersections are extracted from multi-layer CNN

With one convolution layer, the local spatial dependencies of traffic volumes, e.g., movements in the same intersection, can be captured depending on the filter size of convolution. However, with more convolutional layers stacked together, the spatial dependencies between distant intersections can be learned as well [93]. As shown in Figure 5.3, the input layer to the CNN is a road network matrix generated by stacking multiple $4 \times 4$ matrices (in Figure 5.2) of all target intersections together, based on their geometric locations. If the filter size is $3\times3$, the traffic volume of certain movements, e.g., the value of the highlighted cell in the Convolution layer 1, only depends on the traffic volumes of the 9 movements near it in the input layer. However, at a higher layer, the traffic volume of a certain movement (the highlighted cell in the Convolution layer 2) can learn informative information from more nearby movements (i.e., surrounding cells) in the input layer. Thus, the predicted traffic volume for a certain movement at a higher convolution layer depends on a few movements at the same or nearby intersections in the middle convolution layers, which is further correlated with more movements at distant intersections in lower convolutional layers. In this way, the DISTN model is effective to extract spatial features for movement-based traffic volumes at a corridor or a network that contains multiple intersections.

### 5.2.3 *Temporal features extracted from LSTM*

Traffic volume data are characterized by both short term and long-term temporal dependencies. In a discrete-time system where time is discretized into time steps (say every 5 or 15 minutes), “short term” here means a few time steps, while “long term” means ten or more time steps. First, traffic volumes exhibit high correlations in the short term due to the dynamic natures of traffic flow. For example, traffic volumes at two consecutive time steps, e.g., every 15 min, are highly correlated. Moreover, the long term dependencies of traffic volumes also exist, e.g., traffic volumes between

multiple time steps (e.g., a few hours) can be correlated. An example of this dependency is when a football game is over, heavy traffic exiting from the arena needs a long time to travel even for a few miles. Hence, these short-term and long-term dependencies should be both considered when extracting temporal features from traffic volumes. Recently, the LSTM network has been widely used for time series prediction problems. The advantages of LSTM is that it can capture both short term and long term correlations of time series data using gating mechanisms, which can also avoid gradient vanishing and exploding problems [94]. This section introduces how to model the underlying short term and long-term temporal correlations of traffic volume data using LSTM.

Recurrent Neural Network (RNN) is a powerful tool to handle the time series prediction problems [99] by considering temporal correlations. LSTM is one of the RNNs, which was firstly introduced by Hochreiter and Schmiduber [94]. It was developed to address the vanishing and exploding gradients issue of the classic RNN and capture the long-term dependencies. Standard RNNs take their previous hidden states $h_{t-1}$ as the current input and generate a new hidden state $h_t$ as the output, as shown in Figure 5.4. LSTM does the same except that a memory cell $c_t$ is added. It provides an effective way to model the long-term dependencies by using the memory cell $c_t$ to pass relevant information along with the LSTM network. The standard LSTM is applied in this study to account for the temporal features of movement-based traffic volume data because previous research showed that none of the variants of LSTM can significantly improve the performance of the standard LSTM architecture [72].

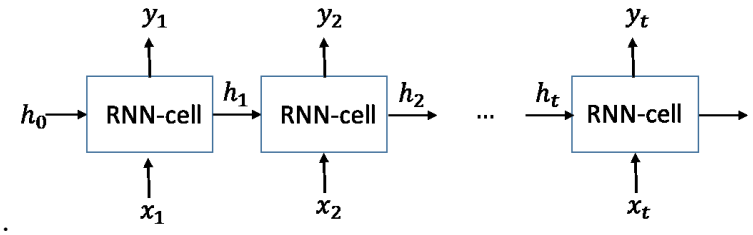



Figure 5.4. Basic RNN Mode

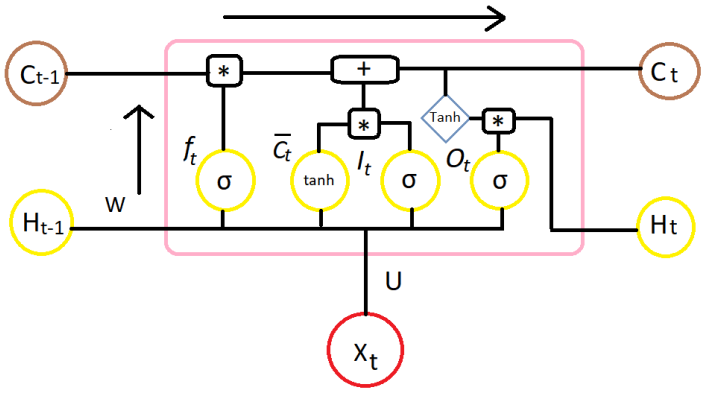



Figure 5.5. LSTM cell [100]

The key to the LSTM is the memory cell state $c_t$ which is updated along the horizontal line at the top of Figure 5.5. It contains either partial or all necessary information for traffic volume prediction at the next time step, which could be short period or long-term dependencies of traffic volumes. LSTM uses three gates: the forget gate, the update gate, and the output gate to control the memory cell state $c_t$ by either removing or adding information to it. At each time step *t*, the forget gate, the input gate, and the output gate are defined as $f_t$, $i_t$, and $o_t$ respectively. Eq. (5.12) - (5.14) shows the functions to estimate the three gates:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5.12}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5.13}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5.14}$$

where $W_f, U_f, b_f, W_i, U_i, b_i, W_o, U_o, b_o$ are parameters that need to be learned. Parameters $W_a, a \in \{f, i, o\}$ and $U_a, a \in \{f, i, o\}$ control the behaviors of the three gates corresponding to the current input cell state $x_t$ and the previous hidden cell state $h_{t-1}$. The function $\sigma$ is the sigmoid function. The three gates $f_t, i_t$, and $o_t$ are vectors of values between 0 and 1. They can be considered as the variables to determine which piece of information can be passed from $c_{t-1}$ to $c_t$.

Eq. (5.15) is used to estimate the candidate cell input state $\bar{c}_t$. The forget gate $f_t$ will be multiplied element-wise by the previous memory cell $c_{t-1}$ in Eq. (5.16) to determine whether the previous memory cell $c_{t-1}$ contains revelant information and will be kept. For example, if the value of $f_t$ at time step *t* is 0 (or close to 0), then it indicates that the LSTM should remove that piece of information in the corresponding component in $c_{t-1}$. If the value is 1, the information will be kept. In a similar way, the input gate $i_t$ is multiplied element-wise by the current candidate cell input state $\tilde{c}_t$ to determine how much the current cell state $c_t$ need to be updated. Eq. (5.17) illustrate what values from the updated cell state $c_t$ will be added to the hidden state output $h_t$.

$$\bar{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5.15}$$

$$c_t = f_t * c_{t-1} + i_t * \bar{c}_t \tag{5.16}$$

$$h_t = o_t tanh(c_t) \tag{5.17}$$

Here $tanh$ is a hyperbolic function.

To apply LSTM for volume prediction, the input to LSTM is a sequence of traffic volumes over time denoted as $S_t = (y_{t-H}, y_{t-H+1}, \dots, y_t)$ where $y_t$ is a spatial feature vector of the CNN output after *K* convolutional layers at time *t* and *H* is the historical time steps. The network is trained to predict the movement-based traffic volumes for all target intersections at next time step $t + 1$. In Figure 5.6, each LSTM cell includes several gates operations, i.e., forget gate, input gate, and output gate. The three gates determine how much the past output (predicted traffic volumes at previous time intervals, denoted as, $h_{t-H}$, $h_{t-H+1}, \dots$) to retain, how much of the current input (the spatial feature extracted from CNN, e.g., $y_{t-H}$, $y_{t-H+1}$, …) to keep, and how much of the long-term dependencies of traffic volume data (memory cell $C_{t-H}$, $C_{t-H+1}$,… ) to output, which makes the LSTM more powerful compared to other time series models to learn temporal correlations of traffic volumes. It is suggested that multilayer LSTM networks can

generate a higher level of representations of sequence data and work more effective to learn long-term dependencies [95]. The structure in Figure 5.6, therefore, stacks multiple LSTM hidden layers to improve prediction performance. Details are omitted here.

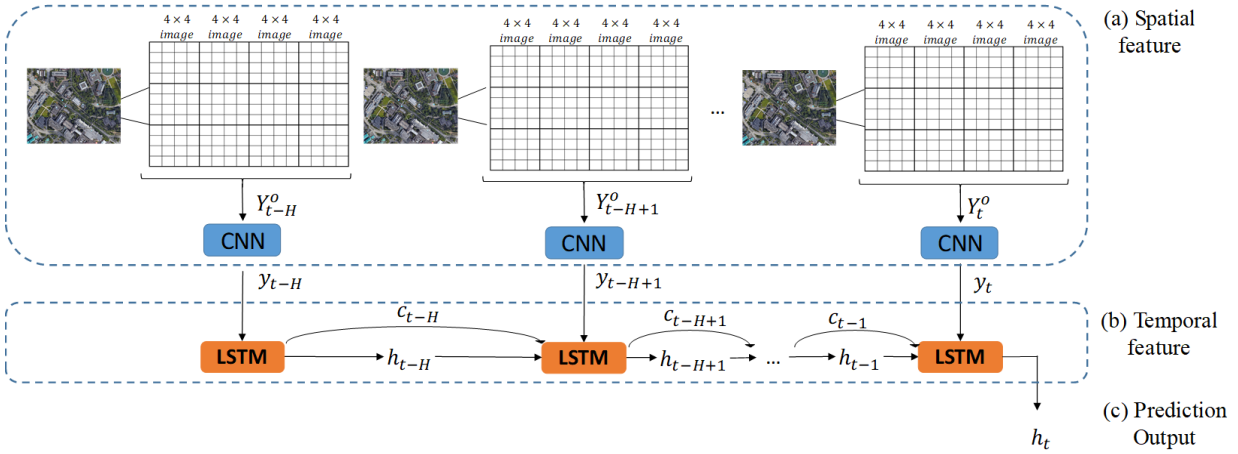



Figure 5.6. The representation of spatial and temporal features extracted from CNN and LSTM

### 5.2.4 *The DISTN model*

In addition to the short-term and long-term temporal correlations within the same day among traffic volumes as captured by the LSTM model above, there are also periodic patterns, e.g., daily and weekly patterns, in the volume data. This means the traffic volume at the current time step on day $d$ can be highly correlated with volumes of the previous time intervals on the previous day $d-1$ and on the same day of the previous week $d-7$. The proposed DISTN model is developed to extract spatial and temporal features of traffic volume data of within-day (i.e., both short term and long-term as captured by the LSTM in the above section), daily periodicity (on the previous day), and weekly periodicity (same day of the previous week), respectively. Recall that the goal is to predict the traffic volume at $t+1$ on day $d$ given the historical sequence until $t$. This leads to the final structure of the DISTN model as shown in Figure 5.7. In the figure, $S_t^d$, $S_t^{d-1}$, $S_t^{d-7}$ are

the output sets from the CNN which contain spatial features of traffic volume during historical time intervals for a near term on the same day $d$, the previous day $d-1$, and the same day of the last week $d-7$, respectively, which are also the input to each of the LSTMs. The outputs of the three LSTMs that contain both spatial and temporal features of traffic volumes are concatenated together:

$$q_t^d = h_t^d \oplus h_t^{d-1} \oplus h_t^{d-7} \tag{5.18}$$

where $h_t^d$, $h_t^{d-1}$, $h_t^{d-7}$ are the outputs of the LSTM of the three periodic sequences, $\oplus$ denotes the concatenation operator, which is a commonly used operation to join a sequence of data along an existing axis [96]. All the concatenated features at time $t$ on day $d$, $q_t^d$, are the input to the fully connected (FC) layer to predict future traffic volume for all movements. The Rectified Linear Unit (ReLU) ($f(z) = max(0, z)$) is used as the activation of the FC layer because it ensures the non-negativity of traffic volumes and easier to converge. The final output of the DISTN model is the predicted traffic volumes at time $t+1$ on day $d$, for all movements $\hat{v}_{t+1}^d$.

To train the DISTN model, a loss function has to be used, which can be defined as:

$$\mathcal{L}(\theta) = Loss(\hat{v}_{t+1}^d - Y_{t+1}^d) \tag{5.19}$$

where $\theta$ are all learnable parameters in the DISTN model, $Loss$ is a function to calculate the error between the predicted traffic volume $\hat{v}_{t+1}^d$ and the ground-truth value of traffic volume $Y_{t+1}^d$ at $t+1$ on day $d$ from the training dataset. The $Loss$ function can be a Mean Squared Error (MSE) loss or Mean Absolute Error (MAE) loss for contiguous-value prediction problems. The DISTN is optimized via Backpropagation Through Time (BPTT) and Adam optimizer [97]. Tensorflow and Keras are often used to implement the proposed model [98]. Details on the training are omitted here.

In summary, a deep learning model DISTN is developed that incorporates CNN and LSTM together to learn the spatial and temporal features of movement-based traffic volume data. As shown in Figure 5.2, movement-based traffic volume data for each intersection are first transferred to a 2D image. Multiple images of all target intersections are stacked together and used as the input to CNN. The CNN includes several convolutional layers that can learn the spatial correlations of movement-based traffic volumes. As shown in Figure 5.6, the LSTM model takes the output from the CNN model as input to learn the (within-day, including both "short term" and "long term") temporal dependencies. To capture the periodic patterns of traffic volume data at different days and weeks, a CNN and a LSTM are designed to model, respectively, the effects of within day, daily periodic, and weekly periodic patterns of traffic volumes. All the CNNs and LSTMs are fused together for intersection volume prediction as shown in Figure 5.7.

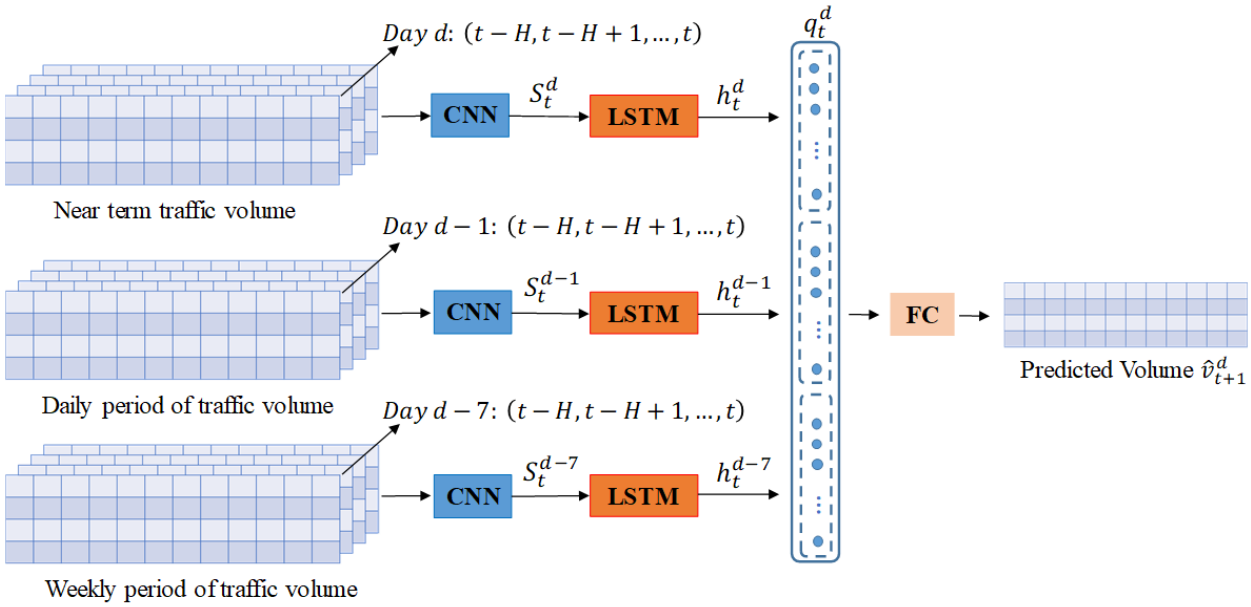



Figure 5.7. The architecture of DISTN for movement-based volume prediction

## 5.3 Fixed-Time Traffic Signal Control based on Predicted Traffic Volume

Section 5.1 and 5.2 illustrate how to estimate and predict movement-based traffic volumes for signalized intersections using real-world data. Practically, the prediction is conducted in a short-term, e.g., every 15 min or 30 min, in order to generate accurate and reliable outputs and capture the general trends of traffic conditions. Based on the predicted volume in very 15 min, the fixed-time traffic signal control method is applied to generate signal timing parameters. The fixed-time signal control is selected for two reasons. First, in order to conduct real-time signal control, the method should be efficient in terms of computational time. Fixed-time traffic signal control generates deterministic signal timing parameters based on experiences, which does not need many calculations. Second, when conducting real-time signal control, only partial observable information can be collected, e.g., traffic volumes, speeds, and locations of sampled vehicles. The benefits of advanced signal timing optimization methods, e.g., the methods present in Chapter 4, cannot be fully exploited and utilized with such limited information. Hence, it is practically efficient and effective to apply fixed-time traffic signal control methods based on 15-min predicted traffic volumes, which is able to adaptive to traffic condition changes in the near future.

For fixed-time signal plans, there are three basic strategies that are commonly used, as was introduced in HCM 2010 [101]. The first strategy aims at equalizing the volume-to-capacity (v/c) ratios for critical lane groups. The green time is determined in proportion to the v/c ratios of the critical lane group for each phase. It is considered as the simplest strategy and might be the only one that is calculated without excessive iterations. A second strategy proposes an optimization function to minimize the total delay of all vehicles. Variations of this strategy involve different performance measures, e.g., the number of stops and fuel consumptions, in their objective

functions. The third strategy aims at equalizing the level of service of all critical lane groups. It improves the first and second strategy in the way that this strategy promotes the level of service of all approaches that are consistent with the overall intersection level of service. This chapter applies the first strategy to calculate signal timing parameters based on predicted traffic volumes because this method is preferred when only limited data, e.g., only volume data, are available for the analysis and only approximated results are desired.

Figure 5.8 illustrates the overall framework of traffic signal timing determination process based on predicted traffic volumes. The simulation network was built in Sumo. It could be a single intersection or a traffic network. Initial traffic demand levels of each movement and traffic signal timing parameters need to be set up before simulation. At each time interval $t$ ($t$ could be 15-30 min), the pre-trained deep learning model (see section 5.2) is applied to predict the traffic volume of each movement at the next time interval $t + 1$. The traffic signal timing parameters, e.g., cycle length, phase duration, and offset (if there are more than one intersection), are calculated based on predicted traffic conditions. These parameters are implemented in Sumo simulation at the next time interval. The program terminated when time step $t$ equal to the total time span $T$.

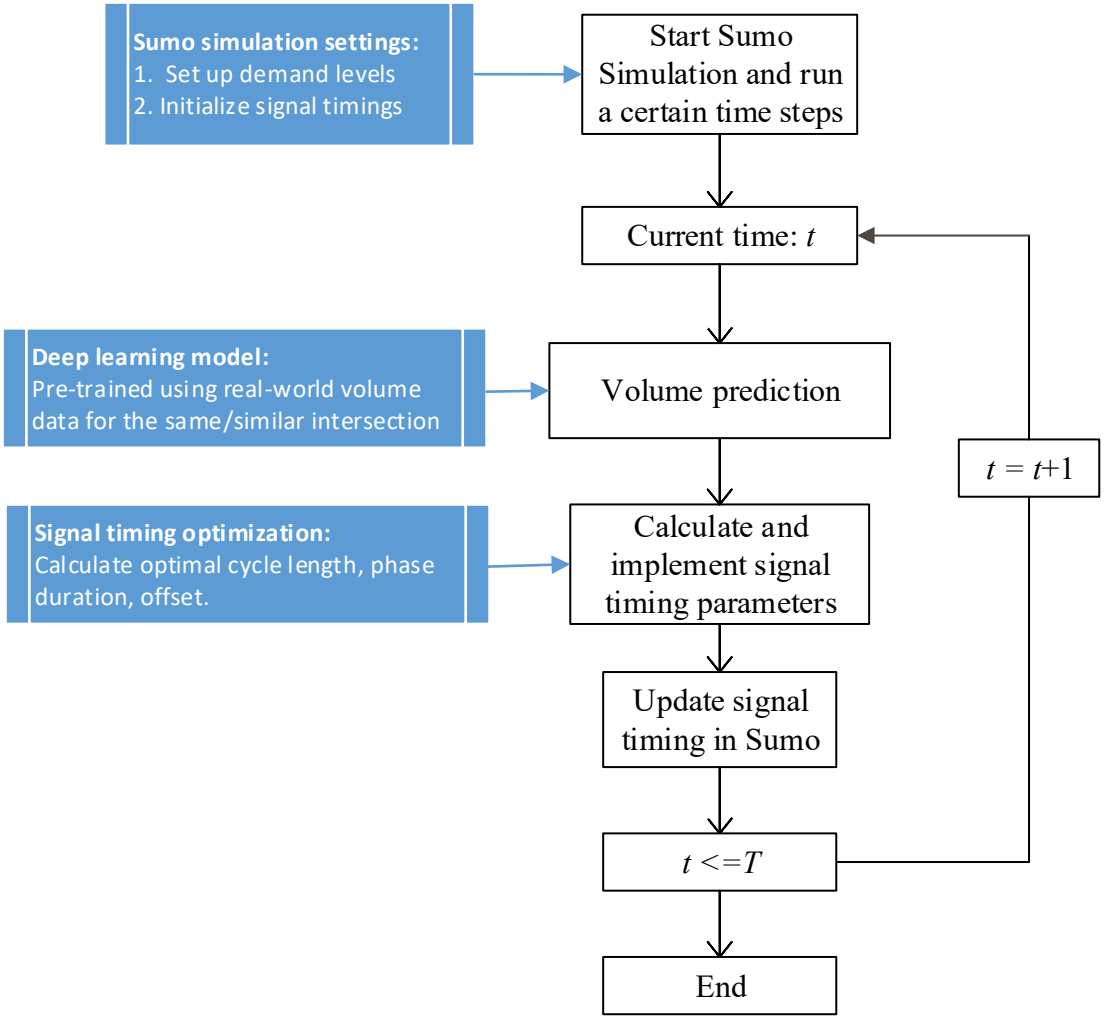



Figure 5.8. The framework of traffic signal timing determination process based on predicted traffic volume

The first step to determine signal timing parameters is to estimate the optimal cycle length. Once the cycle length is determined, the amber and all-red time are subtracted from the total cycle. This results in the total time available for green signal indications. The optimal cycle length is estimated HCM2010 method. Cycle length is usually determined by Eq. (5.21). Noted that the estimated cycle length should always be checked against the reasonable minimum and maximum values. It should not exceed a maximum allowable value by the local jurisdiction (e.g., 150s) and must be long enough to serve queueing vehicles and pedestrians (e.g., 60s may be used if local data are not available).

$$X_c = (\frac{C}{C-L})\sum_{i \in ci} y_{c,i} \tag{5.20}$$

$$C = \frac{LX_c}{X_c - \sum_{i \in ci} y_{c,i}} \tag{5.21}$$

where,

$C$ = cycle length (s),

$L$ = cycle lost time (s),

$X_c$ = pre-seleced critical intersection volume-to-capacity ratio (target value, e.g., 0.8-0.95),

$y_{c,i}$ = critical flow ratio for phase $i = v_i/(N\ s_i)$,

$ci$ = set of critical phases on the critical path,

$X_i$ = volume-to-capacity ratio for lane group $i$,

$v_i$ = demand flow rate for lane group $i$ (veh/h).

Lost time is the total time lost at the beginning of green lights, during yellow change plus red clearance periods that no vehicles are able to pass through an intersection [27]. The HCM defines a default value of 4 seconds per phase for the total lost time. The effective green time (including change and clearance time) for each phase can be computed using Eq.(5.22):

$$g_i = \frac{v_i C}{N_i s_i X_i} = \left(\frac{v}{NS}\right)_i \left(\frac{C}{X_i}\right) \tag{5.22}$$

Where,

$N_i$ = number of lanes in lane group $i$ (ln),

$s_i$ = saturation flow rate for lane group $i$ (veh/h/ln), and

$g_i$ = effective green time for the lane group $i$ (s).

The steps for estimating the cycle length and effective green time for an intersection are summarized:

1. Compute the traffic volume ratio, i.e., $v_i/(N\ s_i)$ for each lane group and identify the critical traffic volume ratio for each phase.
2. Estimate desired cycle length with Eq. (5.21) by substituting a target volume-to-capacity ratio $X_t$ for the critical ratio $X_c$. The value of $X_t$ can be set between the range of 0.8-0.95.
3. Estimate the effective green time for each phase with Eq.(5.22) and the target volume to capacity ratio $X_t$.
4. Check that the effective green time and lost time for each phase sum up to the desired cycle length.

## 5.4 Numerical Experiment

### 5.4.1 *Volume Estimation from PLS*

We test the proposed traffic volume estimation algorithm using taxi GPS data land loop detector data at an intersection in Jinan, China, as shown in Figure 5.9. The loop data and GPS data cover the weekdays and weekends from 8/16/2018 to 9/9/2018.

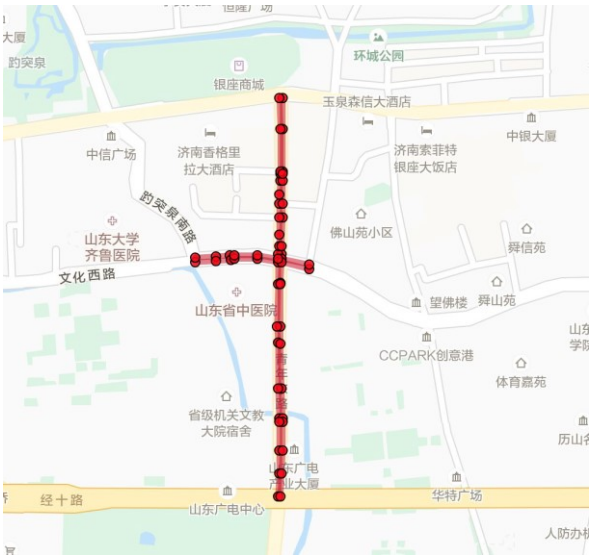
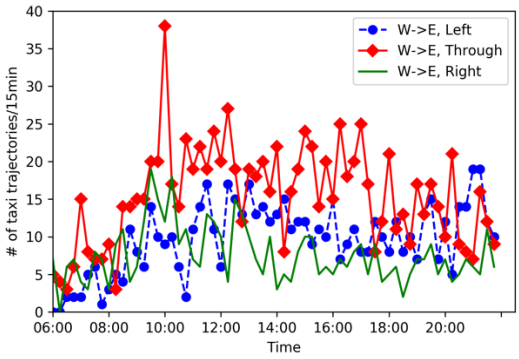


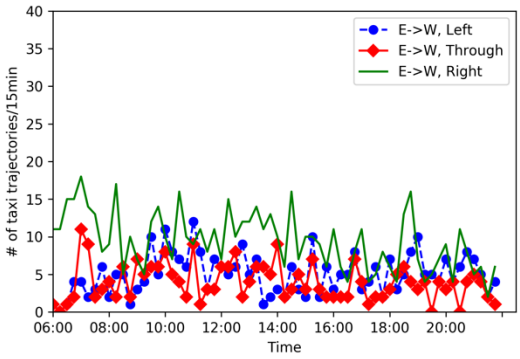

Figure 5.9. The intersection of Wenhuaxi - Luowen

The proposed model is tested on weekdays and weekends separately because of their different traffic patterns. Three features are extracted from taxi trajectory data in every 15 min, i.e., number of trajectories, the average number of stops, and average speed, on each movement. In order to test the PLS-based traffic volume estimation algorithm, all features extracted from taxi trajectory data for all movements from past four hours are used as predictor variables to calculate the ground-truth value of traffic volume at a current time interval.

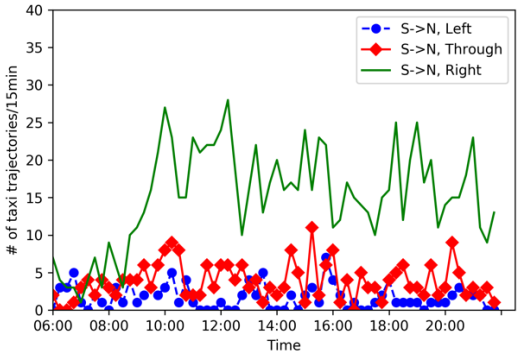
Figure 5.10 - Figure 5.12 depict three features on different movements from 6:00 – 22:00. The issues of outliers and missing data can be identified form the figure. The proposed PLS-based estimation algorithm is robust to the missing data problems because it can extract principal components from the low-rank structure that capture dominant trends of traffic volume, which are extremely significant for volume estimation under relatively low penetration of taxi GPS data.



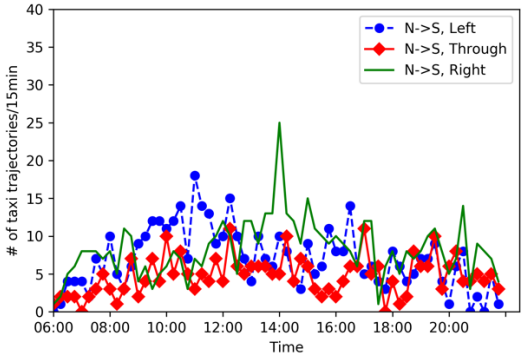

Major Street (W→ E)




Major Street (E→ W)

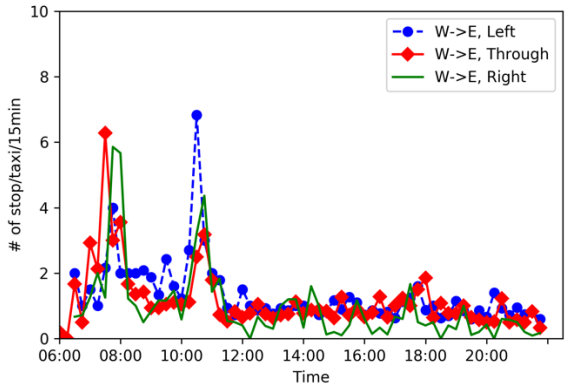



Minor Street (S→N)

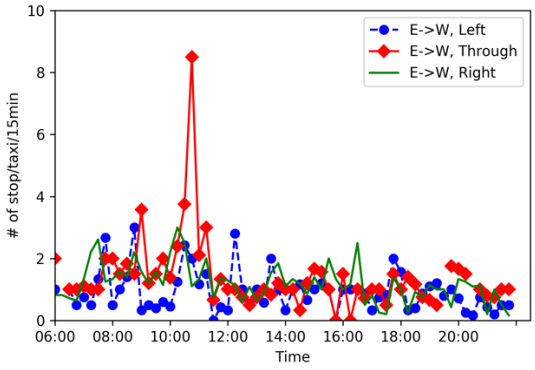



Minor Street(N→S)

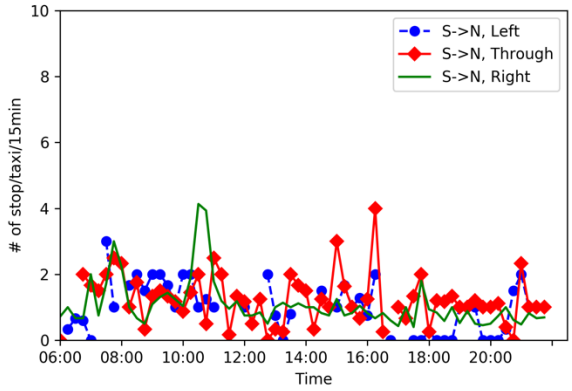
Figure 5.10. # of taxi trajectory/15min for Wenhuaxi Road and Luowen Road



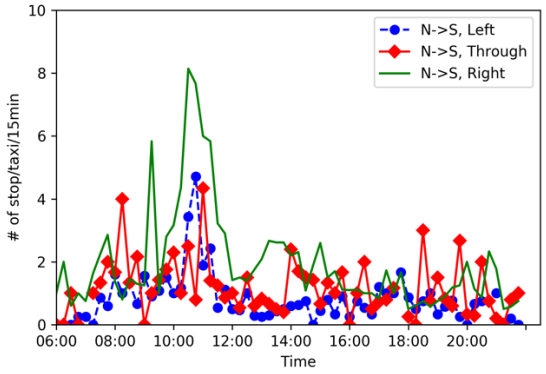

Major Street (W→ E)




Major Street (E→ W)

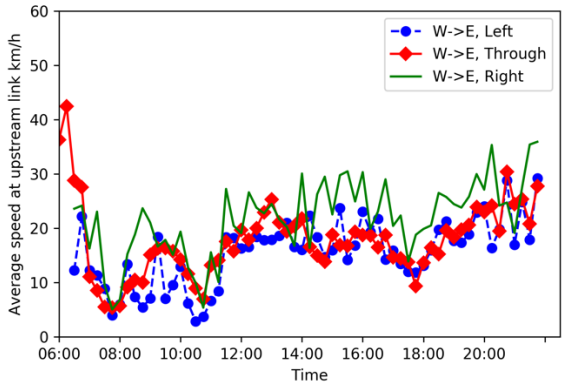



Minor Street (S→N)

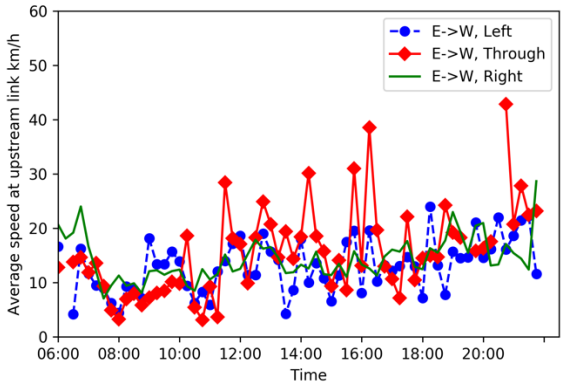



Minor Street(N→S)

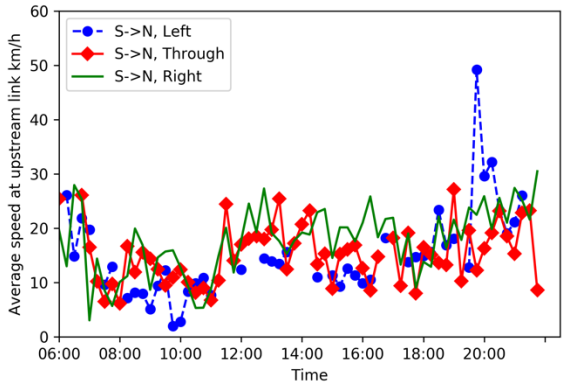
Figure 5.11. # of stop /taxi upstream intersection of Wenhuaxi Road and Luowen Road



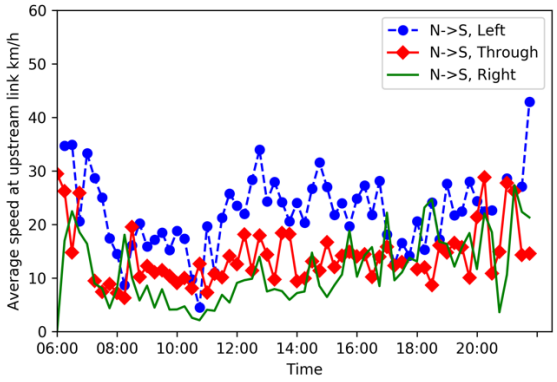

Major Street (W→ E)




Major Street (E→ W)



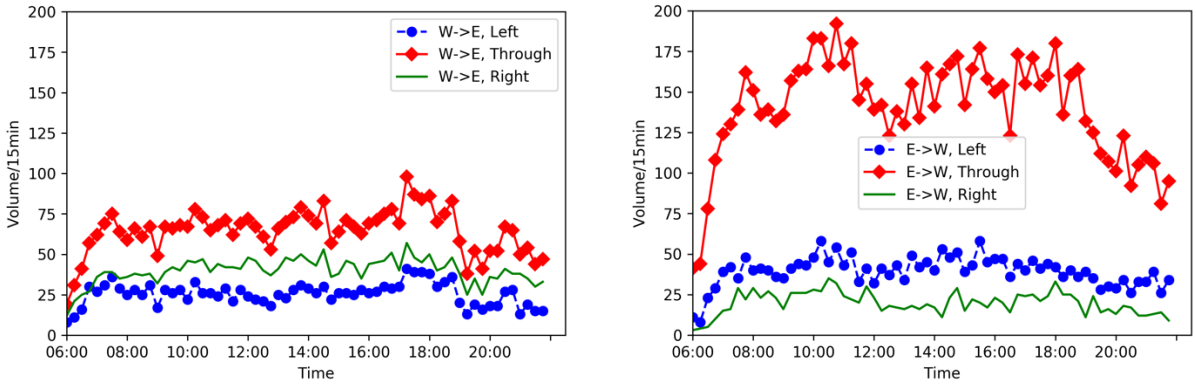

Minor Street (S→N)



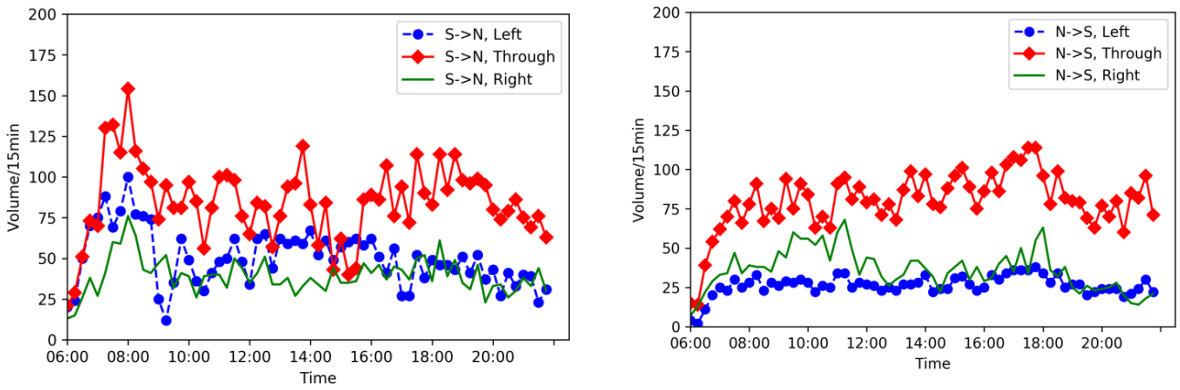

Minor Street(N→S)

Figure 5.12. Average speed upstream intersection of Wenhuaxi Road and Luowen Road

Loop detector data at Luowen – Wenhuaxi Intersection are considered as the ground-truth value of traffic volume, as shown in Figure 5.13. Missing data issue is quite serious in South→North approach, especially on some weekdays from 14:30 to 18:00.




Major Street (W→ E) Major Street (E→ W)




Minor Street (S→N) Minor Street(N→S)

Figure 5.13. Volume at intersection of Wenhuaxi Road and Luowen Road

The input data for PLS have a large feature space, e.g., 12 movement, 48 time intervals per day, and 3 trajectory features (# of trajectories, # of stops and average speed). Such spatial-temporal information contains duplicated information. Figure 5.14 shows the optimal number of PLS components for volume prediction at a certain time step. It suggests that instead of using 12 × 48 × 3 features, 15 optimal predictor components are extracted from the data which produces a minimum of rooted mean square error (RMSE).

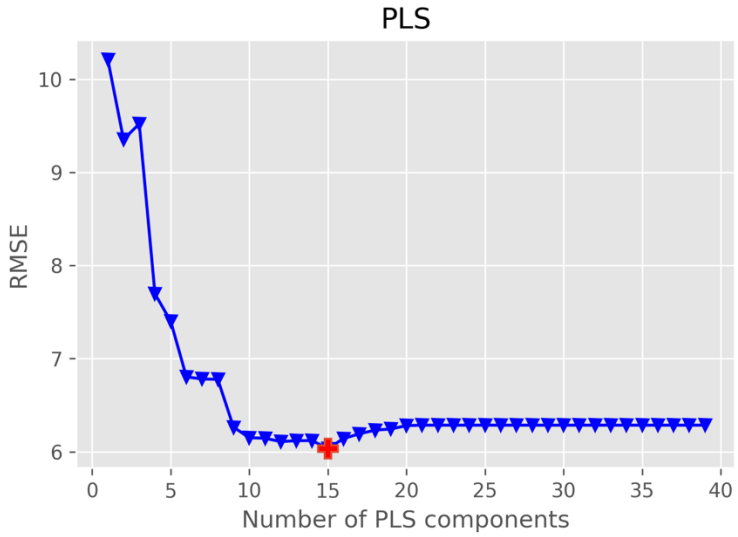



Figure 5.14. Optimal number of PLS components

Table 5.1 shows the overall estimation results for Wenhuaxi – Luowen intersection. Table 5.2 shows the estimation accuracy for each movement. It is observed that movement 7-9 in Southbound have low estimation accuracy at evening peak hours. It is because missing data are existing in loop detectors from 14:45 to 15:45 and from 16:45 to 17:45. For other movements with high-quality loop detector data, the estimation results are satisfied. In addition, the estimation results for through movements are better than turning movements for each approach in term of mean absolute percentage error (MAPE). The higher the volume, the higher estimation accuracy.

Table 5.1. Volume estimation results for an intersection on weekdays

| **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|
| 6.04 | 8.58 | 16.68 |

Table 5.2. Volume estimation results for each movement on weekdays

| **Movement** | **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|---|
| 1: North left | 3.08 | 3.98 | 11.57 |
| 2: North through | 7.40 | 9.43 | 8.99 |
| 3: North right | 6.42 | 8.20 | 19.13 |
| 4: East left | 2.33 | 3.03 | 5.97 |
| 5: East through | 7.16 | 9.12 | 5.21 |
| 6: East right | 2.24 | 2.88 | 11.82 |
| 7: South left | 8.04 | 11.32 | 24.16 |
| 8: South through | 15.54 | 21.96 | 25.36 |
| 9: South right | 8.13 | 10.92 | 36.96 |
| 10: West left | 2.93 | 3.72 | 12.66 |
| 11: West through | 5.30 | 6.54 | 8.52 |
| 12: West right | 3.87 | 4.61 | 9.95 |

Figure 5.15 shows the heat map comparing the estimation results for different movement (vertical axis) and time-of-day (horizontal axis). In additional to the missing data issue existed in movement from 7-9 at evening peak hours, the estimated results can capture the general trend of the ground-truth traffic volume by PLS method.

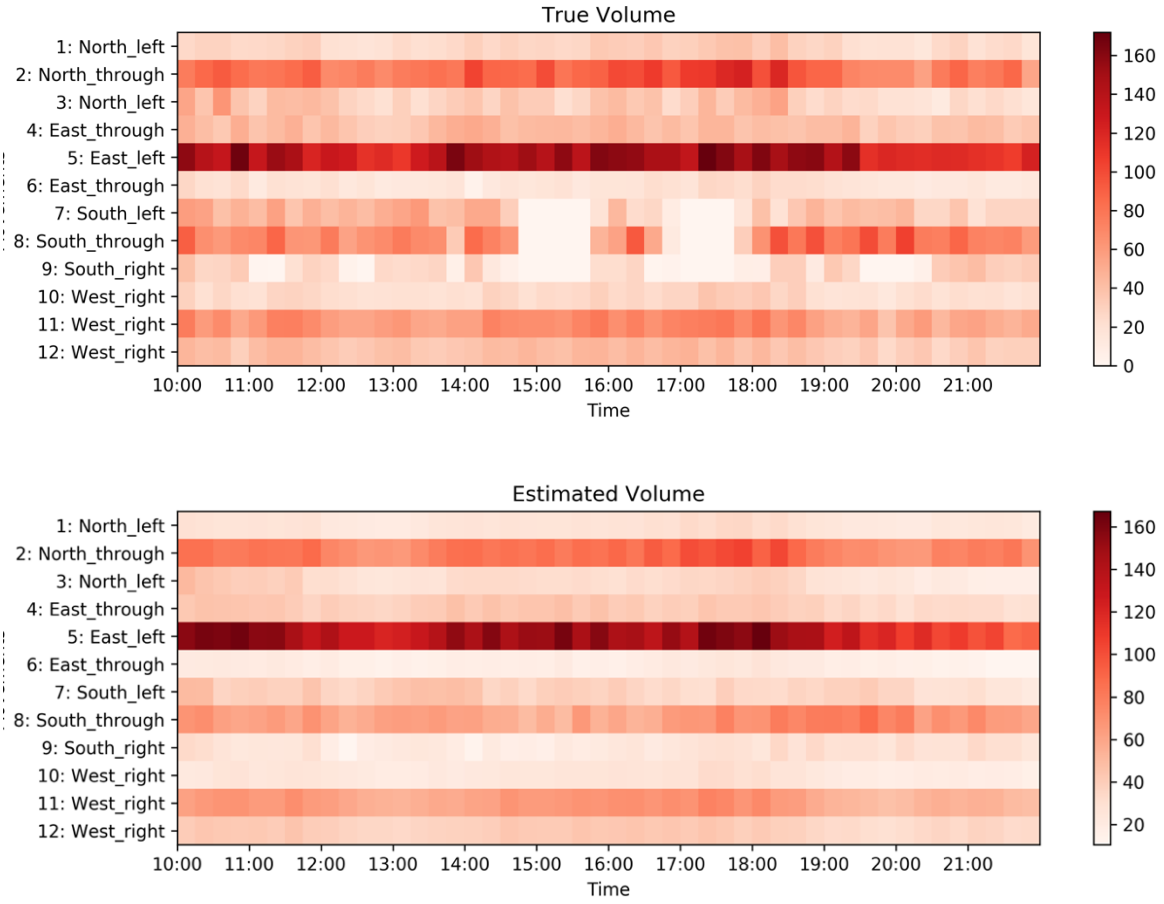



Figure 5.15. Volume estimation results on weekdays

Table 5.3 and Table 5.4 shows estimation results for an intersection and all movements on the weekend. Movement 7-9 has a low estimation accuracy due to missing data existing in loop data. The estimation results for through movement are better than turning movement for each approach. The overall estimation results are similar to weekday case.

Table 5.3. Volume estimation results for an intersection on weekends

| **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|
| 6.02 | 8.13 | 15.72 |

Table 5.4. Volume estimation results for each movement on weekends

| **Movement** | **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|---|
| 1: North left | 2.81 | 3.44 | 12.16 |
| 2: North through | 7.57 | 9.69 | 10.09 |
| 3: North right | 4.53 | 5.47 | 18.42 |
| 4: East left | 4.14 | 5.21 | 10.98 |
| 5: East through | 12.00 | 15.77 | 8.95 |
| 6: East right | 2.70 | 3.58 | 16.11 |
| 7: South left | 6.06 | 7.96 | 27.00 |
| 8: South through | 11.13 | 13.72 | 20.93 |
| 9: South right | 9.38 | 12.02 | 43.02 |
| 10: West left | 2.65 | 3.34 | 12.26 |
| 11: West through | 5.50 | 6.48 | 9.57 |
| 12: West right | 3.84 | 4.76 | 10.53 |

Figure 5.16 compares the estimated volume with the ground truth volume on weekends. The estimated values have relatively low variations. As can be seen from the figure, the colors of nearby cells are similar comparing to the ground-truth heat map. PLS method generates outputs based on selected predicted and response components that ignore irrelevant information from the raw data, as shown in section 5.4.1.

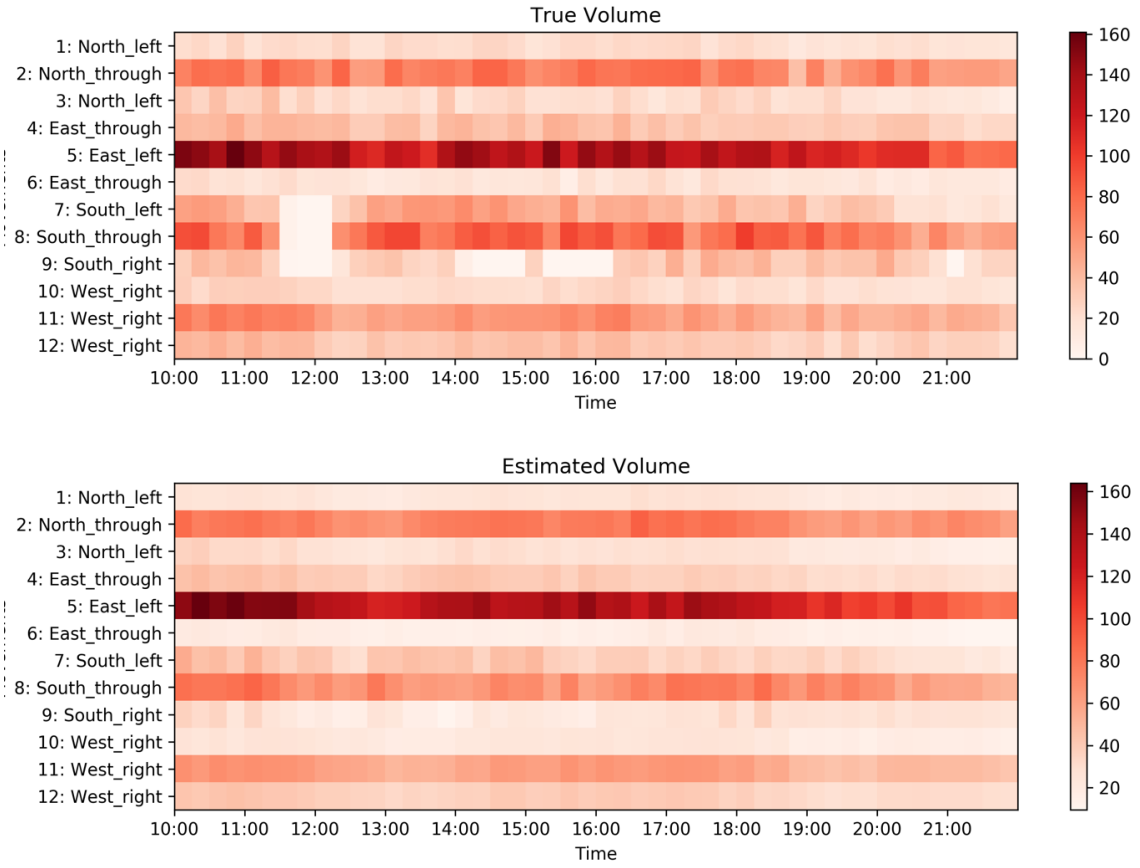



Figure 5.16. Volume estimation results on weekends

Table 5.5 shows the estimation results with regard to various estimation horizon. As the estimation horizon increases, the value of MAPE decreases. Longer estimation horizons lead to lower fluctuations in traffic volume and better estimation performance. However, in order to generate effective signal timing plans, a shorter estimation horizon, e.g., 15 min, is selected for fixed-time signal timing determination in section 5.4.3.

Table 5.5. Volume estimation results for each movement on weekends

| **Estimation Horizon** | **MAE** (Vehicle/Estimation Horizon) | **RMSE** (Vehicle/Estimation Horizon) | **MAPE** (%) |
|---|---|---|---|
| 15 min | 6.04 | 8.58 | 16.68 |
| 30 min | 13.98 | 19.94 | 13.42 |
| 1 hour | 25.02 | 37.56 | 11.78 |
| 2 hour | 47.12 | 74.92 | 11.23 |
| 3 hour | 70.23 | 112.99 | 10.98 |
| 4 hour | 93.28 | 153.33 | 10.86 |

### 5.4.2 *Volume Prediction using DISTM*

#### 5.4.2.1 Real world case study

In this section, we first test the proposed method using real-world traffic volume data on an urban corridor in Jinan, China. The corridor contains three signalized intersections. Each intersection contains four approaches and twelve movements. The geometric layout of the corridor is shown in Figure 5.17. The W-E direction is the main street while the N-S direction is the minor street. Traffic volumes for the three target intersections were collected from loop detectors from 8/16/2018 to 9/9/2018 for every 15 min.

In this experiment, the historical sequence $H= 16$, i.e., 4 hours, is used to predict the movement-based traffic volume for the next 15 min. The total samples are randomly split into the training set and the validation set with a ratio of 8:2. The input data to the first layer of the CNN in the proposed deep learning approach is 4D tensors with the dimension of $[s, H, 4 * Ix, 4 * Iy] = [0.8 * s, 16, 4, 12]$, where $s$ represents the number of samples in the training set. Here $Ix = 1$ and $Iy = 3$ indicating a corridor with three intersections in the West-East direction. The output data of the proposed DISTN model is a 1-dimension vector with 36 components representing the movement-based traffic volumes of three intersections for the predicted time step.


Figure 5.17. Geometric layout of the corridor

For our proposed model, we set $K = 3$ (number of layers), $f = 2 \times 2$ (kernel size), $\lambda = 64$ (number of filters), and $d = 64$ (dimensions of the outputs) in the CNN. For LSTM, as aforementioned, the sequence length is set to be $H = 16$ (i.e., 4 hours). The input dimension of LSTM is $[s, H, d]$, where $s$ represents the number of samples in the training set. The number of layers in LSTM is 2. The output dimension is 36, representing traffic volumes of the 36 movements for the study three intersections at the predicted time step. Activation functions for the FC layer are ReLU. The batch size in the experiment is set to 64. Epoch is set as 50. In order to address the overfitting problem, the weight penalty is applied for regularization using L1 and L2 norms. In the training process, the model minimizes the Mean Square Error loss using Adam optimizer. The evaluation metrics include the Mean Absolute Error (MAE), Rooted Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE). All the hyperparameters (e.g., # of layers, kernel size, # of filters, etc.) in this experiment are tuned multiple times aiming at optimizing the

prediction results. Due to space limitation, the comparisons between different hyperparameters are omitted here.

The overall prediction results for the three intersections are shown in Table 5.6. The overall MAE, RMSE, and MAPE are 2.56, 3.60, and 16.74% respectively.

Table 5.6. Volume prediction results for all intersections

| **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|
| 2.56 | 3.60 | 16.74 |

The proposed model is compared with several traffic volume prediction methods, including ARIMA, Kalman filter, Partial Least Square (PLS), Dynamic Mode Decomposition (DMD), and LSTM. The pros and cons of each method are already summarized in Table 2.1. The comparisons of prediction results are shown in Table 5.7. The proposed DISTN model achieves the lowest MAPE among all methods. ARIMA produces the worst MAPE because it only relies on the historical pattern of the traffic volume data and does not account for any spatial features. Kalman filter and DMD perform better than ARIMA, probably due to their ability to denoise the input data and extract relevant and key information from the dataset. The PLS-based prediction algorithm improves the prediction results from ARIMA, Kalman filters, and DMD. The input data to the PLS is a 2D matrix that contains both spatial features (for different movements) and temporal features (for different historical time steps), which provides more information for prediction. LSTM shows the advantages on processing sequential data and modeling the long-term dependencies. The complex underlying relationships of the traffic volume data can be captured by LSTM, which makes its performance improved from all the other methods. Here, the CNN model takes the time-space matrix as the input and learn the spatiotemporal features simultaneously, see Ma et al. [93]. The prediction results from CNN is also promising, resulting in 17.42% of MAPE. The DISTN

model further improves the prediction performances by capturing the unique geographical structure of the movement-based traffic volumes at signalized intersections. The experimental results in Table 5.7 are consistent with the summary of the pros/cons of the different volume methods in Table 2.1.

Table 5.7. Prediction results for different models

| **Methods** | **MAPE** (%) |
|---|---|
| ARIMA | 31.15 |
| Kalman Filter | 24.04 |
| DMD | 20.86 |
| PLS | 18.39 |
| LSTM | 17.93 |
| CNN | 17.42 |
| **DISTN** | 16.74 |

Figure 5.18 shows the comparisons between the ground truth values and predicted values of traffic volume for the 36 movements for a certain time step. The predicted values estimated from the proposed model can capture fairly well the fluctuations of the true traffic volume data.

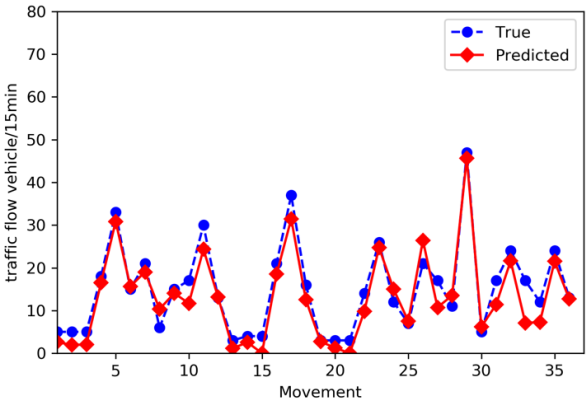
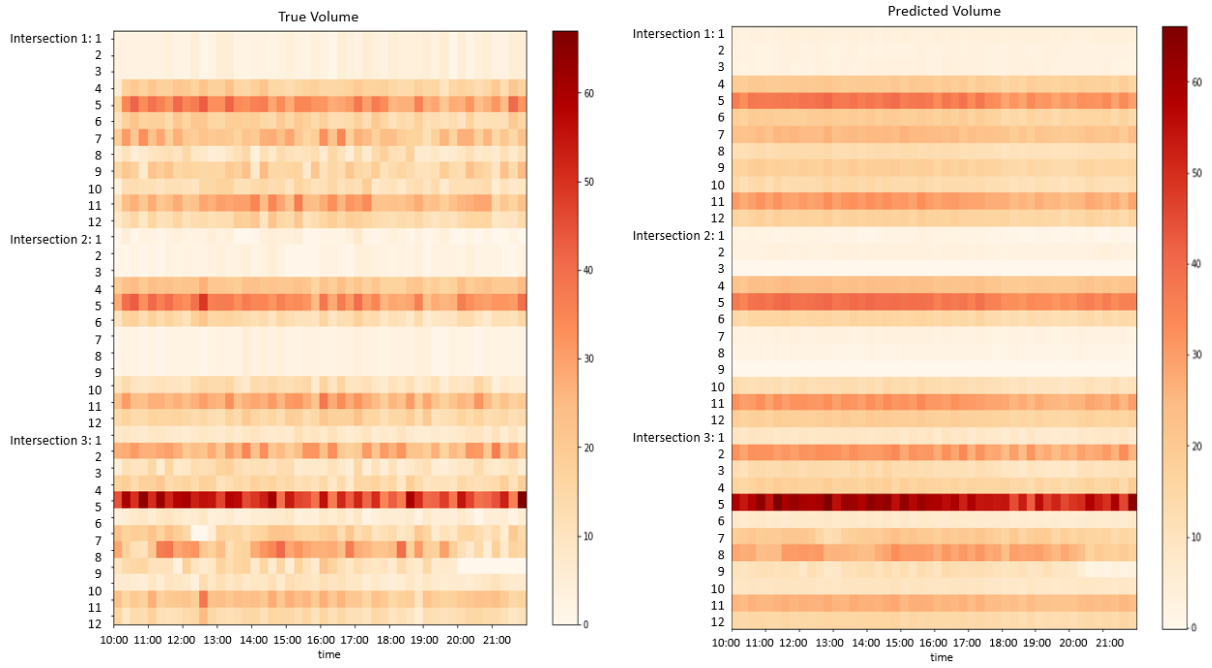



Figure 5.18. Traffic volume for 36 movements (3 intersections) at a certain time step

Figure 5.19 shows the ground truth and predicted traffic volumes for each movement of the three intersections for a certain day. The variations of the predicted volume are generally smaller than the ground truth value. The movement-based traffic volumes are random due to the stop-and-

go conditions at the intersections. The trained model does not aim at capturing such randomness which will easily cause overfitting issues. The figure indicates that the general trends of traffic volumes can be captured by the proposed model. For example, movement 5 of intersection 3 have the highest volumes while movement 1-3 and 7-9 of intersection 2 have relatively low volumes, both of which can be captured by the following heat maps of true volumes and predicted volumes.




(a) True volume (b) Predicted volume

Figure 5.19. Ground truth and predicted traffic volume for 36 movements (3 intersections) at a certain day

After presenting the overall performance of the proposed model, we next present the prediction accuracies of individual movements. Table 5.8 shows the prediction accuracy of each movement for the 2nd intersection. The through movement always achieves the lowest MAPE compared with the left-turn and right-turn movements. For example, Northbound through movement (Movement 2) has a MAPE of 13.14%, which are better than the left and right turns (Movement 1 and 3). Similar trends can be identified for other approaches. This may be due to the larger traffic volumes of the through movements than the turning movements. In another word,

lower traffic volumes will lead to more random traffic arrival patterns, making volume prediction much harder. In addition, the prediction results for the side street, i.e., the Northbound (Movement 1-3) and the Southbound (Movement 7-9), are generally worse than those for the main street.

Table 5.8. Volume predication results for each movement of Intersection 2

| **Movement** | **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|---|
| 1: North left | 1.44 | 1.92 | 14.05 |
| 2: North through | 0.94 | 1.14 | 13.14 |
| 3: North right | 2.52 | 2.76 | 20.86 |
| 4: East left | 2.51 | 3.13 | 19.23 |
| 5: East through | 3.75 | 4.76 | 17.39 |
| 6: East right | 2.18 | 2.78 | 17.49 |
| 7: South left | 0.85 | 1.02 | 25.27 |
| 8: South through | 1.01 | 1.24 | 13.22 |
| 9: South right | 1.98 | 2.21 | 22.12 |
| 10: West left | 1.98 | 2.51 | 20.91 |
| 11: West through | 3.28 | 4.15 | 14.69 |
| 12: West right | 2.06 | 2.69 | 17.77 |

Although the proposed model can predict traffic volume for all movements of the three intersections simultaneously, the prediction accuracies of the three intersections are different. As Intersection 2 is in the middle of the corridor, the upstream and downstream traffic conditions of the intersection are provided by correlations with the volumes of Intersections 1 and 3 in the West-East direction. However, Intersections 1 and 3 are at the boundaries of the corridor with only one upstream or downstream intersection, where less traffic information is available.

Table 5.9 shows the results for the three intersections separately. Intersection 2 has minimal MAE, RMSE, and MAPE comparing with the other two intersections. The results indicate that by considering the spatial dependencies of traffic volumes in a larger area, e.g., considering more intersections together, the prediction results can be improved.

Table 5.9. Volume prediction results for each intersection

| **Intersection** | **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|---|
| Intersection 1 | 2.60 | 3.56 | 18.84 |
| Intersection 2 | 2.03 | 2.74 | 15.33 |
| Intersection 3 | 3.10 | 4.31 | 17.44 |

The proposed model accounts for the periodic patterns of traffic volumes by using the within a day, daily periodic, and weekly periodic features in prediction. In order to validate that the periodic trends can indeed help improve the prediction accuracy, the DISTN model is tested under three scenarios, as shown in Table 5.10. When considering all three features in scenarios 3, the minimum MAPE is attained.

Table 5.10. Volume prediction results using different time sequences

| **Time sequence** | **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|---|
| 1.Within day | 2.97 | 3.96 | 18.48 |
| 2.Within day + daily period | 2.68 | 3.67 | 17.67 |
| 3.Within day + daily period + weekly period | 2.56 | 3.60 | 16.74 |

The sequence length (i.e., *H*) indicates how long we should look back when predicting traffic volumes for the next time step. Figure 5.20 shows the MAPEs for various historical time intervals (represented in hours). The MAPE decreases dramatically up to 4 $hour$, and then starts to increase. In addition, longer sequence length in LSTM requires more parameters to learn, making the training process harder. So, the proposed model takes the historical four-hour traffic volumes as to the input for prediction.

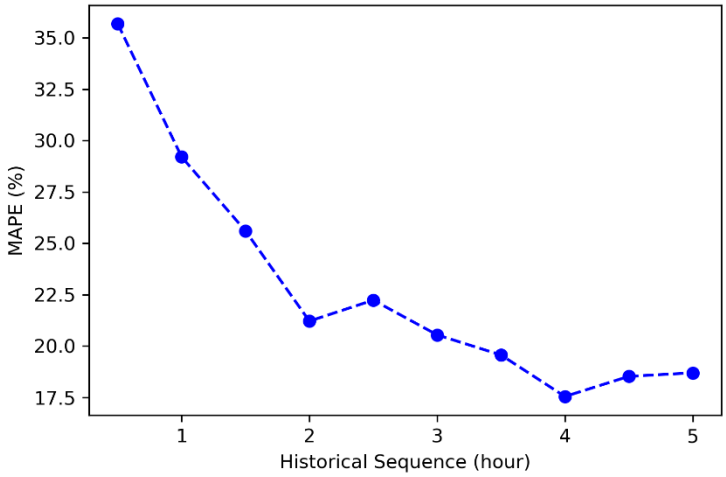



Figure 5.20. MAPE of various historical sequences for the DISTN

Table 5.11 shows the prediction results for various prediction horizons. As the horizon increases, the MAPE decreases. Longer prediction horizon represents a larger traffic volume, which makes the prediction task relatively easier comparing with the short prediction horizon.

Table 5.11. Volume prediction results for various prediction horizon

| **Prediction Horizon** | **MAE** (Vehicle/Estimation Horizon) | **RMSE** (Vehicle/Estimation Horizon) | **MAPE** (%) |
|---|---|---|---|
| 15 min | 2.56 | 3.60 | 16.74 |
| 30 min | 9.98 | 12.94 | 13.42 |
| 1 hour | 15.02 | 27.56 | 11.78 |
| 2 hours | 37.12 | 64.92 | 11.23 |
| 3 hours | 60.23 | 102.99 | 10.98 |
| 4 hours | 83.28 | 143.33 | 10.86 |

5.4.2.2 Simulation Network

In order to test whether the proposed model is robust to various network structures, a testing traffic network is built in the traffic simulator VISSIM, as shown in Figure 5.21. The parallel corridors contain eight intersections with various distances between two neighboring intersections. Based on the geometric relationship of the eight intersections, eight $4 \times 4$ images are stacked together.

The dataset contains 10 days by running simulation in VISSM 10 times with random seeds. The traffic demand is set to be 800 vph for N-S direction and 200 vph for W-E direction. Only the within a day and daily periodic patterns are considered (without weekly patterns) for the simulation network, and traffic volume data for each movement are collected in every 5 min.

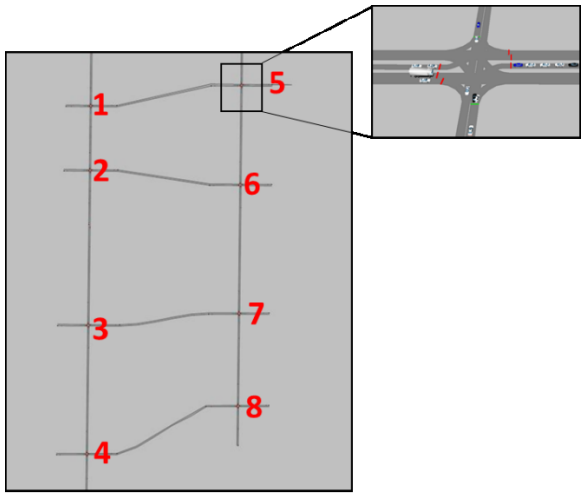



Figure 5.21. A parallel corridor in the simulation network

In this simulation network, we set $K = 3$ (number of layers), $f = 4 \times 4$ (kernel size), $\lambda = 64$ (number of filters), and $d = 128$ (number of outputs) in the CNN. For LSTM, the sequence length is set to be $H = 24$ (i.e., 2 hours). The input dimension of LSTM is $[s, H = 24, d = 128]$. The number of layers in LSTM is 2. The output dimension is 96, representing traffic volumes of the 96 movements for the total 8 intersections at the predicted time step. Activation functions for the fully connected layer are ReLU. The batch size in the experiment is set to 32. Epoch is set as 50.

Comparisons of the prediction results from different models are shown in Table 5.12. Traffic volumes in the real-world corridor have larger fluctuations while the periodic trends of traffic volumes can be identified, e.g., within a day, daily and weekly trends. However, in the simulation network, the traffic demands are relatively stable without clear periodic trends, i.e., the arrival vehicles in simulation follow a certain distribution. As a result, the prediction accuracies of

simulation data for all methods are generally higher than those of the real-world data. For simulation data, the performance improvement of the DISTN model over other methods is not as significant as that for real-world data. For example, the MAPE of the DISTN model is slightly better than CNN and PLS.

Table 5.12. Prediction results of simulation network from different models

| **Methods** | **MAPE** (%) |
|---|---|
| ARIMA | 13.24 |
| Kalman Filter | 11.34 |
| DMD | 11.78 |
| PLS | 9.94 |
| LSTM | 10.84 |
| CNN | 9.71 |
| **DISTN** | 9.57 |

Table 5.13 shows the prediction results for all intersections in the simulation network. The MAPEs of different intersections range from 7.74% to 11.54%. Intersections 4, 5 and 8 have the highest MAPE. They are the intersections at the boundary of the network with limited traffic conditions from upstream of downstream intersections.

Table 5.13. Volume prediction results for all intersections in simulation network 2

| **Intersection** | **MAE** (vehicle/15min) | **RMSE** (vehicle/15min) | **MAPE** (%) |
|---|---|---|---|
| 1 | 10.34 | 11.93 | 7.87 |
| 2 | 12.57 | 14.34 | 8.97 |
| 3 | 11.10 | 14.76 | 8.55 |
| 4 | 13.68 | 15.12 | 11.54 |
| 5 | 12.89 | 14.67 | 10.95 |
| 6 | 10.25 | 12.79 | 7.74 |
| 7 | 11.97 | 15.23 | 8.75 |
| 8 | 14.23 | 14.42 | 11.81 |

Figure 5.22 shows the comparisons between the ground-truth and predicted traffic volume of 96 movements (8 intersections) for a certain time step in the simulation network. The predicted values can capture the trends of the ground-truth traffic volumes.

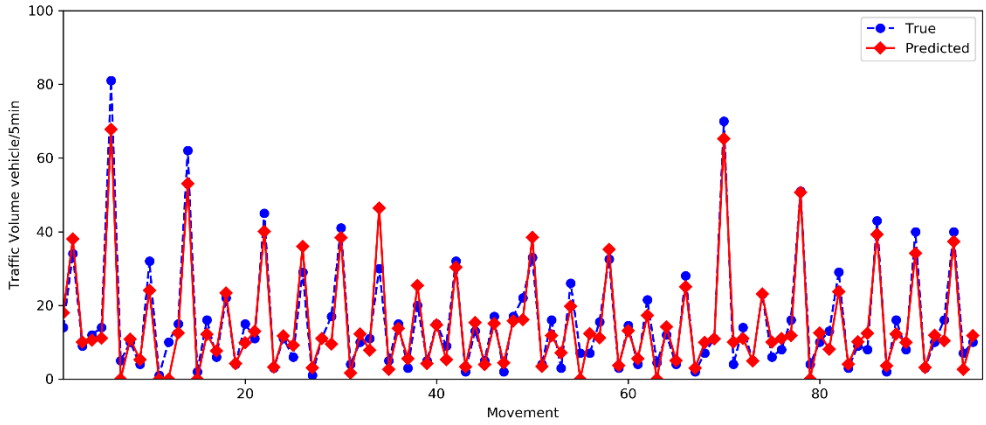
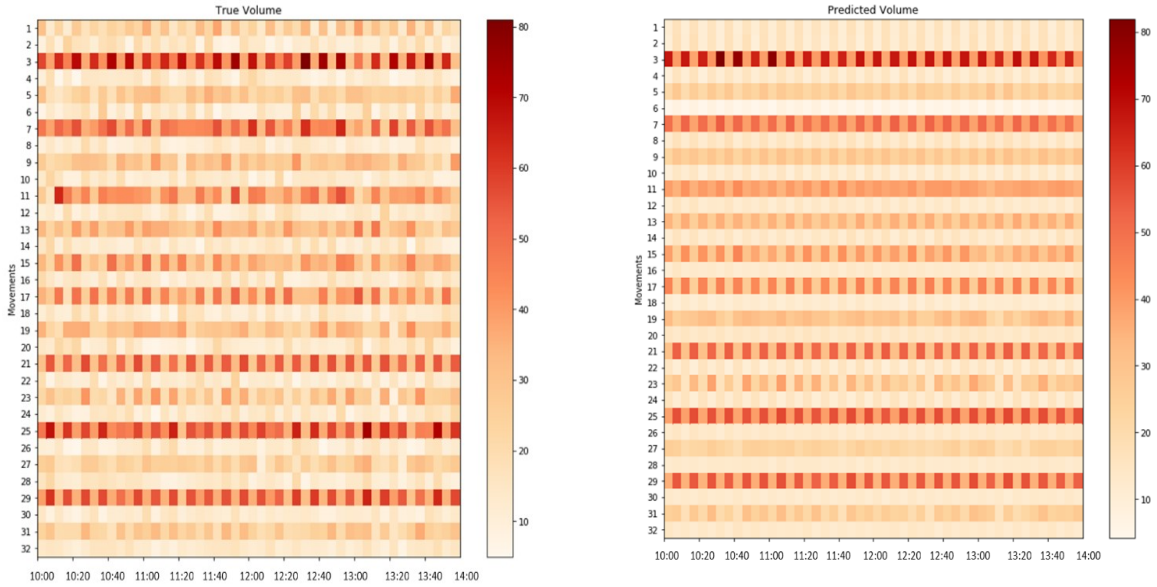



Figure 5.22. Traffic volume for 96 movements (8 intersections) in simulation at a certain time step

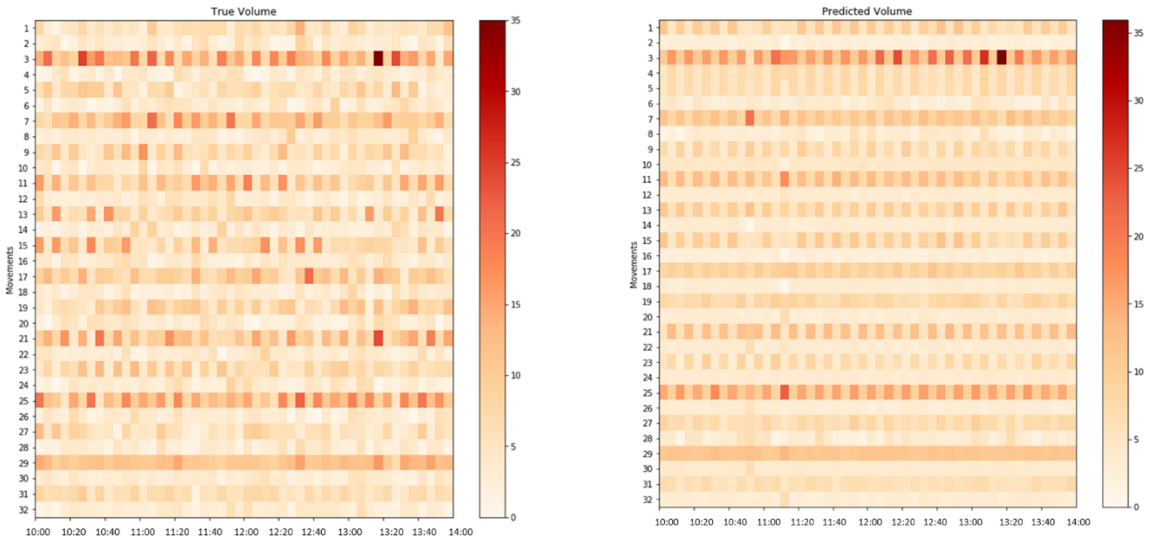
Figure 5.23 shows the comparisons between the true and predicted traffic volumes for 32 through movements for a certain time period. Predicted traffic volumes have smaller variance over time comparing to true values. This is because, again, the proposed model is not developed to capture the randomness of movement-based traffic volume due to stop-and-go conditions at intersections. Similar trends have been found above for real-world data.




**(a)** True volume **(b)** Predicted volume

Figure 5.23. Comparisons for 32 through movements

Figure 5.24 shows the comparisons between true and predicted traffic volumes for 32 left turn movements. The volumes for turning movements are generally smaller compared to the volumes of the through movements. It is observed that the larger the traffic volumes, the more fluctuations of the volumes can be captured by the proposed model. The trends of right turn movements are similar to those of the left turn movements and are thus omitted here to save space.




(a) True volume (b) Predicted volume

Figure 5.24. Comparisons for 32 Left turn movements

Figure 5.25 shows the influence of the historical sequence (i.e., *H*) used in the proposed model on prediction accuracy. MAPE becomes stable if the historical sequence is larger than 2 hours. As a result, the proposed model uses 2-hour historical traffic volumes to predict the future 5 min traffic volumes. Compared to the real-world data in Figure 5.20, the historical sequence used for simulation data is largely reduced (from 4 hours to 2 hours). The temporal correlations of traffic volumes in simulation data are not as strong as those of the real-world data, and this is why longer historical sequences will not make significant contributions to prediction accuracy.

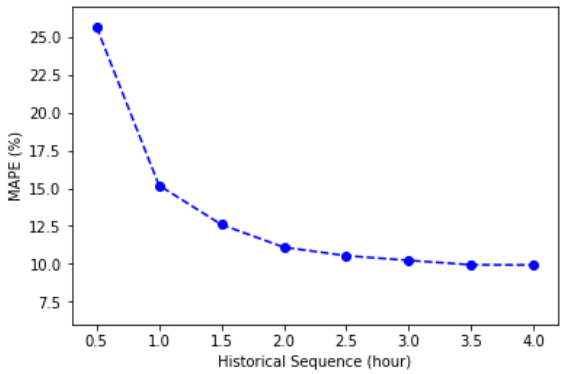



Figure 5.25. MAPE of various historical sequences for simulation data

### 5.4.3 *Fixed time Traffic Signal Control based on predicted traffic volume*

The procedure of fixed-time signal timing determination is illustrated by a sample intersection, Wenhuaxi-Luowen Intersection, Jinan, China as shown in Figure 5.9. Fixed time signal timing parameter determination is developed based on Figure 5.8. The simulation network is built in Sumo that replicates the geometry of the real intersection. The demand levels (arrival vehicle rates) are determined based on real-world traffic data.

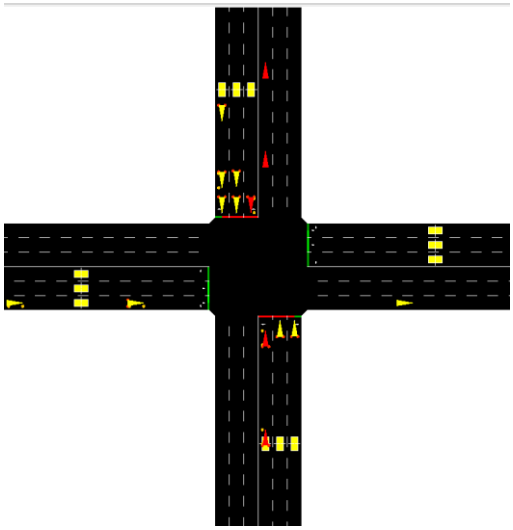


Figure 5.26. Simulation network in Sumo

The loop detector data from 6 am – 10 pm on 8/16/2018 were aggregated to every 15 min and used as traffic demand in Sumo simulation. Given the demand levels in every 15 min, the predicted traffic volumes for each of the 12 movements are calculated using DSTIN proposed in 5.2.4. Table 5.14 shows an example of the predicted traffic volume obtained from real data. '*L*' stands for turning left, '*T* for going straight and '*R*' for turning right. It is noticed that eastbound through movement (East - T) exhibit higher traffic demand than other movements.

Table 5.14. Sample predicted volume data at Wenhuaxi-Luowen Intersection

| Time | Traffiv Volume for Each Movement (vehicle per hour) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | East-L | East-T | East-R | West-L | West-T | West-R | North-L | North-T | North-R | South-L | South-T | South-R |
| 19:00 | 136 | 528 | 68 | 72 | 216 | 148 | 112 | 340 | 120 | 176 | 456 | 112 |
| 19:15 | 160 | 520 | 76 | 76 | 252 | 176 | 92 | 288 | 80 | 128 | 292 | 160 |
| 19:30 | 108 | 420 | 68 | 60 | 188 | 128 | 100 | 328 | 48 | 160 | 360 | 140 |
| 19:45 | 124 | 464 | 76 | 80 | 200 | 124 | 88 | 288 | 84 | 148 | 368 | 100 |
| 20:00 | 116 | 412 | 44 | 80 | 204 | 124 | 56 | 224 | 88 | 192 | 308 | 148 |
| 20:15 | 116 | 416 | 52 | 56 | 160 | 104 | 68 | 200 | 48 | 152 | 272 | 136 |
| 20:30 | 140 | 440 | 56 | 76 | 224 | 152 | 88 | 276 | 84 | 76 | 260 | 92 |
| 20:45 | 136 | 436 | 64 | 80 | 224 | 148 | 80 | 272 | 88 | 120 | 220 | 80 |
| 21:00 | 124 | 360 | 48 | 80 | 252 | 172 | 92 | 312 | 68 | 120 | 288 | 172 |

As shown in Table 5.14, the traffic volumes on left-turn movements are relatively low. The protected left turn phase is recommended when any of the following criteria are met [14]. First, the left turn has demand over 200 vph. Second, the cross product of left turn demand and opposing lane-average through demand for one hour exceeds 50,000. Clearly, this intersection does not need protected left turn phase. Therefore, two phases are applied for Luowen-Wenhuaxi intersection, as shown in Figure 5.27. The first phase serves all movements in eastbound and westbound directions. The second phase serves all movements in northbound and southbound directions.

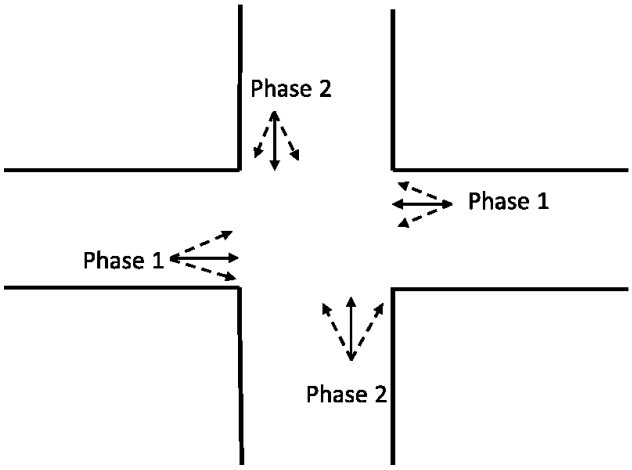



Figure 5.27. Signal phases for Wenhuaxi-Luowen Intersection.

In order to validate the effectiveness of the proposed model, the state-of-the-art methods and the real-world signal plan are implemented in the same Sumo simulation network and compared with the proposed mode. The details of each method are explained:

1. Real world signal plans

The real-world signal plan is implemented in Wenhuaxi-luowen intersection from 01/01/2018 - 12/31/2018 using a fixed-time scheme. The signal timing parameters are determined based on different time-of-day with signal plans variations in every three hours. Hence, there are totally 12 plans implemented per day with various cycle length and phase durations.

2. Actuated signal plan

Actuated traffic signal control system is widely deployed in the US and is recommended for fluctuated demand levels in different movements. For comparison purpose, the signal timing parameters in the actuated plan are set according to the real-world signal plans that the maximum green for each phase is exactly equal to the phase duration in method 1. This is the only signal plan in four mentioned methods that have variable cycle lengths.

3. Proposed fixed-time plan based on predicted volumes

The proposed fixed-time signal plan calculates signal parameters every 15 min based on the predicted traffic volumes. Comparing to other methods, this method aims at adjusting traffic signal parameters based on future traffic conditions. The cycle length is determined based on the HCM 2010 quick estimation method.

4. Proposed fixed-time plan with minimum cycle length constraints

The fourth method is the variation of the third method that a minimum cycle length constraint is added to the fixed-time signal plan. This is because without the constraints, the phase changing frequencies are too high or phase duration is too small, which is not preferable for real-world implementation.

Table 5.15 shows the cycle length every 15 min under the four signal timing methods. Note that actuated signal plans have variable cycle length per cycle because it is determined by the real-

time arrival vehicles. Real-world signal plans implemented in Wenhuaxi-Luowen Intersection have a fixed cycle length for 120s. The proposed fixed-time plan has various cycles of lengths per 15 min that the cycle length decreases from 60s to 25s. It is because the traffic demand decreasing from 19:00 to 21:00 that the cycle length also decreases, see Eq. (5.21). In method 4, the minimum cycle length is set as 50s.

Table 5.15. Cycle length under various signal timing methods

| Methods | Time of Day | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 19:00 | 19:15 | 19:30 | 19:45 | 20:00 | 20:15 | 20:30 | 20:45 | 21:00 |
| 1. Real-world signal plans | 120 | 120 | 120 | 120 | 120 | 120 | 120 | 120 | 120 |
| 2. Actuated Signal plans | — | — | — | — | — | — | — | — | — |
| 3. Proposed Fix-time plans | 60 | 40 | 35 | 35 | 30 | 25 | 25 | 25 | 25 |
| 4. Proposed Fix-time with minimum C constraint | 60 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 |

Table 5.16 shows the phase durations every 15 min under four signal timing methods. The real-world signal plan implements 77s for phase 1 and 37s for phase 2. For actuated signal plan, only the minimum green and maximum green need to be determined. In the numerical test, the maximum green is 77s for phase 1 and 37s for phase 2, which are equal to the phase durations of the real-world plan. The minimum green is 40s for phase 1 and 20s for phase 2. For the proposed signal plan, the phase durations for phase 1 and phase 2 are all decreasing. When a minimum cycle length constraint is added, the phase durations becomes stable.

Table 5.16. Phase durations under various signal timing methods

| Methods | Time of Day | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 19:00 | 19:15 | 19:30 | 19:45 | 20:00 | 20:15 | 20:30 | 20:45 | 21:00 |
| 1. Real-world signal plans | [77,37] | [77,37] | [77,37] | [77,37] | [77,37] | [77,37] | [77,37] | [77,37] | [77,37] |
| 2. Actuated Signal plans | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] | [(40,77),(20,37)] |
| 3. Proposed Fix-time plans | [25, 27] | [17, 15] | [12, 15] | [13, 14] | [10, 12] | [8, 9] | [9, 8] | [9, 8] | [10,7] |
| 4. Proposed Fix-time with minimum C constraint | [25, 27] | [22, 20] | [25, 17] | [23,19] | [26,16] | [25, 17] | [24,18] | [24,18] | [27, 15] |

Table 5.17 shows the average vehicle delay every 15 min under the four signal timing methods. The proposed fixed-time signal plan without minimum cycle length constraint produces

lowest delays comparing to other methods as highlighted in the table, followed by the proposed fixed-time signal plan with the minimum cycle length constraint. It probably due to two major reasons. First, the proposed method incorporates predictions that the cycle lengths are determined by future traffic conditions. It can better accommodate the arrival vehicles in the near future. Second, there are only two phases per cycle. Higher frequency to update phases contribute to dissipating the queueing vehicles waiting in front of the stop line more quickly without significantly sacrificing the overall level of service.

Table 5.17. Vehicle delays under various signal timing methods

| Methods | Time of Day | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 19:00 | 19:15 | 19:30 | 19:45 | 20:00 | 20:15 | 20:30 | 20:45 | 21:00 |
| 1. Real-world signal plans | 30.31 | 32.23 | 31.06 | 31.65 | 30.03 | 29.07 | 27.64 | 27.47 | 27.79 |
| 2. Actuated Signal plans | 25.14 | 27.62 | 24.61 | 23.37 | 23.79 | 19.52 | 21.79 | 20.54 | 19.06 |
| 3. Proposed Fix-time plans | 18.33 | 16.13 | 15.36 | 15.29 | 14.65 | 12.57 | 13.57 | 12.98 | 15.36 |
| 4. Proposed Fix-time with minimum C constraint | 18.33 | 17.80 | 18.08 | 17.18 | 17.11 | 17.10 | 17.24 | 17.42 | 17.58 |

## 5.5 SUMMARY

Movement-based traffic volume estimation and prediction at signalized intersections is challenging due to the various non-linear spatial relationships at different locations/movements and the complicated underlying temporal dependencies. This section developed a PLS method and a deep-learning based method, called Deep Intersection Spatial-Temporal Network (DISTN), for real-time movement-based traffic volume estimation and prediction respectively at signalized intersections. The fixed-time signal control method is then developed based on the short-term predicted traffic volumes.

PLS method extracts dominant information by the predictor and predicted components from traffic features and generate estimation results. It is an efficient and effective method for spatiotemporal real-time estimation tasks. This method is especially useful for signal timing

control when loop detector data are not available. The proposed method can be used instead to estimate traffic volumes using sampled mobile-sensing data.

DISTN consists of CNN and LSTM. In the CNN model, traffic volumes for all movements of one intersection at one-time step can be constructed as an OD matrix and considered as an "image." CNN was applied to learn the spatial dependencies by stacking multiple "images" of multiple intersections together and modeling the graphical similarities (i.e., spatial dependencies) of the movement-based traffic volumes. LSTM was applied after CNN to model the temporal features of traffic volume data. The within-day, daily, and weekly patterns of traffic volume were also considered by constructing CNN and LSTM for each pattern and fusing them together. The proposed model was evaluated using real-world data and simulation data. Experimental results showed that the proposed model outperforms several state-of-the-art methods for short-term traffic volume prediction in term of the MSE and the MAPE. The results also showed that (i) the prediction performance for the main streets is usually better than that for the minor streets due to the larger volumes on the main streets; (ii) the prediction performance for left turn and right turn movements are generally worse than that of the through movements due to the lower volume and random arrival vehicles at the turning movements; (iii) the prediction performance is worse for boundary intersections in the study area compared with those inside the area because either the upstream or downstream intersection information is missing for a boundary intersection; and (iv) the prediction performance is often worse when using real-world data compared with simulation data due to the naturally more uncertain traffic flow in the real world. One of the limitations of the proposed deep learning-based method is that the results are always difficult to interpret. The performance of the model is largely determined by the embedded spatial and temporal features of the collected traffic flow data. For future work, we will first collect more movement-based traffic

volume and further test the proposed model with various structures of the networks. Second, trajectory data from GPS devices can provide more valuable information on traffic conditions, e.g., average speed and a number of stops, etc. These measurements can also be utilized in the proposed model. For this, the DISTN structure provides a fusion framework to integrate additional data sources (e.g., trajectory data). This can be done by constructing a CNN and a LSTM for each additional data source and then fusing them together in DISTN. Third, for large traffic networks, local CNN may have limited ability to model the spatial features across the network. To address this issue, more powerful models may be investigated, e.g., the graph-convolutional networks.

# Chapter 6. TRAFFIC SIGNAL CONTROL UNDER MEDIUM-TO-HIGH PENETRATION RATES OF CVS

Total benefits of CVs will ultimately depend on the level of CV penetration. It is expected that by 2045, the CV fleet penetration may reach 95% in the market [81]. It will thus take a relatively long transition period to achieve full market penetration of CV. This chapter presents methods tot trajectories of the entire traffic population based on available CV trajectories that are just samples of the traffic flow, under medium penetration of CVs (roughly one CV per cycle). After the estimation, trajectories of all vehicles are available, for which the methods in Chapter 4 can be applied directly for signal timing optimization.

In the literature, many studies focused on modeling the traffic flow of CVs by integrating traffic flow simulation and network communication [82][83]. They calibrated parameters of the car-following behaviors for CVs, which describe the process of how following CVs replicate the driving behavior of its leading CV in the traffic stream [84]-[86]. These studies usually assumed a 100% penetration rate of CVs. Goodall [87] developed a technique to estimate the locations of the unequipped vehicle in an arterial given CV location. The method could roughly estimate the number of unequipped vehicles between two stopped CVs or between a predefined boundary (e.g., 50m in front of the stop line) and the stop line. The method, however, cannot reproduce trajectories of unequipped vehicles.

For the mixed traffic in a platoon, there are three car following behaviors:

1. Unequipped vehicle following CVs or unequipped vehicles.
2. CVs following CVs.
3. CVs following unequipped vehicles.

For the first case, the car following behaviors of unequipped vehicles are modeled by IDM, which is the same to Eq. (4.16) - (4.21) except that the parameters used in IDM for the unequipped vehicle are different from CVs. For the second case, the trajectories for both the leading and following vehicles are known. This chapter aims to solve the third case where the leading vehicle (unequipped vehicle) is unknown.

In this study, the Kalman filter is applied to reconstruct the short-term vehicle trajectories for the unequipped vehicles based on the sampled data from CVs under medium to a high penetration rate of CVs. Kalman filter is a well known approach for trajectory estimation [88]. Here, medium-to-high level represents the case where at least one CV exists in a vehicle platoon passing through an intersection, refer to Table 3.1. In addition, loop detector data are assumed to be available. Vehicles can be detected when they pass by the detector. The distance between each leading unequipped vehicle and the following CV pair in the same lane can be estimated. The trajectories of CVs are considered as measurements of the states with errors and trajectories of unequipped vehicles as hidden states. After estimating trajectories of the unequipped vehicle using the Kalman filter method, the trajectory-based signal control methods for a single intersection or a corridor in Chapter 4 can be applied to optimize traffic signal timings.

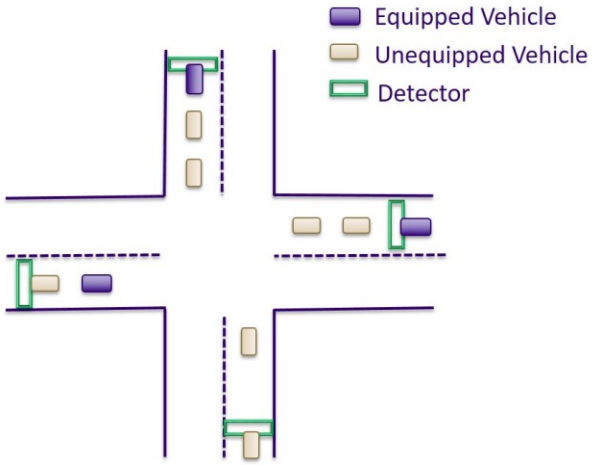



Figure 6.1. Trajectories of CV

## 6.1 Model Development

Here a simplified car following the model proposed by Newell [89] is applied to describe the state dynamics due to its simplicity in formulation and effectiveness in simulation. The CV is considered as the reference vehicle in a platoon while the states of the unequipped vehicles are estimated according to the simplified car-following model. In a single lane, the unequipped vehicles replicate the driving behaviors of CVs. The relationship between them is given by:

$$s_t^{n+1} = d_j + v_t^{n+1}\tau_{n+1} \tag{6.1}$$

where $s_t^{n+1}$ represents the spatial headway between the leading vehicle $n$ and the following vehicle $n+1$ at time *t*. The parameter $\tau_n$ denotes the reaction time of vehicle $n$ and $d_j$ denotes the jam traffic headway. It is assumed that $\tau_n$ and $d_n$ follow a bivariate normal distribution.

The variable $g_t$ and $v_t$ denote the location and speed of unequipped vehicles.

$$g_{t+1} = g_t + v_t\delta + w_{g,t} \tag{6.2}$$

$$v_{t+1} = v_t + w_{v,t} \tag{6.3}$$

where $w_{g,t}$ and $w_{v,t}$ denote the disturbance of position and speed. $\delta$ is time step size. The state variables $x_t = \begin{bmatrix} g_t \\ v_t \end{bmatrix}$. The model of states can be written as:

$$x_{t+1} = Ax_t + w_t \tag{6.4}$$

where $A = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix}$, $w_t = \begin{bmatrix} w_{g,t} \\ w_{v,t} \end{bmatrix}$ which follows normal distribution $N(0, Q)$ with zero mean and covariance $Q$. It is assumed that $w_t$ follow a normal distribution with zero mean and covariance $Q = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$. Thus, the linear model of measurements can be obtained as:

$$y_t = g_t + \tau\bar{v}_t - d + \gamma \tag{6.5}$$

where $\bar{v}_t$ represent the speed of the CV (following vehicle) and $\gamma$ is random noise.

Newell's simplified car following model assumes the speed relationship between the leading vehicle and the following vehicles as $\bar{v}_t = v_{t-\tau}$, which can be approximated by

$$\bar{v}_t = v_t - w_{v,t} \tag{6.6}$$

We can rewrite (6.5) as:

$$y_t = g_t - \tau - d + \gamma - \tau w_{v,t} \tag{6.7}$$

We further define a new variable $\lambda_t = \gamma + \tau w_{v,t}$, and then we have:

$$y_t = Cx_t + d + \lambda_t \,,\ \lambda_t \sim N(0, R) \tag{6.8}$$

where $C = \begin{bmatrix} 1 \\ -\tau \end{bmatrix}'$. Noted that the matrix A and C are known. There is no correlation between the location noise and speed noise in the update process. According to Zhu and Ukkusuri, the model given by (6.5) is identifiable if Q is diagonal.

The Kalman filter consists of two steps: the prediction step and update step [91]. In the prediction step, the prior distribution of $X_{t+1|t}$ and $X_{t|t}$ need to be computed. In the update step, the posterior distribution of $x_{t+1|t+1}$ is estimated. The algorithm can be summarized as follows.

1. **Initialize** model parameters: $\sigma_1^2$, $\sigma_2^2$, $d$, $\tau$, $R$.
2. **Predict:**

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t} \tag{6.9}$$

$$P_{t+1|t} = AP_{t|t}\mathrm{A}' + \mathrm{Q} \tag{6.10}$$

3. **Update**:

$$K_{t+1} = P_{t+1|t}C'(CP_{t+1|t}C' + R)^{-1} \tag{6.11}$$

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1}(y_{t+1} - C\hat{x}_{t+1|t} + d) \tag{6.12}$$

$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1}CP_{t+1|t} \tag{6.13}$$

where $K_{t+1}$ denotes the Kalman gain of Kalman Filter.

## 6.2 Numerical Results

Trajectories of CV and unequipped vehicles are collected from traffic simulation using IDM (with different parameters in IDM for CVs and unequipped vehicles). IDM utilizes multiple sensitivity parameters to model realistic driving behavior. It also incorporates a collision avoidance function. The most significant parameter in the Kalman filter is $d_f$, which is the traffic jam headway in congested traffic condition. Under the uncongested situation, $d_f$ denotes the distance between a vehicle pair. This value is retrieved from loop detector data.

One of the leader-follower pair has been selected to illustrate how the algorithm could be used for trajectory estimation of unequipped vehicles. Figure 6.2 and Figure 6.3 shows the updates of speed and location of both CV and unequipped vehicle over 60s. In Figure 6.2, the blue line indicates the speed profile of the leader (unequipped vehicle), which travels at free flow speed at the beginning and then decelerate to stop. The red dotted line represents the true speed of CV that generates from the simulation. Black dotted line is the speed of CV generated from the proposed Kalman Filter method. The results show that the following vehicle does not exactly follow the speed disturbance of its leader but well captures the speed dynamics compared with the true speed.

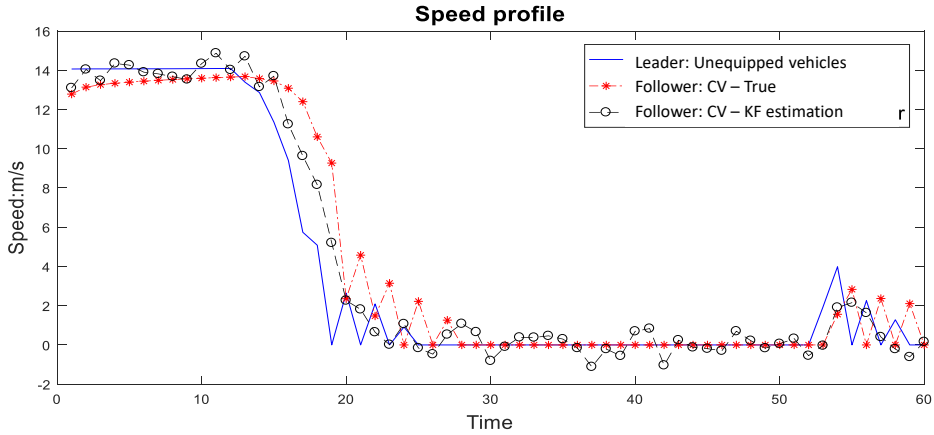



Figure 6.2. Speed profile of CV and NCV

Figure 6.3 shows the location profile of CV and unequipped vehicles. Before first 20s, the spacing between the leading vehicle and CV (estimated from Kalman Filter) is closer than the true value. It is because in the simulation or the real world, the driver needs to consider the speed limit while it is not accounted for in Kalman Filter estimation. Thus, the estimated speed for CV is larger than the true value, making its spacing to the leader smaller. Figure 6.4 shows the trajectories of another leader-follower pair. The same trend is identified as the previous pair.

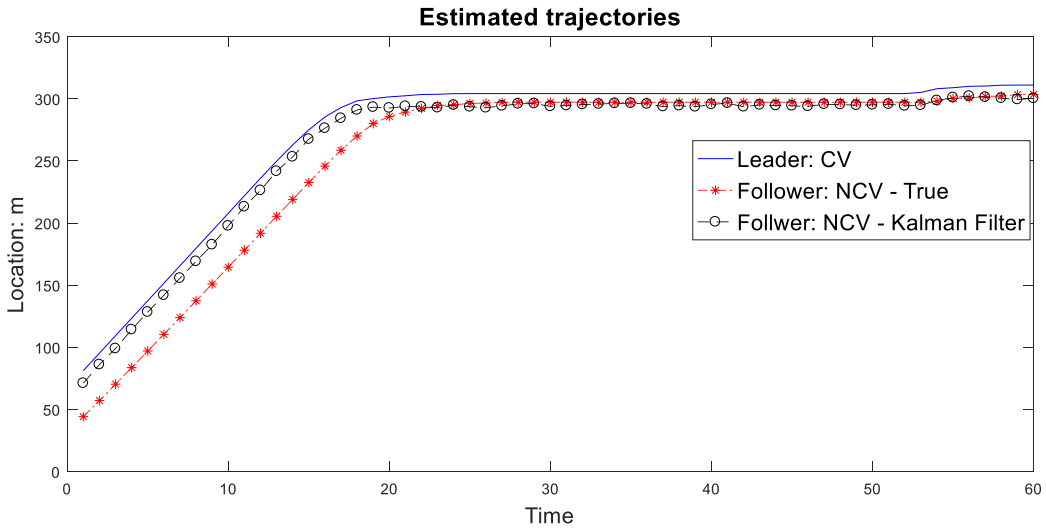



Figure 6.3. Location profile of CV and NCV

In order to test the algorithm, the Mean absolute error (MAE) is calculated to see the error of estimation by comparing the true trajectories of unequipped vehicles from simulation and Kalman filter estimation. It is observed that the MAE for speed states is relatively small while large for locations. It is mainly because that the relative spacing between the leader and the follower under free-flow speed in the Kalman filter is much smaller than the true value. The parameter $d$ in car following model is predefined as 10 as the initial value, which is the jam traffic headway. The MAE could be further improved if this parameter could be adjusted under different congestion levels. In the future study, we could incorporate this parameter into the Kalman filter estimation in order to compute the unknown states and model parameter simultaneously.

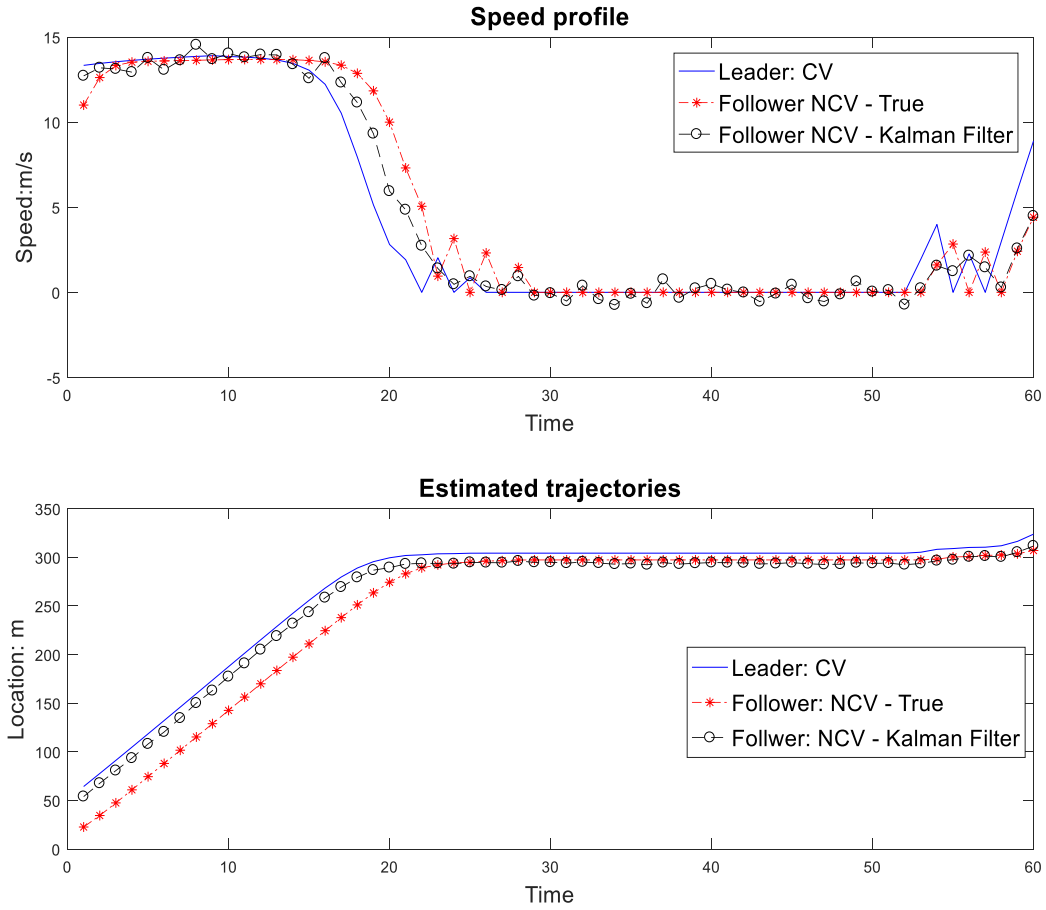



Figure 6.4. Speed and location profile of another CV and NCV pair

Table 6.1.Trajectories estimation results

| State | MAD |
|---|---|
| Speed (m/s) | 0.8 |
| Location (m) | 9 |

# Chapter 7. CONCLUSION

The advent and deployment of CV and V2X communications offer the potential to significantly improve the efficiency of traffic signal control systems. This study presented a modeling framework, solution techniques, and multiple numerical experiments for signal timing optimization problems under different penetration rates of CVs. Under full penetration of CVs, the proposed signal timing determination algorithm utilized trajectory information (speeds, locations, etc.) from CVs as the input to optimize the green times by considering vehicles' fuel consumption and travel time for both single intersections and corridors with multiple intersections. At the intersection level, the problem was first formulated as a MINLP by applying the IDM to predict vehicle trajectories. Such a formulation has a large dimension and a complex car-following model (the IDM). A DP formulation was then developed to approximate the MINLP. The overall problem was divided into stages (one stage for each signal phase). The objective was the summation of the objective of each stage. The objective function of a stage was approximated as a function of the state and decision variables of the stage only, by approximating the vehicle speeds and delays. It was showed that imposing the fixed cycle length constraint would invalidate the DP formulation. Then a two-step method was applied to address this issue. In the first step, an end-stage cost was added to the DP formulation, defined by how much the DP solution violates the fixed cycle length constraint. This step forced the DP to produce a solution with a cycle length that is close to the given fixed cycle length. The second step was a branch and bound method to further refine the DP results to obtain a solution that produces the given cycle length exactly. For a corridor that contains multiple intersections, the problem was also formulated in a centralized scheme as a MINLP considering the fixed cycle length constraint to optimize the phase durations and offsets in one mathematical program. IDM was applied to estimate and predict vehicle trajectories

considering 100% penetration rate of CV. Due to the complexity of the model, we decentralized the problem into two levels: an intersection level to generate optimal phase durations using the DP method and a corridor level to update the optimal offsets for all intersections. The two-level model reduced the complexity of the MINLP formulation. In order to solve the two-level model, we developed a prediction-based solution technique that can solve the problem iteratively. The performance of the algorithm was evaluated using data generated from traffic simulation. For a single intersection, results of the proposed DP model were compared with two other models. The first one was the traditional actuated signal timing plan generated by SYNCHRO. The second was to solve the MINLP formulation directly using the NOMAD solver in MATLAB. The results showed that the proposed DP method is always superior to SYNCHRO under all cases and can generate similar (slightly worse) solutions compared with NOMAD. However, NOMAD has difficulties finding optimal solutions when the number of variables is relatively large. This makes the proposed DP method more favorable when dealing with large problems (e.g., for multiple cycles). For a corridor, the results from MINLP and the two-level model both outperformed the signal optimization and coordination plan generated by SYNCHRO. This was tested for six cases that consider various combinations of traffic volumes and vehicle types. The results also suggested that signal coordination may bring limited benefits to intersections with low traffic volumes or to the vehicles on the minor street.

Under medium-to-high penetration rates of CVs, e.g., at least one CV in a vehicle platoon within a cycle, Kalman filter was applied to estimate and predict trajectories for unequipped vehicles given sampled CV trajectories. Specificity, a vehicle pair was considered in this study. There are three scenarios of the car following behaviors in a mixed traffic environment. If an equipped vehicle following a CV/unequipped vehicle, car following model IDM was applied to

calculate vehicle trajectories of the following vehicle (unequipped vehicle). If a CV following another CV, their trajectories are available due to the V2V technologies. Kalman filter technique was applied to address the challenge of the third scenario, which is to estimate and predict the trajectories of the leading unequipped vehicle given trajectories of CV (follower). Trajectory-based signal timing method in Chapter 4 was applied to calculate signal timing partakers based on the estimated trajectories from the Kalman filter and available CV trajectories.

Under relatively low penetration rates of CVs, trajectory-based traffic signal control method can hardly apply because it is really challenging to estimate trajectories for unequipped vehicles given limited data of CVs. Hence, the aggregated performance measurements, e.g., traffic volumes for each movement, can be estimated and predicted for signal timing optimization. Two cases were discussed in this research. If loop detector/cameral data were available, the traffic volume of each movement for a given intersection can be obtained directly. A deep learning approach was developed to predict short-term traffic volume, which was further utilized for traffic signal timing optimization. If the loop detector/camera data were not available, the partial least regression (PLS) method was applied to estimate traffic volumes first given sampled vehicle trajectory data. The proposed model was evaluated using real-world data and simulation data. Experimental results showed that the proposed model outperformed several state-of-the-art methods for short-term traffic volume estimation and prediction in term of the MSE and the MAPE. The results also showed that (i) the volume estimation and prediction performance for the main streets are usually better than that for the minor streets due to the larger volumes on the main streets; (ii) the estimation and prediction performance for left turn and right turn movements are generally worse than that of the through movements due to the lower volume and random arrival vehicles at the turning movements; (iii) the prediction performance is worse for boundary intersections in the

study area compared with those inside the area because either the upstream or downstream intersection information is missing for a boundary intersection; and (iv) the prediction performance is often worse when using real-world data compared with simulation data due to the naturally more uncertain traffic flow in the real world.

There are a few limitations of the current research. First, under full penetration rates of CVs, optimal signal timing parameters are calculated from a deterministic mathematical model (aims at minimizing total fuel consumption and travel time) using MINLP or DP. The proposed algorithms are suffered from the computational burden, especially for MINLP, so that it is hard to deploy in the real world. In addition, relationships between the CV trajectories and the signal states are affected by many other factors which are impossible to be captured completely by the proposed mathematical model. The complex underlying relationships can be learned from more advanced learning-based model, e.g., reinforcement learning-based traffic signal control methods [102]. Second, under relatively low penetration rates of CVs, a DISTN was applied to predict movement-based traffic volume in short-term. However, for large traffic networks, local CNN may have limited ability to model the spatial features across the network. To address this issue, more powerful models may be investigated, e.g., the graph-convolutional networks [103][104]. These topics will be investigated in future research.

# BIBLIOGRAPHY


[1] EIA. How Much Gasoline Does the United States Consumed. U.S. Energy Information Administration. (2016).

[2] Schrank, D., Eisele, B., Lomax, T., & Bak, J. 2015 urban mobility scorecard. Texas Transportation Institute, August. (2015).

[3] Sunkari, S., Songchitruksa, P., Charara, H., & Zeng, X. *Improved Intersection Operations During Detector Failures*. FHWA/TX-10/0-6029-1, Washington, DC: US Department of Transportation. (2009).

[4] Chen, Y., & Wang, J. Adaptive vehicle speed control with input injections for longitudinal motion independent road frictional condition estimation. *IEEE Transactions on Vehicular Technology*, *60*(3), (2011), pp. 839-848.

[5] Rhodes, A., Bullock, D., Sturdevant, J., Clark, Z., & Candey Jr, D. Evaluation of the accuracy of stop bar video vehicle detection at signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, (1925), (2005), pp. 134-145.

[6] John, D. The connected car data explosion: the challenges and opportunities. Information Age. (2015). https://www.information-age.com/tconnected-car-data-explosion-123473363/

[7] Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., & Bayen, A. M. (2010). Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, *18*(4), 568-583.

[8] Ban, X. J., & Gruteser, M. (2012, August). Towards fine-grained urban traffic knowledge extraction using mobile sensing. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing* (pp. 111-117). ACM.

[9] NHTSA. National Highway Traffic Safety Administration 49 CFR Part 571. Docket No. NHTSA-2016-0126. (2016).

[10] SAE, D. J2735 Dedicated Short Range Communications (DSRC) Message Set Dictionary. *Society of Automotive Engineers, DSRC Committee*. (2009).

[11] Jain, G., & Pai, M. Talking Cars: A survey of Protocos for Connected Vehicle Communication. Wipro Digital. (2018) https://wiprodigital.com/2018/06/20/talking-cars-a-survey-of-protocols-for-connected-vehicle-communication/

[12] USGAO. Vehicle-to Infrastructure Technologies Expected to Offer Benefits, but Deployment Challenges Exist. United States Covernment Accountability Offic. Retreived from: http://www.gao.gov/assets/680/672548.pdf. (2015).

[13] CITS. Cooperative Traffic System - safe and intelligent. Retreived from: http://c-its-korridor.de/?menuId=1&sp=en

[14] Roess, R. P., Prassas, E. S., & McShane, W. R. (2011). Traffic engineering.

[15] Sims, A. G. The Sydney coordinated adaptive traffic system. In *Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems, 1979, Pacific Grove, California, USA*. (1979).

[16] Hunt, P. B., Robertson, D. I., Bretherton, R. D., & Royle, M. C. The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering & Control*, *23*(4). (1982).

[17] Shelby, S. Single-intersection evaluation of real-time adaptive traffic signal control algorithms. *Transportation Research Record: Journal of the Transportation Research Board*, (1867), (2004), pp. 183-192.

[18] Gordon, R. L., Reiss, R. A., Haenel, H., Case, E. R., French, R. L., Mohaddes, A., & Wolcott, R. *TRAFFIC CONTROL SYSTEMS HANDBOOK-REVISED EDITION 1996* (No. FHWA-SA-95-032). (1996).

[19] Hoar, R., Penner, J., & Jacob, C. Evolutionary swarm traffic: if ant roads had traffic lights. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* (Vol. 2, pp. 1910-1915). IEEE. (2002).

[20] Datesh, J., Scherer, W. T., & Smith, B. L. (2011, June). Using k-means clustering to improve traffic signal efficacy in an IntelliDrive SM environment. In *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on* (pp. 122-127). IEEE.

[21] Liu, H. X., & Di, X. (2010). Development of Algorithms for Travel Time-Based Traffic Signal Timing, Phase I–A Hybrid Extended Kalman Filtering Approach for Traffic Density Estimation along Signalized Arterials.

[22] Smith, B. L., Venkatanarayana, R., Park, H., Goodall, N., Datesh, J., & Skerrit, C. IntelliDriveSM Traffic Signal Control Algorithms. *University of Virginia*. (2010).

[23] Mannion, P., Duggan, J., & Howley, E. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic Road Transport Support Systems* (pp. 47-66). Springer International Publishing. (2016).

[24] Goodall, N. J. Traffic Signal Control with Connected Vehicles. A dissertation. (2013).

[25] Henry, R. D. *Signal timing on a shoestring* (No. FHWA-HOP-07-006). (2005).

[26] Warberg, A., Larsen, J., & Jørgensen, R. M. *Green wave traffic optimization-a survey*. Informatics and Mathematical Modelling. (2008).

[27] Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beaird, S., Braud, C., ... & Urbanik, T. *Traffic signal timing manual* (No. FHWA-HOP-08-024). (2008).

[28] Lämmer, S., & Helbing, D. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(04), P04019.

[29] Henry, R. D. *Signal timing on a shoestring* (No. FHWA-HOP-07-006). (2005).

[30] Bowers, D. A. Progressive timing for traffic signals. *Proc. Inst. Traff. Engrs*, (1947)., PP. 93-100.

[31] Petterman, J. L. Timing progressive signal systems. *Traffic Engineering*, *29*, (1947), PP. 194-199.

[32] Davidson, B. M. Design of signal systems by graphical solutions. *Traffic Engineering*, *31*(2), (1960), PP. 32-4.

[33] Gartner, N. H., Assman, S. F., Lasaga, F., & Hou, D. L. A multi-band approach to arterial traffic signal optimization. *Transportation Research Part B: Methodological*, *25*(1), (1991), PP. 55-74.

[34] Messer, C. J. A. Report on the User's Manual for Progression Analysis and Signal System Evaluation Routine-Passer II Texas Transportation Institute. *Texas A & M University*. (1974).

[35] Coogan, S., Kim, E., Gomes, G., Arcak, M., & Varaiya, P. Offset optimization in signalized traffic networks via semidefinite relaxation. Transportation Research Part B: Methodological, 100, (2017), PP. 82-92.

[36] Hu, H. & Liu, H. X. Arterial offset optimization using archived high-resolution traffic signal data. *Transportation Research Part C: Emerging Technologies*, *37*, (2013). PP. 131-144.

[37] He, Q., Head, K. L., & Ding, J. PAMSCOD: Platoon-based arterial multi-modal signal control with online data. *Transportation Research Part C: Emerging Technologies*, *20*(1), (2012). PP. 164-184.

[38] Lee, J., Park, B., & Yun, I. Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. *Journal of Transportation Engineering*, *139*(10), (2013), PP. 1020-1029.

[39] Feng, Y., Head, K. L., Khoshmagham, S., & Zamanipour, M. A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, *55*, (2015), PP. 460-473.

[40] Beak, B., Head, K. L., & Feng, Y. Adaptive Coordination Based on Connected Vehicle Technology. *Transportation Research Record: Journal of the Transportation Research Board*, (2619), (2017), PP. 1-12.

[41] Li, W. & & Ban, X. J. (2018). Connected vehicle based traffic signal timing optimization. *IEEE transaction on Intelligent Transportation System*. DOI: 10.1109/TITS.2018.2883572

[42] Zhao, J., Li, W., Wang, J., & Ban, X. (2016). Dynamic Traffic Signal Timing Optimization Strategy Incorporating Various Vehicle Fuel Consumption Characteristics. *IEEE Transactions on Vehicular Technology*, *65*(6), 3874-3887.

[43] Xu, B., Ban, X. J., Bian, Y., Li, W., Wang, J., Li, S. E., & Li, K. (2018). Cooperative Method of Traffic Signal Optimization and Speed Control of Connected Vehicles at Isolated Intersections. *IEEE Transactions on Intelligent Transportation Systems*, (99).

[44] Li, Z., Chitturi, M. V., Zheng, D., Bill, A. R., & Noyce, D. A. Next-generation intersection control algorithm for autonomous vehicles. In *Transportation Research Board 92nd Annual Meeting* (No. 13-2185). (2013).

[45] Li, L., & Wang, F. Y. Cooperative driving at blind crossings using intervehicle communication. *IEEE Transactions on Vehicular Technology*, *55*(6), (2006), PP.1712-1724.

[46] Li, L., Wen, D., & Yao, D. A survey of traffic control with vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, *15*(1), (2014), PP. 425-432.

[47] Guo, Q., Li, L., & Ban, X. J. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*.

[48] Priemer, C., & Friedrich, B. A decentralized adaptive traffic signal control using V2I communication data. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on* (2009) pp. 1-6.

[49] Cai, C., Wang, Y., & Geers, G. Adaptive traffic signal control using vehicle-to-infrastructure communication: A technical note. In *Proceedings of the Second International Workshop on Computational Transportation Science* (2010). pp. 43-47. ACM.

[50] Li, J., & Qiu, T. Z. Improving Throughput of a Signalized Intersection in a Connected Vehicle Environment. *Present Transportation Research Board 96th Annual meeting*. (No. 17-02541). (2017).

[51] Islam, A., Bin, S. M. A., & Hajbabaie, A. Distributed coordinated signal timing optimization in connected transportation networks. *Transportation Research Part C: Emerging Technologies*, *80*. (2017).

[52] Sen, S., & Head, K. L. (1997). Controlled optimization of phases at an intersection. *Transportation science*, *31*(1), 5-17.

[53] Chen, S., Sun, D. J., Guan, S., & Guan, S. T. (2016). An Improved Adaptive Signal Control Method for Isolated Signalized Intersection. In *Transportation Research Board 95th Annual Meeting* (No. 16-6940).

[54] Wunderlich,k., Burnier, C., Larkin, J., Shah, V., and Vasudevan, M. (2008) Vehicle-Infrastructure Integration (VII) Probe Data Characteristics: Analyses of Three Field Data Sets. Final Report. Federal Highway Administration. Retrieved from: https://www.pcb.its.dot.gov/connected_vehicle/508/Library/Library%20-%20RRs%20-%20Tech%20-%20Apps%20-%20Traffic%20Management/VII%20DC%20Field%20Data%20FINAL.htm#toc62

[55] Ban, X. J., Hao, P., & Sun, Z. (2011). Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transportation Research Part C: Emerging Technologies*, *19*(6), 1133-1156.

[56] Hao, P., Ban, X. J., Guo, D., & Ji, Q. (2014). Cycle-by-cycle intersection queue length distribution estimation using sample travel times. *Transportation research part B: methodological*, *68*, 185-204.

[57] Ahmed, M. S., & Cook, A. R. (1979). Analysis of freeway traffic time-series data by using Box-Jenkins techniques (No. 722).

[58] Kamarianakis, Y., & Prastacos, P. (2003). Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record: Journal of the Transportation Research Board*, (1857), 74-84.

[59] Williams, B., Durvasula, P., & Brown, D. (1998). Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record: Journal of the Transportation Research Board*, (1644), 132-141.

[60] Lee, S., & Fambro, D. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, (1678), 179-188.

[61] Ojeda, L. R. L., Kibangou, A. Y., & De Wit, C. C. (2013). Adaptive Kalman filtering for multi-step-ahead traffic flow prediction. *In 2013 American Control Conference (ACC 2013)*.

[62] Guo, J., Huang, W., & Williams, B. M. (2014). Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, *43*, 50-64.

[63] Xie, Y., Zhang, Y., & Ye, Z. (2007). Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *22*(5), 326-334.

[64] Coogan, S., Flores, C., & Varaiya, P. (2017). Traffic predictive control from low-rank structure. *Transportation Research Part B: Methodological*, *97*, 1-22.

[65] Davis, G. A., & Nihan, N. L. (1991). Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, *117*(2), 178-188.

[66] Chang, H., Lee, Y., Yoon, B., & Baek, S. (2012). Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences. *IET intelligent transport systems*, *6*(3),

[67] Jeong, Y. S., Byon, Y. J., Castro-Neto, M. M., & Easa, S. M. (2013). Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *14*(4), 1700-1707. 292-305.

[68] Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, *7*(1), 124-132.

[69] Lv, Y., Y. Duan, W. Kang, Z. Li, and F. Wang. (2015). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, no. 2, pp. 865-873.

[70] Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., & Li, Z. (2018). Modeling Spatial-Temporal Dynamics for Traffic Prediction. *arXiv preprint arXiv:1803.01254*.

[71] Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, *90*, 166-180.

[72] Cui, Z., Ke, R., & Wang, Y. (2016). Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In *6th International Workshop on Urban Computing (UrbComp 2017)*.

[73] Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *79*, 1-17.

[74] Jia, Y., Wu, J., & Xu, M. (2017). Traffic flow prediction with rainfall impact using a deep learning method. *Journal of advanced transportation*, *2017*.

[75] Gordon, R. L., Reiss, R. A., Haenel, H., Case, E. R., French, R. L., Mohaddes, A., & Wolcott, R. *TRAFFIC CONTROL SYSTEMS HANDBOOK-REVISED EDITION 1996* (No. FHWA-SA-95-032). (1996).

[76] USDOT. Traffic Control Systems Handbook: Chapter 7. Local Controllers. Retrieve from: http://ops.fhwa.dot.gov/publications/fhwahop06006/chapter_7.htmk

[77] Koonce, P. (2008). *Traffic signal timing manual* (No. FHWA-HOP-08-024). United States. Federal Highway Administration.

[78] Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, *62*(2), 1805.

[79] Khondaker, B., & Kattan, L. (2015). Variable speed limit: A microscopic analysis in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, *58*, 146-159.

[80] Danczyk, A., & Liu, H. X. (2011). A mixed-integer linear program for optimizing sensor locations along freeway corridors. *Transportation Research Part B: Methodological*, *45*(1), 208-217.

[81] USDOT. Connected Vehicles: Benefits, Roles, Outcomes. (2019). https://www.its.dot.gov/research_areas/WhitePaper_connected_vehicle.htm

[82] Huang, H. Y., Luo, P. E., Li, M., Li, D., Li, X., Shu, W., & Wu, M. Y. (2007). Performance evaluation of SUVnet with real-time traffic data. *IEEE Transactions on Vehicular Technology*, *56*(6), 3381-3396.

[83] Lochert, C., Barthels, A., Cervantes, A., Mauve, M., & Caliskan, M. (2005, September). Multiple simulator interlinking environment for IVC. In *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks* (pp. 87-88). ACM.

[84] Ossen, S., & Hoogendoorn, S. (2005). Car-following behavior analysis from microscopic trajectory data. *Transportation Research Record: Journal of the Transportation Research Board*, (1934), 13-21.

[85] Kesting, A., & Treiber, M. (2008). Calibrating car-following models by using trajectory data: Methodological study. *Transportation Research Record: Journal of the Transportation Research Board*, (2088), 148-156.

[86] Chen, C., Li, L., Hu, J., & Geng, C. (2010, July). Calibration of MITSIM and IDM car-following model based on NGSIM trajectory datasets. In *Vehicular Electronics and Safety (ICVES), 2010 IEEE International Conference on* (pp. 48-53). IEEE.

[87] Goodall, N. J. (2013). *Real-time prediction of vehicle locations in a connected vehicle environment* (No. FHWA/VCTIR 14-R4).

[88] Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, *62*(2), 1805.

[89] Newell, G. F. (2002). A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, *36*(3), 195-205.

[90] Zhu, F., & Ukkusuri, S. V. (2017). An Optimal Estimation Approach for the Calibration of the Car-Following Behavior of Connected Vehicles in a Mixed Traffic Environment. *IEEE Transactions on Intelligent Transportation Systems*, *18*(2), 282-291.

[91] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35-45.

[92] Zheng, J., & Liu, H. X. (2017). Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies*, 79, 347-362.

[93] Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4), 818.

[94] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

[95] Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.

[96] Mishra, R. 2017. Concatenating Two Lists in Python. *Medium*. Accessed on Jan. 19, 2019. https://blog.usejournal.com/concatenating-two-lists-in-python-3cf9051da17f.

[97] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[98] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Kudlur, M. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI* (Vol. 16, pp. 265-283).

[99] Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, *90*, 166-180.

[100] Sanjeevi, M. (2018). DeepNLP—LSTM (Long Short Term Memory) Networks with Math. https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235.

[101] HCM 2010. (2010). Highway Capacity Manual. Chapter 31. Signalized Intersections: Supplemental. Transportation Research Board of the National Academies. Washington, DC.

[102] Wei, H., Zheng, G., Yao, H., & Li, Z. (2018, July). Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2496-2505). ACM.

[103] Henaff, M., Bruna, J., & LeCun, Y. (2015). Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

[104] Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2018). High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *arXiv preprint arXiv:1802.07007*.

[105] Li, W., & Ban, X. J. (2017, June). Traffic signal timing optimization in connected vehicles environment. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1330-1335). IEEE.

# VITA

Wan Li is a Ph.D. in transportation engineering at the University of Washington. She got her MS degree in transportation engineering from the Louisiana State University in 2014 and BS degree from the Sun Yat-sen University in 2012. Her research interests mainly focus on data-driven spatiotemporal forecasting, transportation big data analytics, traffic system modeling and simulation, and urban transportation network operation and control. In particular, she is interested in integrating data-driven methods, optimization algorithms, and big data analytics tools into intelligent transportation systems to explore how they could support the research and benefit the development of smart cities.