

**Short-term Link Travel Time Prediction and  
Intersection Priority Control for  
Urban Traffic Control and Management System**

**TANG Ruotian**



**Short-term Link Travel Time Prediction and  
Intersection Priority Control for  
Urban Traffic Control and Management System**

by

Tang Ruotian

Department of Civil Engineering

Nagoya University

Submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Engineering

Nagoya, Japan  
March 2020



## **ABSTRACT**

Traffic congestion has bothered people for a long time since motorization and urbanization took place in many cities around the world. Traffic congestion results from the imbalance between traffic demand and supply, so many governments used to construct more roads to meet the increasing traffic demand. However, it turned out to stimulate more congestion on the road. With the rapid development of computer and sensor technologies, decision-makers are trying to control and manage the traffic based on the intelligent transportation system (ITS) which shows great potential in reducing the congestion. To handle more and more complicated traffic conditions, many researchers devoted their efforts to expand traffic control and management from the real-time into the future and from highways to urban networks.

This study contributed to an ITS-based traffic control and management strategy which dynamically controls the priority at the intersection and employs short-term urban link travel time prediction so that it can catch up with the frequent change in the urban traffic condition. As an important component in modern traffic control and management systems, short-term travel time prediction models are widely studied using probe vehicle data, but the low penetration rate of probe vehicles limits their real-world applications in highways and urban arterials. Therefore, this study proposed non-parametric models to enhance the coverage rate of the prediction under the low data

penetration rate and expand the prediction horizon so that they can be applied on the whole urban network in the real world. When the study site changes from the highway to the urban network, the traffic condition becomes more complex. It is more doable to solve the congestion problem over a small scale where the traffic condition has a similar characteristic instead of reducing the congestion on the whole network. Therefore, the intersection which is the main potential bottleneck of the urban network attracts our attention. This study proposed a strategy to dynamically determine the priority of each incoming link at the intersection to reduce the congestion. Three main sections were employed to explain the details of this study.

Firstly, this study aimed to relieve the limitation of study sites caused by the low data penetration rate when predicting short-term travel time at urban networks. It was realized through a non-parametric model which is based on Bayes' theorem under a "prediction-resampling" structure. To predict the travel time for a link, most previous approaches used data only from vehicles on the target link. In contrast, the proposed model in this study considered data from vehicles in the crossing direction. With the data from both directions, the coverage rate of an application using the proposed model can be expanded, especially when the data penetration rate is low. Besides, the signal pattern was estimated through the utilization of relationships between vehicles in both directions. The proposed model was evaluated in a computer simulation to test its robustness and reliability under different data penetration rates. The results implied that

the proposed model has a high coverage rate and stable and acceptable performance at different penetration rates.

Secondly, this study further expanded the prediction horizon of the proposed model mentioned above, which can only make predictions within a signal cycle due to its short time interval and expensive computational cost. To make multistep predictions into a longer future, most research chose to aggregate probe data to obtain useful samples when the penetration rate is low. However, the aggregation of probe data can only provide a general description of the travel time, which cannot capture changes in travel time during a short time interval. To overcome this limitation, a non-parametric model using disaggregate probe data based on dynamic time warping (DTW) was developed. Besides, instead of estimating the signal pattern, data from the target link were separated into different signal phases by the data from the crossing direction. A classical k-nearest neighbor (KNN) model and a naïve model were compared with the proposed model. The models were tested using the same computer simulation data. Moreover, data from two real-world cases in Nagoya, Japan were also used to evaluate the models. The results showed that the proposed model outperforms the other two models under different data penetration rates because its utilization of vehicle data is more efficient.

Lastly, to find out a practical method which can reduce congestion on the urban network in the real world, attention is concentrated on small but easy-congested locations such as the intersection. This study combined the link transmission model and a local

linear programming formulation to dynamically determine the priority of each incoming link at each intersection on the urban network. This model provided a local optimum solution for the traffic congestion so it can be added to other approaches that have different global objectives. The proposed model and models using other intersection priority strategies are tested by simulation data. Results showed that the congestion level in the proposed model was lower than other models, while the travel cost remained similar.



## **ACKNOWLEDGEMENTS**

It has been five years since I started my study in Morikawa, Yamamoto & Miwa lab at Nagoya University. It is the most pleasant and meaningful experience in my life. On the occasion of completing my doctoral study, I would like to express my sincere appreciation to many people for their support and assistance.

Firstly, I would like to express my most sincere gratitude to my supervisor, Professor Toshiyuki Yamamoto, who provides continuous guidance and immense knowledge for my research. I am also inspired by his rigorous academic attitude and passionate spirit for research. I also would like to extend my sincere appreciation to my advisor, Associate Professor Ryo Kanamori, who is so patient that spent a lot of time discussing the details of my research.

Secondly, I would like to thank the dissertation committee members: Associate Professor Tomio Miwa, Associate Professor Miho Iryo, Professor Junji Kato, and Associate Professor Koji Suzuki, for insightful comments and valuable advice. With their suggestions, my dissertation was improved from various perspectives. Also, I would like to give my grateful thanks to other staffs in our lab, Professor Takayuki Morikawa, Associate Professor Hitomi Sato, Associate Professor Toshiyuki Nakamura, Lecturer Mutsumi Tashiro, Lecturer Meilan Jiang, Ms. Junko Tsuda, and Ms. Chiharu

Kikata, who always support both my study and daily life.

Thirdly, I would like to acknowledge my lab mates, Mingyang Hao, Yanyan Li, Xun Zheng, Lisha Wang, Jiahang He, Jiangbo Wang, Lanhang Ye, Shasha Liu, Zhiguang Liu, Yefang Zhou, Lei Gong, and Weiliang Zeng, for encouraging me when I had difficulties. I also would like to give special thanks to my old friends, Yifan Wu, Yicheng Guan, Yingrui Yang and Hengyang Lu for their long-distance company. In particular, I am grateful to Professor Ge Wu for recommending me to continue my studies in our lab after the graduation of undergraduate school.

Last but not least, I would like to acknowledge the Program for Leading Graduate Schools: Ph.D. Professional Toryumon, for sponsoring most of my study in Japan. I would like to give deep thanks to my families for their support and encouragement. I could never accomplish this work without them.

# TABLE OF CONTENTS

<b>ABSTRACT .....</b>	<b><u>I</u></b>
<b>ACKNOWLEDGEMENTS .....</b>	<b><u>V</u></b>
<b>TABLE OF CONTENTS.....</b>	<b><u>VII</u></b>
<b>LIST OF FIGURES .....</b>	<b><u>XI</u></b>
<b>LIST OF TABLES .....</b>	<b><u>XII</u></b>
<b>ABBREVIATIONS AND ACRONYMS.....</b>	<b><u>XIII</u></b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b><u>1</u></b>
<b>1.1 BACKGROUND.....</b>	<b><u>1</u></b>
<b>1.1.1 TRAFFIC CONGESTION .....</b>	<b><u>1</u></b>
<b>1.1.2 ITS-BASED TRAFFIC CONTROL AND MANAGEMENT .....</b>	<b><u>2</u></b>
<b>1.1.3 ITS SOLUTION FOR REDUCING CONGESTION ON THE URBAN NETWORK .....</b>	<b><u>2</u></b>
<b>1.2 PROBLEM STATEMENT.....</b>	<b><u>4</u></b>
<b>1.2.1 LOW PENETRATION RATE PROBLEM IN SHORT-TERM TRAVEL TIME PREDICTION.....</b>	<b><u>4</u></b>
<b>1.2.2 DRAWBACKS FOR AGGREGATING DATA IN SHORT-TERM TRAVEL TIME PREDICTION .....</b>	<b><u>6</u></b>
<b>1.2.3 DIFFICULTIES OF SODTA APPROACHES FOR REDUCING THE URBAN CONGESTION .....</b>	<b><u>7</u></b>
<b>1.3 RESEARCH OBJECTIVES .....</b>	<b><u>8</u></b>
<b>1.4 RESEARCH STRUCTURE.....</b>	<b><u>10</u></b>
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b><u>12</u></b>
<b>2.1 TRAVEL TIME PREDICTION MODELS.....</b>	<b><u>12</u></b>
<b>2.2 EXTENSIONS OF TRAVEL TIME PREDICTION TO URBAN NETWORKS .....</b>	<b><u>15</u></b>
<b>2.3 LINK TRANSMISSION MODEL AND ITS APPLICATIONS .....</b>	<b><u>17</u></b>
<b>CHAPTER 3 CONSIDERING VEHICLES IN THE CROSSING DIRECTION TO ENHANCE THE COVERAGE RATE OF PREDICTION UNDER LOW PENETRATION RATE .....</b>	<b><u>21</u></b>
<b>3.1 INTRODUCTION.....</b>	<b><u>21</u></b>
<b>3.2 METHODOLOGY .....</b>	<b><u>22</u></b>
<b>3.2.1 DESCRIPTIONS OF THE PROPOSED MODEL .....</b>	<b><u>22</u></b>
<b>3.2.2 DETAILS OF THE PROPOSED MODEL.....</b>	<b><u>25</u></b>
<b>3.3 EXPERIMENTS .....</b>	<b><u>32</u></b>

3.3.1 DATA DESCRIPTION .....	32
3.3.2 RELATIONSHIPS BETWEEN INDIVIDUAL VEHICLES .....	33
3.3.3 MODELS FOR COMPARISON .....	36
3.3.4 EXPERIMENT SETTING .....	39
3.3.5 RESULTS AND DISCUSSION .....	41
3.4 SUMMARY .....	47
<b><u>CHAPTER 4 APPLYING DYNAMIC TIME WARPING TO EXPAND THE PREDICTION HORIZON</u></b>	
<b><u>USING DISAGGREGATE PROBE DATA .....</u></b>	<b>48</b>
4.1 INTRODUCTION.....	48
4.2 METHODOLOGY .....	49
4.2.1 SAMPLE SELECTION .....	49
4.2.2 DTW ALGORITHM IN THE PROPOSED MODEL .....	51
4.2.3 PREDICTION .....	55
4.3 EXPERIMENTS .....	57
4.3.1 DATA DESCRIPTION .....	57
4.3.2 PARAMETER CALIBRATION .....	60
4.3.3 RESULTS AND DISCUSSION .....	64
4.4 SUMMARY .....	69
<b><u>CHAPTER 5 INTERSECTION PRIORITY CONTROL FOR URBAN TRAFFIC MANAGEMENT TO</u></b>	
<b><u>REDUCE TRAFFIC CONGESTION.....</u></b>	<b>70</b>
5.1 INTRODUCTION.....	70
5.2 METHODOLOGY .....	71
5.2.1 CLASSIC LTM.....	71
5.2.2 LOCAL LINEAR FORMULATION TO REDUCE THE CONGESTION .....	75
5.3 EXPERIMENTS .....	77
5.3.1 SIMULATION AT A SIMPLE GRID NETWORK .....	78
5.3.2 SIMULATION AT THE SIOUX FALLS NETWORK .....	81
5.4 SUMMARY .....	90
<b><u>CHAPTER 6 CONCLUSIONS AND FUTURE WORK .....</u></b>	<b>91</b>
6.1 CONCLUSIONS .....	91
6.2 FUTURE WORK.....	94
<b><u>APPENDIX A ALGORITHM FOR SIGNAL TIMING ESTIMATION .....</u></b>	<b>96</b>
<b><u>APPENDIX B CHANGES IN THE PRIORITY IN THE PROPOSED MODEL AND THE DEMAND</u></b>	
<b><u>MODEL.....</u></b>	<b>98</b>

**APPENDIX C SIMULATION OF TIDAL TRAFFIC ON THE SIOUX FALLS NETWORK..... 101**

**BIBLIOGRAPHY ..... 103**



# LIST OF FIGURES

FIGURE 1.1 STRUCTURE OF THE DISSERTATION.....	11
FIGURE 3.1 VEHICLES AT A REGULAR INTERSECTION (WHERE THE PENETRATION RATE IS 50%) .....	23
FIGURE 3.2 FRAMEWORK OF THE PROPOSED MODEL: (A) MODEL STRUCTURE; (B) EXAMPLE OF USING THE DATA FOR PREDICTION.....	25
FIGURE 3.3 DISTRIBUTION OF SIMULATION TRAVEL TIMES.....	33
FIGURE 3.4 RELATIONSHIP BETWEEN THE TRAVEL TIMES OF TWO OBJECT VEHICLES: (A) LTD WITHIN 60 s; (B) LTD MORE THAN 60 s .....	34
FIGURE 3.5 RELATIONSHIP BETWEEN OBJECT VEHICLES AND CROSSING VEHICLES .....	35
FIGURE 3.6 MODELS USED IN THIS STUDY: (A) KNN-BASED MODEL; (B) PF-BASED MODEL; (C) PROPOSED MODEL.....	38
FIGURE 3.7 COVERAGE RATE UNDER DIFFERENT PENETRATION RATES .....	42
FIGURE 3.8 PREDICTIONS OF THE PROPOSED MODEL UNDER DIFFERENT PENETRATION RATES: (A) TIME POINTS FROM 10 TO 70; (B) TIME POINTS FROM 248 TO 308.....	46
FIGURE 4.1 EXAMPLES OF SAMPLES WITH TWO SIGNAL CYCLES.....	50
FIGURE 4.2 ILLUSTRATIONS OF A WARPING PATH .....	52
FIGURE 4.3 CHANGES IN TRAVEL TIME ON THE TARGET LINK IN THE COMPUTER SIMULATION .....	58
FIGURE 4.4 CHANGES IN AVERAGE NUMBER AND TRAVEL TIME OF TAXIS GOING STRAIGHT DURING ONE DAY.....	60
FIGURE 4.5 MAPE OF THE KNN MODEL WITH DIFFERENT COMBINATIONS OF REFERENCE SECTION LENGTH AND K UNDER DIFFERENT PENETRATION RATES .....	62
FIGURE 4.6 MAPE OF THE PROPOSED MODEL WITH DIFFERENT REFERENCE SECTION LENGTHS UNDER DIFFERENT PENETRATION RATES .....	63
FIGURE 4.7 ACCURACY OF DIFFERENT MODELS UNDER DIFFERENT PENETRATION RATES IN THE SIMULATION.....	65
FIGURE 4.8 RESULTS OF THE RESIDUAL ANALYSIS.....	68
FIGURE 5.1 THE SIMPLE GRID NETWORK .....	79
FIGURE 5.2 ACI AND TTT FOR MODELS UNDER DIFFERENT UPDATE TIME INTERVALS .....	80
FIGURE 5.3 THE SIOUX FALLS NETWORK.....	83
FIGURE 5.4 CHANGES IN THE PRIORITY IN THE PROPOSED MODEL AND THE DEMAND MODEL.....	86
FIGURE 5.5 THE DIFFERENCE OF CUMULATIVE TRAFFIC FLOW BETWEEN THE PROPOSED MODEL AND THE DEMAND MODEL.....	88
FIGURE B.1 CHANGES IN THE PRIORITY IN THE PROPOSED MODEL AND THE DEMAND MODEL .....	100
FIGURE C.1 FIRST 10 LINKS SORTED BY THE DIFFERENCE OF CUMULATIVE TRAFFIC FLOW BETWEEN THE PROPOSED MODEL AND THE DEMAND MODEL .....	102

## LIST OF TABLES

TABLE 3.1 DEFINITION OF TERMS RELATED TO VEHICLES .....	23
TABLE 3.2 ALGORITHM I: PROPOSED TRAVEL TIME PREDICTION PROCESS .....	28
TABLE 3.3 ALGORITHM II: RESAMPLING .....	30
TABLE 3.4 ACCURACY FOR THE PROPOSED MODEL UNDER DIFFERENT PENETRATION RATES .....	43
TABLE 3.5 STUDIES ON THE IMPACT OF THE CAR SHARING .....	44
TABLE 4.1 MODIFIED CLASSIC DTW ALGORITHM .....	54
TABLE 4.2 EXAMPLE OF SEARCHING THE SHORTEST WARPING PATH IN THE LOCAL COST MATRIX .....	55
TABLE 4.3 SAMPLE SIZE AND COMPOSITION .....	66
TABLE 4.4 ACCURACY OF DIFFERENT MODELS IN LINK 1 .....	67
TABLE 4.5 ACCURACY OF DIFFERENT MODELS IN LINK 2 .....	67
TABLE 5.1 ALGORITHM OF UPDATING THE CUMULATIVE NUMBER OF VEHICLES.....	72
TABLE 5.2 MODELS FOR COMPARISON.....	78
TABLE 5.3 ATTRIBUTES OF THE SIMPLE GRID NETWORK. ....	79
TABLE 5.4 RESULTS OF THE T-TEST BETWEEN THE PROPOSED MODEL AND OTHER LTM-BASED MODELS. .....	80
TABLE 5.5 ATTRIBUTES OF THE SIOUX FALLS NETWORK.....	82
TABLE 5.6 RESULTS OF THE SIMULATION ON THE SIOUX FALLS NETWORK. ....	84
TABLE A.1 ALGORITHM A-I: SIGNAL TIMING ESTIMATION .....	96
TABLE C.1 RESULTS OF THE SIMULATION OF TIDAL TRAFFIC ON THE SIOUX FALLS NETWORK.....	101



## ABBREVIATIONS AND ACRONYMS

<b>ACI</b>	Average congestion index
<b>ARIMA</b>	Autoregressive integrated moving average
<b>CTM</b>	Cell transmission model
<b>DNL</b>	Dynamic network loading
<b>DTA</b>	Dynamic traffic assignment
<b>DTW</b>	Dynamic time warping
<b>ETD</b>	Exit time difference
<b>FIFO</b>	First in first out
<b>GLTM</b>	General link transmission model
<b>ITS</b>	Intelligent transportation system
<b>KF</b>	Kalman filter
<b>KNN</b>	K-nearest neighbor
<b>KWT</b>	Kinematic waves theory
<b>LSTM</b>	Long short-term memory
<b>LTD</b>	Leaving time difference
<b>LTM</b>	Link transmission model
<b>MAPE</b>	Mean absolute percentage error
<b>MSA</b>	Method of successive average
<b>NN</b>	Neural network
<b>PF</b>	Particle filtering
<b>PPCA</b>	Probabilistic principal component analysis
<b>RMSE</b>	Root mean squared error
<b>SIR</b>	Sampling importance resampling
<b>SODTA</b>	System optimum dynamic traffic assignment

<b>SSNN</b>	State-space neural network
<b>STARIMA</b>	Space-time autoregressive integrated moving average
<b>STDNN</b>	Space-time delay neural network
<b>TTT</b>	Total travel time

# **CHAPTER 1 Introduction**

## **1.1 Background**

### **1.1.1 Traffic congestion**

With the rapid urbanization and motorization, traffic congestion has been an increasing social problem in both developed and developing countries. Traffic congestion does not only bring unpleasant experience to individuals but also cause actual financial loss and excess fuel energy consumption (Jayasooriya and Bandara, 2017). Transportation is recognized as one of the main sources for greenhouse gas emission and vehicles produce more emissions in the congestion (Barth and Boriboonsomsin, 2008). For example, congestion in the U.S. has kept growing since 1982 regardless of the city size (Texas Transportation Institute and INRIX, 2015). In 2014, 6.9 billion hours of extra time and 3.1 billion gallons of fuel were wasted due to the congestion in the 471 U.S. urban areas, and the corresponding economic cost to the average auto commuter was \$ 960. Most developing countries experiencing urbanization and motorization are also facing the same problem which developed countries already have. Cities in developing countries are more vulnerable to congestion problems than cities in developed countries due to the higher urban densities but less public transport (Barte, 2000). Traffic congestion results from the imbalance between traffic demand and supply, so many governments used to solve the congestion problem by constructing more roads (Hook and Replogle, 1996). However, the expansion of road capacity inversely stimulates additional

traffic demand, so attention has changed from infrastructure construction to traffic control and management.

### **1.1.2 ITS-based traffic control and management**

Traffic control and management usually focus on the demand side to reduce congestion. For example, one of the most widely used traffic management strategies is road pricing which is aimed to restrict the use of private cars. According to the report by Tri-State Transportation Campaign (2018), road pricing has successfully reduced the congestion and improved air quality in London, Stockholm, and Singapore. However, Richards (2006) argued that pricing policy like congestion charge might have a negative effect on the economy of the target area and the burden of the extra pricing might finally fall on people living in the target area. With the development of computer science and data collection technology, the intelligent transportation system (ITS) solution for congestion is drawing increasing attention in the field of traffic control and management. Instead of reducing people's travel demand, the ITS-based traffic control and management focus on optimizing the utilization of the transportation resources. For example, the introduction of the electronic toll collection system can significantly improve vehicle throughput on highways; drivers can avoid traffic congestion on their route based on the traffic information provided by the ITS.

### **1.1.3 ITS solution for reducing congestion on the urban network**

To support the rapid urbanization and motorization, the research interest of the ITS-based traffic control and management is switching from highways to urban roads. Traffic condition

on the urban network is more complex than the highway, so various pieces of traffic information are needed to describe the traffic condition. Among all types of traffic information, travel time plays the most important role because it is the most sensitive and easy-understanding information to individual drivers. Since traffic conditions on the urban network change frequently and there is always a time lag between systems and users, especially for the ITS applying pull-based strategies, the real-time information cannot always reflect the current traffic condition (Xu et al., 2018). Therefore, short-term travel time prediction, which makes predictions from several seconds to several hours into the future, has become increasingly crucial for the ITS-based traffic control and management (Vlahogianni et al., 2014).

On the other hand, many researchers focused on solving the congestion problem or other traffic-related problems using the dynamic traffic assignment (DTA) over the whole network. However, since urban traffic condition differs according to the location on the network, it is better to control and manage the traffic over a small scale where the traffic condition has a similar characteristic. In the real world, it is more doable to solve the congestion problem over a small scale instead of solving the optimization problem over the whole network by the DTA approach. Another advantage of focusing on the small scale is the future traffic information can be predicted more accurately. Consequently, the intersection which is the main potential bottleneck on the urban network attracts our attention. To reduce the traffic congestion on a large urban network, it is necessary to develop an ITS-based traffic control and management strategy that manages the traffic flow at the intersection. The control and management

strategy on each intersection will affect drivers' behaviors and further improve the traffic condition on the whole network. It is also worthwhile adding short-term travel time prediction to the ITS-based traffic control and management so that it can react to the complex and changeable traffic condition on the urban network.

## **1.2 Problem Statement**

### **1.2.1 Low penetration rate problem in short-term travel time prediction**

Traffic conditions are so complicated and unstable that large amounts of data are required to catch up with the changes in travel time. For research which utilizes probe data, the amount of data collected relies on two variables: the penetration rate of probe vehicles and their frequency (Bucknell and Herrera, 2014). The data frequency can be increased easily as long as the computer is able to handle and store these data, whereas it is difficult to increase the penetration rate because the public is reluctant to change their vehicles into probe vehicles. Lu et al. (2018) pointed out that a high frequency requires the penetration rate at a relatively high level, so the influence of penetration rate on the short-term travel time prediction should be fully considered. Most approaches to short-term prediction using probe data examined highways or urban arterials (Chen and Rakha, 2014; Guo et al., 2014, Habtemichael and Cetin, 2016), but they did not consider the low penetration rate because traffic on highways or urban arterials is relatively high. Argote-Cabañero et al. (2015) emphasized that higher traffic results in higher variability so it might be an increased need for higher penetration

rates.

On the other hand, there is an increasing motivation to expand the forecasting scale to urban networks where the penetration rate is low. On the urban network, most probe data are collected from probe vehicles (e.g. taxis and buses) which only account for part of vehicles. Abundant data can be collected to build up the historical database for the travel time prediction as long as the collection period is long enough. However, in real-time situations, if the penetration rate is low, which means there are not enough probe vehicles on the link during a time period, even though there are normal vehicles going through the link, no successive predictions can be made during that time period. Even though each vehicle might be equipped with onboard GPS units in the future, most researchers agreed that the penetration rate of vehicles sending probe data to the transportation system is still limited due to the privacy issue, and the cost for data processing and storage (Alrukaibi et al., 2018; Bellavista et al., 2018; Cheu et al., 2002; Jenelius and Koutsopoulos, 2013; Sanaullah et al., 2016; Srinivasan and Jovanis, 1996; Wan et al., 2016). Therefore, a part of the literature has focused on determining the optimal penetration rate for providing accurate traffic information (Lu et al., 2018; Cheu et al., 2002; Srinivasan and Jovanis, 1996). To obtain reliable traffic information, the optimal penetration rate ranges from 1% to 60% because it depends on network characteristics and estimation methods. Even though there might be an agreement on the best penetration rate of probe vehicles, it seems far from real-world cases of practical application because it is impractical to assume that probe vehicles are distributed uniformly on the network (Bellavista et al., 2018).

### **1.2.2 Drawbacks for aggregating data in short-term travel time prediction**

Probe data usually distribute unevenly over time, so many researchers tend to aggregate the raw data to obtain useful samples when the prediction horizon is long. Fusco and Gori (1995) pointed out that a 5-min interval is required in the advanced traveler information system. Therefore, the data are usually aggregated in 5 min in many aspects of research (Cai et al., 2016; Chen and Rakha, 2014; Elhenawy et al., 2014; Xia et al., 2011). However, some studies have adopted a longer time interval, such as 15 min (Guo et al., 2014; Habtemichael and Cetin, 2016; Polson and Sokolov, 2017).

Although the aggregation of data is necessary, especially for multistep prediction, there are two drawbacks for application on urban networks. First, forecasting results are also aggregated. Because the urban traffic condition changes quickly, forecasting results that are aggregated by minutes cannot react to changes in traffic conditions that occur in seconds. Second, the influence of the traffic signal cannot be reflected properly. The variation of travel time in urban networks comes mainly from the traffic signal. Thus, because the length of most aggregation time intervals that have been adopted is longer than the length of a signal cycle, the aggregation might reduce the variation of travel time. The reduction of this variation is good for the model, but the temporal change of the traffic condition could be missed. Although some aggregate models can provide travel time distributions for signalized sections of the roadway, their assumptions about distributions might vary according to the study site (Cao et al., 2014). Such distributions can provide a general description of travel time during the time interval, but they cannot reflect temporal changes in a short period.



### **1.2.3 Difficulties of SODTA approaches for reducing the urban congestion**

Most research at urban networks devoted effort to reducing the urban traffic congestion by solving the system optimum dynamic traffic assignment (SODTA) problem. The SODTA usually employs a linear programming formulation that aims to minimize the total travel time. With the expansion of the urban network scale, the computational cost is increasing. Since the traffic condition differs according to the locations on the urban network and changes frequently, the global solution for the urban congestion based on the SODTA may be difficult to catch up with the change in the current traffic condition. When more factors are taken into consideration to reproduce the real-world situation, the problem may become nonlinear. It is more difficult and costly to solve the nonlinear programming problem and sometimes the only approximate global solution can be achieved. Even though the SODTA can be realized in time, most drivers might not follow the routes resulting from the SODTA in the real world because they might not be the best choice for individual drivers. Moreover, there is an increasing tendency to directly consider other congestion-related issues like environmental sustainability instead of minimizing the total travel time on the network (Wang et al., 2018). Although different combinations of the SODTA model and the linear programming formulation can adequately solve various traffic-related problems, one formulation can only solve one problem because only one objective function can be used for one global optimization. Therefore, it is sometimes even counterintuitive that the SODTA is realized when the road is congested (Bruechner, 2011).

### **1.3 Research Objectives**

As stated already, the ITS-based traffic control and management has been widely recognized as an effective solution for traffic congestion but there are still limitations when applying to urban networks in the real world. To reduce the urban traffic congestion, this study contributes to ITS-based traffic control and management strategy which dynamically controls the priority at the intersection and employs short-term urban link travel time prediction so that it can catch up with the frequent change in the urban traffic condition. This objective is mainly realized by solving the three problems mentioned in the last section.

Firstly, the penetration rate of probe vehicles influences both coverage rate and accuracy of travel time prediction at urban networks, but most researches only focused on the accuracy (Srinivasan and Jovanis, 1996). Although there is no agreement on the optimal penetration rate, the low penetration rate situation is expected to exist for a long time, which will restrict the application of the ITS. This study aims to enhance the coverage rate and maintain accuracy when predicting the travel time at urban networks under the low penetration rate.

Secondly, there is an increasing requirement of timely and reliable traffic information for applications such as online personal car navigation and driverless vehicle. Current ITS usually extracts information after aggregating the data, especially when making predictions, but these pieces of information cannot describe the frequent changes in traffic conditions. This study contributes to utilizing disaggregate data to provide future information that can reflect the variations in travel time at urban networks.

Thirdly, to reduce the computational burden for solving the global optimization problem of congestion so that real-world applications can be realized, it is necessary to split the whole network into several parts to solve the urban congestion problem. Although there are researchers focusing on reducing the congestion at the urban intersection by managing traffic signals (García-Nieto et al., 2012; Park et al., 2000; Wiering, 2000), it is difficult to apply these methods to wide networks because they are restricted by the situation of their study sites. This study aims to reduce the congestion at each intersection on the urban network by managing the traffic flow of each incoming link. Another advantage for solving the congestion at an intersection scale is that it is more flexible to combine with other global solutions that have different goals.

To catch up with the frequent changes in urban traffic conditions, the predicted travel time is used to reflect the traffic condition in a short future. Based on the predicted travel time, the movement of the vehicle on the whole network can be simulated by a dynamic network loading (DNL) model. Among all DNL models, the link transmission model (LTM) is the most computationally efficient one that can replicate traffic flows and reflect relationships between links and nodes. LTM focuses on the updating of the number of vehicles passing through the endpoints of each link so the traffic flow control at the intersection which is proposed in this study can be integrated to reduce the congestion over the whole network based on the LTM. Moreover, the advantage of the LTM in computational ability makes it possible to apply in the real world where a large amount of calculation is required.

In addition, many researchers focused on route travel time, but Shi et al. (2017) argued that

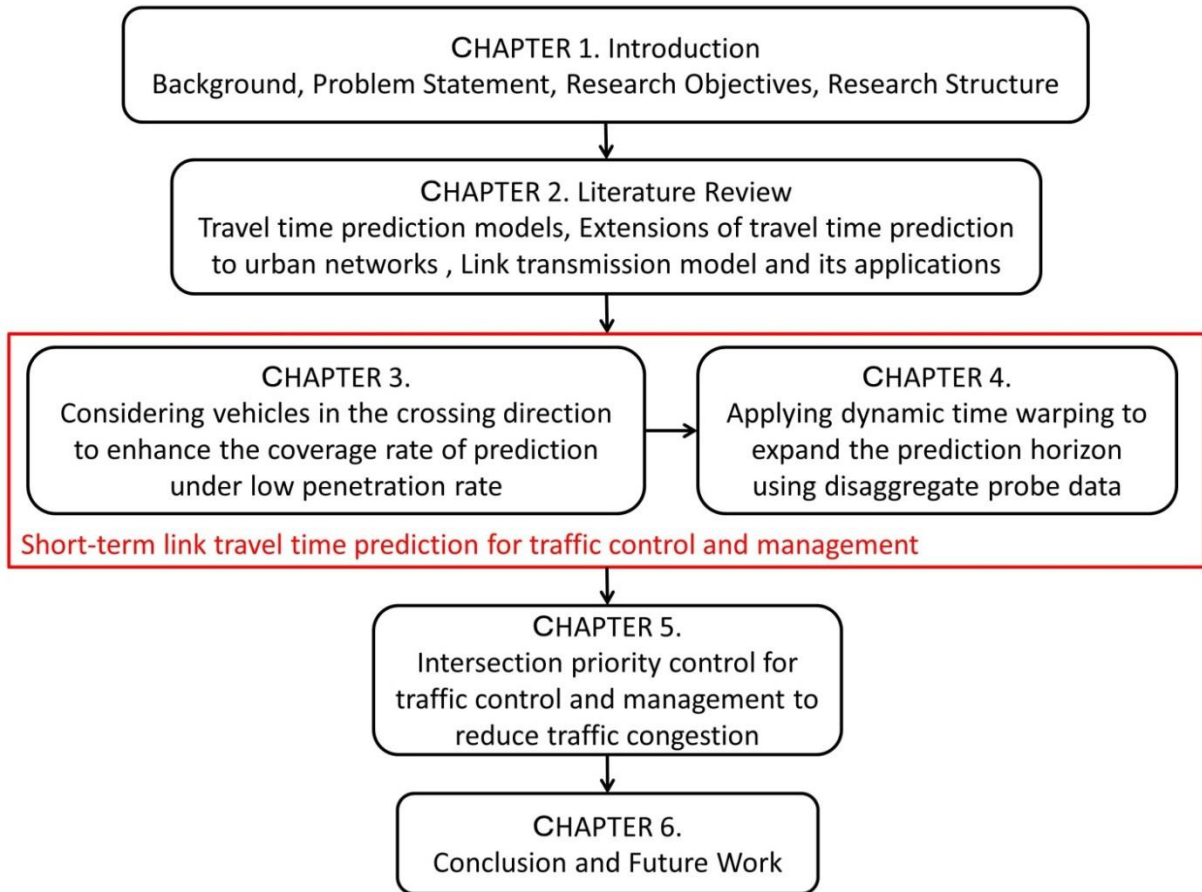
the route travel time can be formulated as the sum of the link travel times. Using this formulation, the distinct travel time delays due to traffic signals and different turning behaviors can be well captured. It is also more flexible to use the link travel time because it is difficult to predict the travel demand and the routes that travelers may take. Therefore, this study concentrated on the link travel time rather than the route travel time.

## **1.4 Research Structure**

The research structure of this study is shown in Figure 1.1. The dissertation consists of six chapters. In Chapter 1, the traffic congestion problems around the world and the ITS-based traffic control and management for the congestion reduction were introduced. Problems for extending the ITS-based traffic control and management to urban networks and the corresponding objectives of this study were pointed out. In Chapter 2, three sections were employed to review the related work about travel time prediction models, extensions of travel time prediction to urban networks, and the LTM and its applications. Chapter 3, 4 explained the details of relieving the limitations on the short-term urban link travel time prediction when extending to the urban network. This was realized by considering vehicles in the crossing direction to enhance the coverage rate of short-term travel time prediction and applying Dynamic Time Warping (DTW) to expand the prediction horizon based on disaggregate probe data. To reduce traffic congestion, Chapter 5 proposed a method which controls the intersection priority. The short-term travel time prediction can be combined with the priority control at the

intersection so that it can react to the frequent change in traffic conditions on the urban network.

Finally, conclusions and future work of this study were given in Chapter 6.



**Figure 1.1 Structure of the dissertation**

## **CHAPTER 2 Literature Review**

### **2.1 Travel Time Prediction Models**

To date, many models have been developed to predict travel time. They can be categorized into four types, namely, the naïve model, the traffic flow-based model, the data-driven model, and the hybrid model (Mori et al., 2015). Naïve models use the average value of the instantaneous or historical data as the prediction. Despite their low accuracy, they are employed widely due to their low computation cost and convenient implementation. They are also used as a benchmark for comparison with other complex models in academic research. Traffic flow-based models include macroscopic, microscopic, and mesoscopic simulation models (Ben-Akiva et al., 2001). They can provide detailed information, such as the delay and the queue length, for decision-making based on traffic theory. However, the real traffic condition is so complex that it is difficult to make a general simulation. Unlike traffic flow-based models, data-driven models perform better under complex and dynamic traffic conditions because there is no restricted theory. Although there is a lack of interpretability, these models can make considerable predictions as long as there are enough data available at the study site. Hybrid models refer to combinations of traffic flow-based models and data-driven models (Elhenawy et al., 2014; Hofleitner et al., 2012). Widespread application of hybrid models in the future could be promising because they have the

advantages of both the traffic-flow model and the data-driven model. For more details, readers can refer to reviews of up-to-date methods concerning traffic forecasting (Mori et al., 2015; Van Hinsbergen et al., 2007; Vlahogianni et al., 2014).

Since the application of probe vehicles makes large amounts of data available, data-driven models are drawing increasing attention. There are two main types of data-driven models: the parametric model and the non-parametric model. Among the different types of parametric models, time series models, which take advantage of the temporal relationship between travel time and traffic condition, have received the most attention. For example, a Kalman filter (KF) is commonly used to estimate the traffic condition and can be associated with an autoregressive integrated moving average (ARIMA) model, which can project the recurrent pattern of the traffic condition into the future (Guo et al., 2014; Xia et al., 2011). Other methods such as the Bayesian dynamic linear model (Fei et al., 2011) are also used to make predictions under both recurrent and non-recurrent traffic conditions. Most parametric models can provide accurate predictions under linear situations, such as highways and arterials, but they have difficulty addressing non-linear situations, such as urban roads. In the case of a non-linear situation, non-parametric models are preferred.

Among the different types of non-parametric models, neural network (NN) models and pattern recognition models are most popular. Researchers have devoted substantial efforts to developing different types of NN models. For example, Jiang and Zhang (2001) proposed a generalized regression neural network that can convert the average speed into a travel time and tested the model on an arterial road in China; Lint et al. (2005) proposed a state-space

neural network (SSNN) to predict travel time on freeways with missing data; Lint (2008) further combined SSNN and KF to solve the inherently delayed travel time prediction problem. Fusco et al. (2016) applied a NN model to an urban network, and the results showed that the NN model outperforms the ARIMA model under recurrent congestion conditions. Although NN models are capable of extracting the complex relationships between different traffic variables, there are three main drawbacks that constrain them from practical applications on large networks. First, NN models require long training processes and have to train at each study site. Second, NN models require large storage capacity to store a huge amount of data, but the storage capacity of a vehicle's onboard unit is limited because of other smart functions. Third, NN models lack interpretability due to their 'black box' nature (Elhenawy et al., 2014; Mori et al., 2015; Xu et al., 2018).

On the contrary, pattern recognition models are easy to implement and transfer to different sites without data training. For example, the KNN model, which is more popular than other pattern recognition models such as particle filtering (PF) model (Chen and Rakha, 2014), support vector machine (Wang and Shi, 2013), and cluster analysis (Xia et al., 2012), is used widely to predict different traffic variables. Smith and Demetsky (1994) showed that KNN models have the potential to serve as accurate and portable prediction models and have advantages over NN models. Habtemichael and Cetin (2016) compared the KNN model with the adaptive KF model and the seasonal ARIMA model when predicting the traffic flow on freeways. The results demonstrated that the KNN model outperforms the adaptive KF model and the seasonal ARIMA model in real-time traffic control and management for freeways.



Robinson and Polak (2005) predicted the urban link travel time in London and tested their model using different KNN design parameters.

## **2.2 Extensions of travel time prediction to urban networks**

When study sites of travel time estimation extended to urban networks where the penetration rate of probe data is still low, many researchers devoted their efforts to improve the estimation. For example, Wan et al. (2016) reconstructed the maximum likelihood trajectory to estimate the time spent on each segment of the road; Li et al. (2018) reconstructed vehicle trajectories based on a KNN regression algorithm so as to support travel time estimation; Jenelius and Koutsopoulos (2013) proposed a statistical regression model to estimate urban road travel time by vehicle trajectory data and included correlation between travel times of different links based on a spatial moving average structure to capture the spatiotemporal variations in speeds; Sanaullah et al. (2016) used a distance and time proportion method based on the map-matched points from adjacent links to enhance the coverage and to reduce the uncertainty of the estimation, and further applied spatial and temporal moving average to improve the accuracy; Alrukaibi et al. (2018) used simulation to increase the sample size and employed neighbor links to estimate the travel time of the target link. However, there are few researchers who consider problems resulted from the low penetration rate when making travel time prediction because most study sites of travel time prediction were highways and urban arterials where data are easy to collect.

Unlike the highway, traffic of roads on the same urban network often affects each other, especially those on adjacent links. For example, Bauer et al. (2019) investigated the travel time covariance between pairs of urban links when making the travel time prediction and found out that there were significant correlations between residuals on adjacent links. They pointed out it is necessary to add the travel time covariance when summing up the predicted link travel time and its prediction error to generate the predicted route travel time. Therefore, spatiotemporal relationships were often added to different types of models to improve accuracy when predicting travel time and other traffic-related parameters. Wang et al. (2016) reviewed traditional statistical models which included space-time autoregressive integrated moving average (STARIMA) and proposed a space-time delay neural network model (STDNN) which integrated the spatiotemporal autocorrelation of road traffic networks using NN. They used data collected from Automatic Number Plate Recognition cameras in London for travel time prediction and showed that the STDNN is more accurate than the STARIMA because it can capture spatiotemporal autocorrelation locally and dynamically. Cai et al. (2016) used a spatiotemporal state matrix to describe the traffic state instead of only a time series as in the classic KNN model at urban networks. The results showed that the improved KNN model outperformed the Elman-NN model. To make multi-time-step travel time prediction for route buses, Petersen et al. (2019) proposed a convolutional long short-term memory (LSTM) neural network which used a convolutional layer to capture the spatial correlations between different route segments and used the LSTM layer to capture temporal travel time pattern. Jenelius and Koutsopoulos (2018) proposed a multivariate probabilistic

principal component analysis (PPCA) model which can capture spatiotemporal correlations from historical data to predict urban network travel time. They tested the PPCA model under different penetration rates and applied an EM algorithm to deal with missing data. However, the lowest penetration rate in their study was 55% which can be hardly achieved in the real world.

In addition, traffic signals at urban networks should also be considered but most researchers chose to neglect the influence of traffic signals for the sake of simplicity. For instance, Fusco et al. (2016) touched upon the connection between traffic signals and individual speeds by considering the distribution of individual speeds but did not consider their overall influence. Feng et al. (2014) developed a Bayesian method to determine actual traffic conditions in real-time based on synthetic GPS data and a signal timing that is known and static.

### **2.3 Link Transmission Model and its applications**

So far, many DTA models have been developed and they can be categorized into two groups—the analytical model and the simulation-based model (Peeta and Ziliaskopoulos, 2001). The simulation-based approach was preferred when describing the spatiotemporal interactions and traffic flow propagation because analytical approaches cannot replicate traffic relationships adequately. The DNL model plays a critical role in simulation-based approaches because it can capture the progression of the traffic flow which accounts for the congestion and

delay on networks (Osorio and Flötteröd, 2014).

Among all DNL models, the cell transmission model (CTM) seems a suitable choice because it can capture traffic flow variability on each link based on the kinematic waves theory (KWT) (Daganzo, 1994). However, its application in the real world is constrained by the triangle shape of the fundamental diagram. Although Sumalee *et al.* (2011) introduced stochastic elements to relieve this constraint, Gentile (2010) criticized that the CTM suffered from high computational cost because it divided a link into small cells, which also deteriorated the accuracy of the CTM. Therefore, the LTM which can capture the progression of traffic flow in terms of cumulative counts (Newell, 1993) at the link's boundaries is preferred in this dissertation. It is proven that the LTM is more computationally efficient and robust than the CTM because it applies simplified KWT without separating the link (Chakraborty *et al.*, 2018; Gentile, 2010; Nezamuddin and Boyles, 2014).

Yperman (2007) first combined the cumulative curves and the CTM to propose the classic LTM which was based on the triangle fundamental diagram. Because the assumption of triangle fundamental diagram limited the application of the classic LTM, Gentile (2010) proposed a general LTM (GLTM) which was based on any concave fundamental diagram. Van der Gun *et al.* (2017) made a similar effort to extend the classic LTM to any continuous concave fundamental diagram in addition to a capacity drop. Although this extension had desirable properties like realism, it increased the computational cost and required temporal discretization to find an approximate solution. Consequently, Bliemer and Raadsen (2018) proposed on-the-fly multi-step linearization techniques to reduce the computational cost and it

led to an exact solution in continuous time. The LTM was applied only to road networks until Gentile (2017) extended the LTM to transit and pedestrian networks. To further describe the traffic situation in the real world, Flötteröd and Osorio (2017) added the stochasticity at the upstream and downstream boundaries of a link and decomposed the network to capture stochastic dependencies between queues. So far most LTM research was based on computer simulation, only a few researchers (Hajiahmadi *et al.*, 2013; Himpe *et al.*, 2016) tested the LTM with data from the real world.

Since the LTM is computationally efficient and can adequately capture the progression of traffic flow, it is widely used to address different issues. Although there was no explicit velocity equation in the LTM, Hajiahmadi *et al.* (2013) used the delays generated from the LTM to provide variable speed limit control for traffic networks. Levin (2017) solved the shared autonomous vehicle routing problem resulting from the combination of the dial-a-ride service constraints and the linear program for the SODTA which was modeled by the LTM. To address the environmental issue, Long *et al.* (2018) used SODTA models to minimize total system emissions in single destination networks based on the LTM. Chakraborty *et al.* (2018) applied the LTM to solve the network design problem by minimizing the difference between the inflow and the outflow of each link under the flow-conservation and budget constraints. Gentile (2015) presented a general framework to reproduce network congestion using GLTM. To reduce the network congestion, Van de Weg *et al.* (2016) reformulated the LTM into a linear programming problem to make sure the link outflow is no more than the corresponding inflow considering the shock-wave dynamics, but this attempt also ended up minimizing the

difference between the inflow and the outflow of each link.

# **CHAPTER 3 Considering Vehicles in the Crossing Direction to Enhance the Coverage Rate of Prediction under Low Penetration Rate**

## **3.1 Introduction**

As living standards improve, the public is no longer satisfied with a transportation system that can only deal with traffic problems that have already happened. A timely, reliable, and safe transportation system that can predict traffic conditions is hence required (Smith et al., 2002). Recently, the probe vehicle has been widely recognized as a promising source for collecting large amounts of traffic data to support such kind of system. However, the penetration rate of probe vehicles is low at current urban networks and is expected to be restricted in the future. The penetration rate of probe vehicles has a significant influence on traffic prediction so many researchers focused on improving the accuracy of the travel time prediction. Except for the accuracy, the coverage rate of the travel time prediction is also critical for its application in the real world. Although some developed models can make quite accurate predictions, it is doubtful that they have a high coverage rate for the practical application when the penetration rate is low. In this study, the coverage rate is defined as the proportion of targets a model can predict. The objective of this chapter is to enhance the coverage rate of the travel time prediction and maintain its accuracy when the penetration rate is low.

To realize the objective, this chapter proposed a non-parametric model which not only utilized the spatiotemporal relationship between vehicles in the target link, but also took advantage of the spatiotemporal relationship between vehicles in the target link and vehicles in the crossing direction. Most non-parametric models need a large amount of probe data for the frequent training and calibration requirements because of their data-driven characteristics (Dhivyabharathi et al., 2016). Therefore, the computer simulation was often used to evaluate the performance of the model and the influence of different penetration rates of probe vehicles (Bucknell and Herrera, 2014; Zheng and Zuylen, 2013). In this chapter, a computer simulation was used to compare the performance of the proposed model with two conventional non-parametric models based on the KNN and the PF approaches, at different penetration rates.

## **3.2 Methodology**

### **3.2.1 Descriptions of the proposed model**

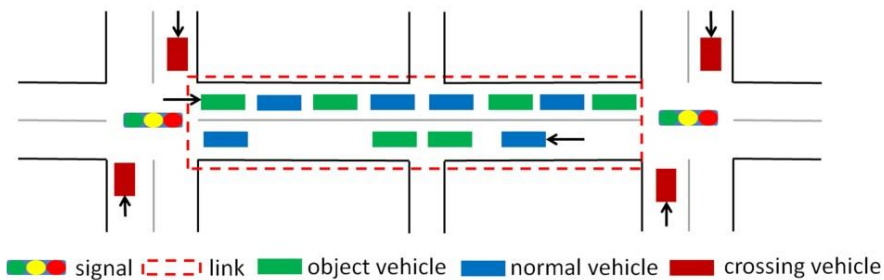
Urban link travel time is influenced by many factors such as signal timing, overtaking behavior, and turning at intersections. In this chapter, to capture the influence of traffic signals, a link is defined as a segment of the road which is separated by two adjacent signalized intersections. Many researchers have demonstrated that different turning movements experience different delays at signalized intersections and exhibit significantly different distributions (Lu et al., 2018). For simplicity, attention was focused on predicting



the travel time for vehicles going straight because vehicles turning left or right from the opposing direction must give way to vehicles going straight, which usually form the majority of the total traffic (Feng et al., 2014). Moreover, in this study, it was assumed that there is a right-turn lane (in the case of Japan) so vehicles going straight were not influenced by vehicles with other turning choices. Because vehicles going straight have priority when going through non-signalized intersections, roads separated by non-signalized intersections were treated as one link with signalized intersections at its end-points. The link travel time consists of both the time taken to traverse the link and the stopping time due to the traffic and the traffic signal at the downstream intersection. Hence, the exit time of a link was used as the time stamp. For clarity, the main terms in this chapter are defined in Table 3.1 and illustrated in Figure 3.1.

**Table 3.1 Definition of terms related to vehicles**

Term	Definition
Object vehicle	Probe vehicle that travels straight through the downstream signalized intersection.
Normal vehicle	Vehicle that cannot send probe data.
Crossing vehicle	Probe vehicle traveling in the crossing direction that goes through the same downstream signalized intersection.
Penetration rate	The ratio of probe vehicles to all vehicles
Coverage rate	The proportion of time points that can be predicted



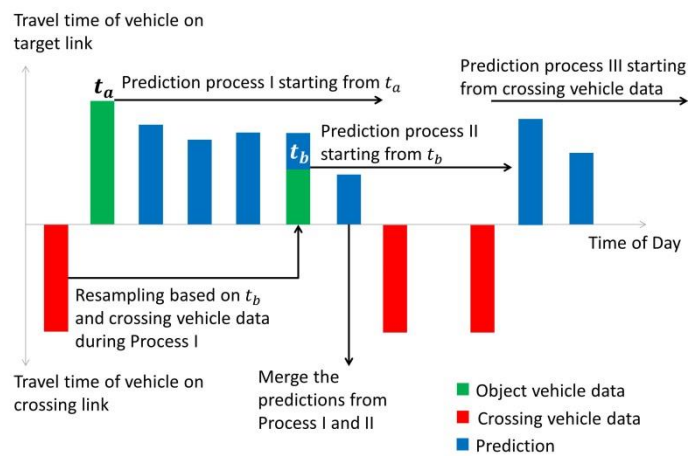
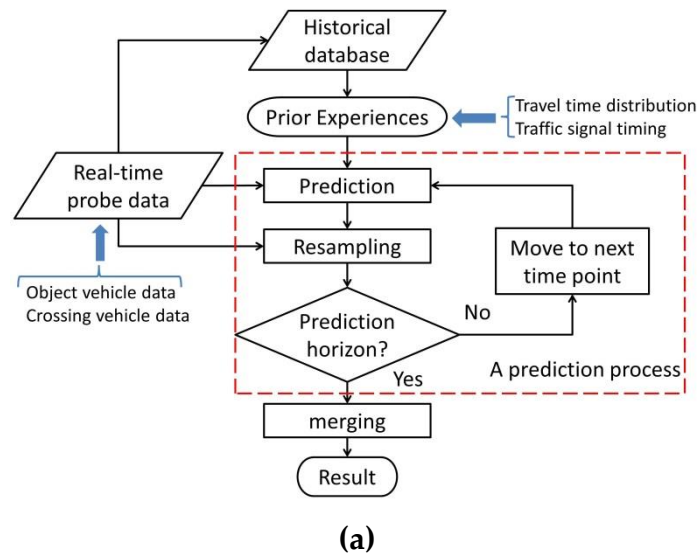
**Figure 3.1 Vehicles at a regular intersection (where the penetration rate is 50%)**

There are two main applications of the proposed model in the real world: travel time

reliability analysis and reliable route searching. The prediction in the proposed model is represented by a distribution instead of the weighted summation. Travel time distribution can provide more information than the weighted summation such as the travel time variability which plays an important role in travel time reliability measurements. The proposed model can make dynamic link travel time predictions in the form of distribution and these predictions can be used as the input for reliability analysis models such as the mean-variance model (Carrion and Levinson, 2012; Li et al., 2010). Li et al. (2010) pointed out that risk-averse travelers are willing to pay for the reduction in travel time variability rather than travel time savings. Some of them prefer the more reliable route, even though the expected travel time is higher in comparison to other routes with shorter expected travel time and higher uncertainty. Chen et al. (2016) proposed a two-stage reliable path-finding algorithm and compared it with other algorithms on the urban network in Wuhan, China. They used the link travel time distributions which were estimated by existed data from a floating-car system as the input of the reliable path-finding algorithms. Because our proposed model could achieve a high coverage rate on urban networks when the penetration rate is low in the real world, its predictions of link travel time distributions can be used for real-time reliable route searching by different reliable path-finding algorithms (Chen et al., 2016). When generating the route travel time, both link travel time and travel time covariance are summed up. In this study, the focus was limited on the link travel time prediction so the travel time covariance estimation remains as future work.

### 3.2.2 Details of the proposed model

The structure of our proposed model is shown in Figure 3.2 (a). The proposed model is based on the prediction process consisting of prediction and resampling. According to Bucknell and Herrera (2014), there is not much difference in prediction when the time interval is shorter than 5 s. To capture the frequent changes in travel time on the urban network, the prediction time interval should be set as short as possible. Therefore, it was set to 5 s in this chapter.



**Figure 3.2 Framework of the proposed model: (a) Model structure; (b) Example of using the data for prediction**

The proposed model can make successive predictions as long as the computation resources are available, which means it is able to make predictions for several minutes or even hours later. However, to reflect the influence of signal timing, the prediction period is based on the average length of the green-light signal phase. Although information about signal timing might be accessible to researchers or even the public in the future, at present, it is still difficult to access in some countries such as Japan. Several methods that used trajectory data from probe vehicles have been developed to estimate signal timing (Axer et al., 2017; Fayazi et al., 2015; Kerper et al., 2012; Yu and Lu, 2016). However, in this study, there were no trajectory data available before the vehicles stop, so a simple algorithm was developed to approximately estimate the signal timing using the data collected when vehicles exit a link. This algorithm can also be applied to estimate the actuated traffic signal by data from links with similar characteristics and traffic conditions. Details of the estimation algorithm are given in Appendix A.

In the proposed model, data from both object vehicles and crossing vehicles are used for the real-time prediction and are recorded in the historical database for estimating the travel time distribution and the signal timing. Figure 3.2 (b) gives an example of how data are used in the proposed model. If travel time  $t_a$  from an object vehicle is observed in a data stream, a prediction process (Process I) will start and the travel time is predicted every 5 s until the prediction horizon. If a new object vehicle whose travel time is  $t_b$  is observed during the Prediction Process I, candidates for the travel time prediction at that time point will be resampled based on  $t_b$  and the crossing vehicle in the preceding red phase. At the same time,

a new prediction process (Process II) will start. If there are prediction results from different prediction processes at the same time point, they will be merged together. In addition, if no object vehicle is observed after the red phase, prediction can also be made based on the crossing vehicle data in the preceding red phase (Process III).

For one prediction, travel time is determined by both the first observed object vehicle's travel time and the previous travel time prediction. Provided the prediction process starts when there is an observed data  $t_n$  at time point  $n$ , the probability of travel time  $t_{n+l}^j$  at time point  $(n + l)$  can be calculated by

$$P(t_{n+l}^j) = \alpha P(t^j | t_{n+l-1}^i, D_{n+l-1}^{n+l}) + (1 - \alpha) P(t^j | t_n, D_n^{n+l}) \quad (3-1)$$

where  $P(t^j | t_q, D_q^p)$  represents the probability of travel time  $t^j$  at time point  $p$  given travel time  $t_q$  at time point  $q$  ( $p \geq q$ ) and  $D_q^p$  is the leaving time difference (LTD) between  $p$  and  $q$ . Parameter  $\alpha$  is defined as  $l/(l + 1)$ , which states that the object vehicle receives more influence from a vehicle that is closer to it. Next, one of the travel time candidates at time point  $(n + l)$  is calculated based on the first  $k$  possible travel times with the biggest probability using

$$t_{n+l}^i = \sum_j^k P(t_{n+l}^j) t^j / \sum_j^k P(t_{n+l}^j) + \delta \quad (3-2)$$

where  $\delta$  is an error term which follows a standard Gaussian distribution. To compare the proposed method with the KNN-based model, a weighted summation of these candidates is used to represent the prediction and weight  $w_m^i$  is calculated by

$$w_m^i = p_N(t_m - t_m^i) \quad (3-3)$$

where  $t_m$  is the observed travel time and  $p_N$  is a likelihood function, which is chosen to be a standard Gaussian distribution.

**Table 3.2 Algorithm I: Proposed travel time prediction process**

- 
1. Initialize prediction horizon  $M_G$  using Algorithm A
  2. If at time point  $n$ , there is an observed travel time  $t_n$ ,
  3. For  $i = 1:100$ ,
  4.     Generate possible candidate  $t_n^i$  using  $P(t^j|t_n, D_n^n)$  with error term  $\delta$ ;
  5.     Calculate similarity  $w_n^i$  for each candidate  $t_n^i$  using Eq. (3-3);
  6. End For
  7. For  $l = 1: M_G$ ,
  8.     For  $i = 1:100$ ,
  9.         For each possible travel time  $t^j$ , calculate probability  $P(t_{n+l}^j)$
  10.         at time point  $(n + l)$  using Eq. (3-1);
  11.         Calculate travel time candidate  $t_{n+l}^i$  using Eq. (3-2);
  12.     End For
  13.     If object vehicle  $t_{n+l}$  is observed at time point  $(n + l)$ ,
  14.          $w_{n+l}^i = p_N(t_{n+l} - t_{n+l}^i)$ ,  $i \in [1,100]$ ;
  15.         Begin the resampling process to modify the candidates.
  16.     Else if a crossing vehicle is observed at time point  $(n + l)$ ,
  17.         If there is an observed crossing vehicle at time point  $(n + l - p)$ ,
  18.              $t_{n+l}^i = t_{n+l-p}^i$  ( $l-p < M_R, p < l$ );
  19.         Else
  20.              $t_{n+l}^i = t_{n+l}^i + 5M_R$ ,  $i \in [1,100]$ ;
  21.              $w_{n+l}^i = w_{n+l-1}^i$ ,  $i \in [1,100]$ ;
  22.         Else,  $w_{n+l}^i = w_{n+l-1}^i$ ,  $i \in [1,100]$  ;
  23.     End For
- 

\* $M_G$ =length of green phase/5 and  $M_R$ =length of red phase/5

Algorithm I for the prediction process is listed in Table 3.2. The prediction horizon is defined to be the estimated green phase length, so each prediction process may have a different length (Line 1). The prediction process starts when an object vehicle is observed (Line 2). If there is more than one object vehicle observed at the same time point, the average

travel time is used because the travel times are similar within 5 s in most cases. Here, 100 candidates are generated according to probability  $P(t^j|t_n, D_n^n)$  with the corresponding weight  $w_n^i$  (Lines 3-6). Following that, candidates at step  $l$  are calculated (Lines 8-12). During the prediction, if a new object vehicle is observed at the time point  $(n + l)$ , a resampling process begins and the weights are updated (Lines 13-15). If a crossing vehicle is found and it is the first crossing vehicle in the same red phase, the travel time candidate is increased by the length of a red phase because it is assumed that the vehicle must come to a full stop to wait for the signal at the intersection (Line 20). Otherwise, the lengths of the candidates remain the same as those at the earlier step if a crossing vehicle is observed (Line 18). Moreover, if no new object vehicle is observed, the weights remain the same as that of the preceding step (Lines 21, 22).

If there is no observed object vehicle, Algorithm I cannot be applied, and this situation often happens because of the low penetration rate. To address this situation, an object vehicle's travel time can be estimated using the travel time distribution under the corresponding exit time difference (ETD)  $d_n$ . The ETD is defined as the difference between the exit time of an object vehicle and the last observed crossing vehicle that goes through the intersection before it.

To be specific, the 100 candidates are generated according to the conditional probability  $P(t|d_n)$ . An error term  $\delta$  is also added to reflect some unexpected situations. The average travel time of the 100 candidates is used as the object vehicle's travel time  $t_n$ . Given  $t_n$  and 100 candidates at time point  $n$ , Algorithm I can be applied.

The prediction process tends to decrease the travel time because, in the green phase, the queue is decreasing. Although the travel time might be long at the beginning, it will soon drop to a normal level after several prediction steps. However, when congestion or accidents happen, the queue exists for longer, so the prediction is shorter than the real travel time. The resampling process relieves this problem by replacing candidates with smaller weights with candidates with bigger weights, just as in the conventional PF model. During the prediction process, if a new object vehicle is observed, it will be used to resample candidates at the current step. For example, if the real-time traffic condition is congested, the short travel-time candidates will be replaced by long travel-time candidates. However, this method lacks diversity because it resamples from existing candidates. To improve diversity, candidates derived from probability  $P(t|d_n)$  are introduced.

**Table 3.3 Algorithm II: Resampling**

- 
1. If at time point  $m$ , an object vehicle data  $t_m$  is observed,
  2.     Sort candidates according to their weight in decreasing order,
  3.     and remove the later 50 candidates;
  4.     If  $d_m < M_G$  at time point  $m$ ,
  5.         For  $j = 1:100$ ,
  6.             Select  $t_m^j$  according to  $P(t|d_m)$ , calculate weight  $w_m^j$  using Eq. (3-3);
  7.             If  $w_m^i < w_m^j$  ( $i = 50$ ),  $t_m^k = t_m^j, w_m^k = w_m^j$  ( $k = 1 \dots K$ );
  8.     End For
  9.     Combine  $\{t_m^1, t_m^2, t_m^3 \dots t_m^{50}\}$  with  $\{t_m^1, t_m^2, t_m^3 \dots t_m^K\}$
  10.     and sort candidates according to weight in decreasing order;
  11.     Else  $K = 0$ ;
  12.     If  $50 + K > 100$ , remove the later  $(K - 50)$  candidates;
  13.     If  $50 + K < 100$ ,
  14.         Select  $(50 - K)$  candidates randomly from  $\{t_m^1, t_m^2 \dots t_m^{50+K}\}$
  15.         according to their weight and add them to  $\{t_m^1, t_m^2 \dots t_m^{50+K}\}$ ;
-



Algorithm II for resampling samples is shown in Table 3.3. Here, the resampling rate is set to 50%. Providing the resampling process starts at time point  $m$  ( $n < m \leq n + M_G$ ), the 50 candidates with the lowest weights are removed (Lines 1-3). If the ETD is less than the length of the green phase, another 100 candidates are selected according to conditional probability  $P(t|d_m)$  and their weights are calculated by Eq. (3-3) (Lines 4-6). Then, new candidates whose weights are bigger than  $w_m^i$  ( $i = 50$ ) are added to  $\{t_m^i\}$  ( $i \in [1,50]$ ) (Lines 7-10). If the size of the new candidate set is larger than 100, the candidates with the smallest weight are removed. Otherwise, candidates are copied randomly until the size increases to 100 (Lines 13-16).

Because a prediction starts whenever there is an observed object vehicle or an estimated one and the process continues until it reaches the prediction horizon, there might be several predictions for the same time point. Therefore, it is necessary to merge these predictions into one. The candidates from each prediction are determined. The number of candidates is proportional to the duration between the starting and merging time points. The weight of each selected candidate is normalized, and then either a weighted summation or the candidate with the maximum weight is used to compare with the prediction results of the KNN-based and the PF-based models.

## 3.3 Experiments

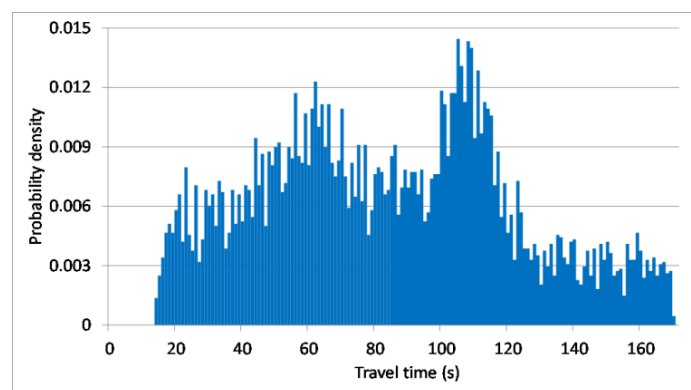
### 3.3.1 Data description

In this study, models have to be tested under different penetration rates to show the influence of the penetration rate on the coverage rate. Since the probe data are usually collected from special vehicles such as taxis and buses, the penetration rate of real-world data is at a low level. Therefore, the simulation data are required to obtain traffic data at different penetration rates. VISSIM has been proven to be an effective simulation model to reproduce real-world traffic flow under different traffic conditions at both microscopic and macroscopic levels because it applies a psycho-physical car-following model that can adapt different driving behaviors (Bloomberg and Dale, 2000; Fellendorf and Vortisch, 2001). In addition, VISSIM allows users to adjust model parameters so that it can reflect the traffic condition for a particular real-world case (Fellendorf and Vortisch, 2001). However, since this study focused on enhancing the coverage rate instead of improving the prediction accuracy on a certain link, default settings for the car-following model and vehicle delivery were used.

The traffic simulation was constructed using VISSIM (version 7.0) at a normal cross intersection with four 200-m-long links and attention was focused on only one link called the target link. For simplicity, vehicles only moved straight through the intersection so there was only one lane for each link. There was no non-signalized intersection within a link. To reproduce the real-world traffic condition, the speed when vehicles enter the link varied randomly from 10 to 50 km/h which is based on the real-world situation in Nagoya. To

reproduce the appearance and disappearance of traffic congestion during a day in the real world, it is based on the distribution of vehicle trips from the 2017 American National Household Travel Survey so as (U.S. Department of Transportation, 2017), traffic flow of the target link was set to 200, 600, and 100 vehicles/h in the first, second, and third 10 min respectively, and this pattern was repeated every 30 min. As for the other three links, the traffic volume was fixed at 200 vehicles/h for simplicity. The signal pattern was set to have two phases without the all-red phase, and the length of each phase was 60 s.

To obtain sufficient data, the simulation was repeated 30 times. One trial was used as the test data and the remaining 29 simulations made up the historical database. The number of data in the testing database was 295, whereas the number of data in the historical database was 8,568. The average link travel time in the testing database was 90.5 s, whereas in the historical database, it was 85.7 s. The distribution of the target link’s travel times is illustrated in Figure. 3.3.

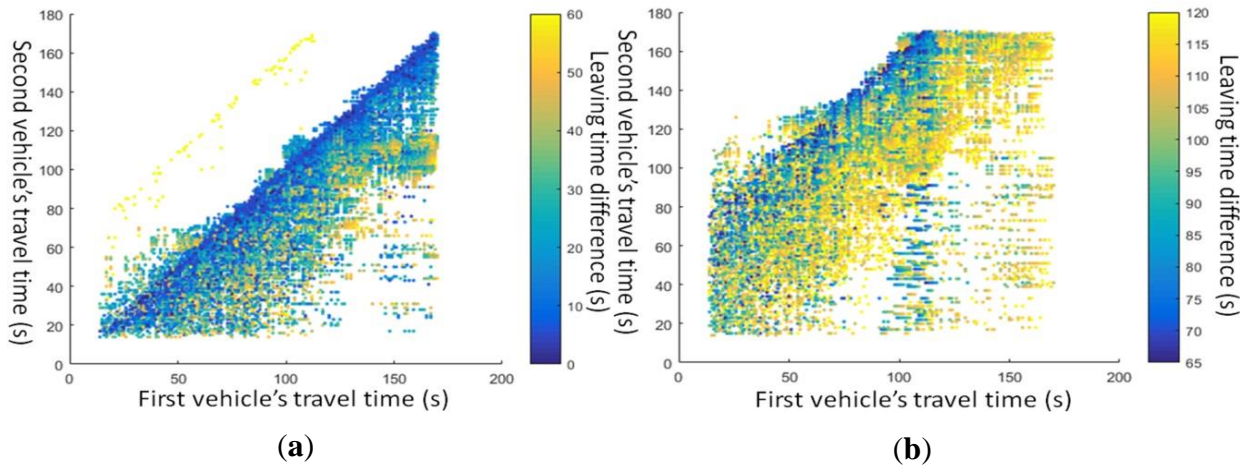


**Figure 3.3 Distribution of simulation travel times**

### 3.3.2 Relationships between individual vehicles

The prediction process of the proposed model was based on the spatiotemporal

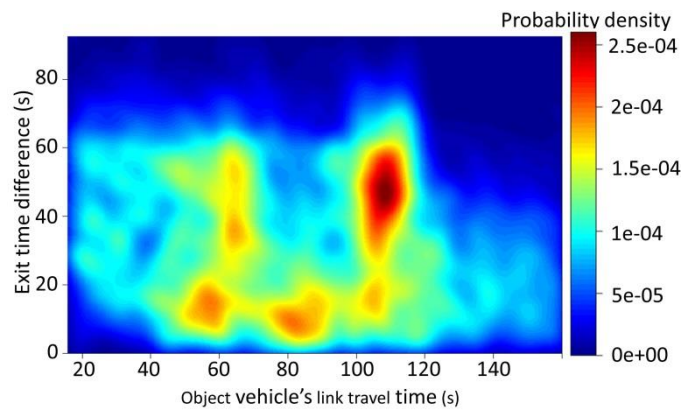
relationship between individual vehicles. Bayes' theorem was introduced to describe the interactions among vehicles and reflect the influence of traffic signals. Lu et al. (2018) pointed out that travel time concerning signalized intersections might follow various distributions. Therefore, in this chapter, no specific distribution was employed.  $P(t^j|t_q, D_q^p)$  was derived from the relationship between the travel times for two object vehicles, while  $P(t|d_n)$  was derived from the relationship between the ETD and object vehicle's travel time. The relationship between the travel times for two object vehicles is presented in Figure. 3.4.



**Figure 3.4 Relationship between the travel times of two object vehicles: (a) LTD within 60 s; (b) LTD more than 60 s**

The horizontal axis represents the travel time of the first vehicle, whereas the vertical axis represents the travel time of the second vehicle. The color indicates the difference between the times when the two vehicles leave the link, and each dot represents a pair of travel times with different LTDs. In Figure 3.4(a), the diagonal consists of pairs (dark blue dots) that are close to each other, which means that travel times were similar for small LTD. Most pairs are distributed under the diagonal, which means that after the signal turned green, vehicles that arrived later had a higher probability of going through the intersection without

delay or needing to stop. There is a line of yellow dots parallel with the diagonal (with a vertical axis intercept of 60 s). These dots represent two vehicles that were suddenly separated by the red signal. In Figure 3.4(b), most pairs are distributed above the diagonal because the second vehicle was separated from the first vehicle by a red signal and was likely to stop at the intersection (the LTD ranges from 60 to 120 s). Consequently, if the first vehicle's travel time  $t_{first}$  and the LTD  $D_{first}^{second}$  are known, the probability of travel time of the second vehicle  $P(t|t_{first}, D_{first}^{second})$  can be inferred by Bayes' theorem.



**Figure 3.5 Relationship between object vehicles and crossing vehicles**

The relationship between the ETD and the object vehicle's travel time is presented in Figure. 3.5. When the ETD was low (e.g., less than 20 s), the travel time tended to be longer (e.g., more than 60 s) because if vehicles go through the intersection right after the signal turns green, they probably have a stop at that intersection. When the ETD increased, the probability density of short travel time increased because if vehicles go through the link during the middle or end of the green phase, they are more likely to go through the intersection without a stop. The two peaks for the long travel times when the ETD is relatively large represent congestion in which the vehicles must stop at least once at the

intersection. Consequently, if the ETD is known, the probability of the travel time of the object vehicle  $P(t|d_n)$  can be inferred by Bayes' theorem.

### 3.3.3 Models for comparison

Two non-parametric models based-on the KNN and the PF approaches were used to compare with the proposed model. For the KNN-based model, it selected time-sequential samples  $\{x_m^{(i)}, t_{m+l}^{(i)}\}$  ( $x_m^{(i)} = [t_{m-n}^{(i)}, t_{m-n+1}^{(i)}, \dots, t_m^{(i)}]$ ) at different time points which were indicated by  $m$  from historical data. Here,  $n$  is the length of the sample and samples can be collected from different days. To predict the travel time, historical samples were compared with the current traffic condition which is measured by  $x_c = [t_{c-n}, t_{c-n+1}, \dots, t_c]$ . Each sample has a weight  $w_m^{(i)}$  that represents its similarity with the current traffic condition. There are several ways to calculate the weight by selecting the distance metric, but Robinson and Polak (2005) concluded that the KNN-based model is not sensitive to the distance metric. In this study, the Euclidean distance is used to calculate the weight of sample  $i$  as follows:

$$w_m^{(i)} = 1/\sqrt{(x_c - x_m^{(i)})(x_c - x_m^{(i)})^T} \quad (3-4)$$

The KNN-based model can use the first  $k$  samples with the biggest weights to make the prediction at  $l$  step forward until the prediction horizon  $L$  ( $n+l < L$ ) as long as there are corresponding samples, as follows:

$$t_{c+l} = \sum_{i=1}^k w_m^{(i)} \cdot t_{m+l}^{(i)} / \sum_{i=1}^k w_m^{(i)} \quad (3-5)$$

Unlike the KNN-based model, the PF-based model did not traverse all the historical data. It randomly selected  $N$  candidates  $\{\mathbf{x}_m^{(i)}, t_{m+1}^{(i)}\}$  and resampled part of them using the sampling importance resampling (SIR) filter to solve the problem of degeneracy (Chen and Rakha, 2014). According to the SIR, candidates were resampled based on their weight as follows:

$$w_m^{(i)} \propto w_{m-1}^{(i)} \cdot \frac{p(\mathbf{x}_c | t_m^{(i)}) p(t_m^{(i)} | t_{m-1}^{(i)})}{q(t_m^{(i)} | t_{m-1}^{(i)}, \mathbf{x}_c)} \quad (3-6)$$

In the SIR, the importance density  $q(t_m^{(i)} | t_{m-1}^{(i)}, \mathbf{x}_c)$  was assumed to have the same value as the transitional prior pdf  $p(t_m^{(i)} | t_{m-1}^{(i)})$ , so Eq. (3-6) can be simplified as Eq. (3-7) where  $p(\mathbf{x}_c - \mathbf{x}_m^{(i)})$  represents the similarity between the candidate and the current traffic condition. There are also several ways to calculate the weight, but the same calculation as Eq. (3-4) was used for the PF-based model in this study. Readers can refer to the work of Chen and Rakha (2014) for more details about the PF in travel time prediction.

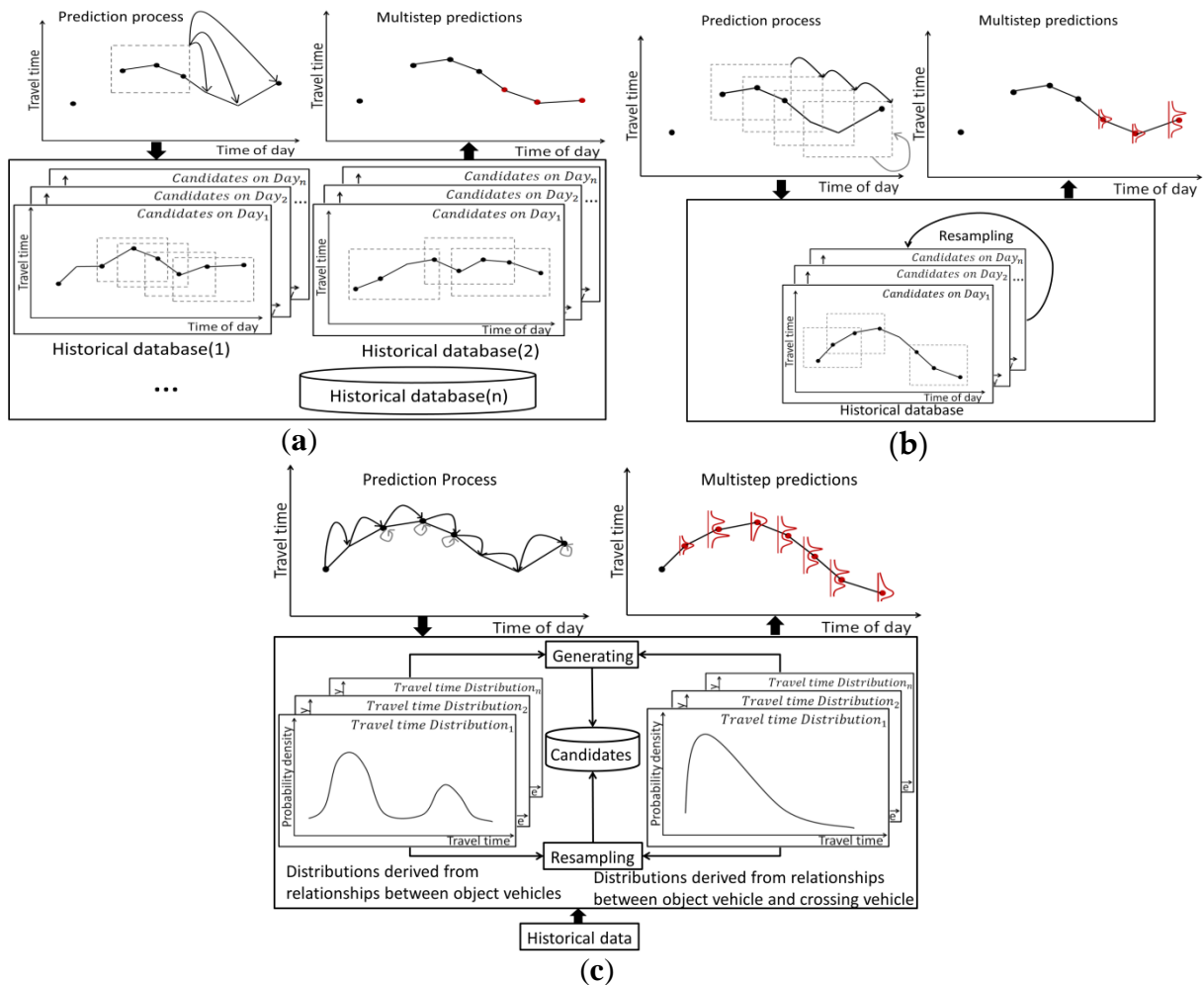
$$w_m^{(i)} \propto p(\mathbf{x}_c | t_m^{(i)}) = p(\mathbf{x}_c - \mathbf{x}_m^{(i)}) \quad (3-7)$$

The PF-based model can make predictions successively by shifting the time window one step forward at each step until the prediction horizon  $L$ , as follows:

$$t_{c+1} = \frac{\sum_{i=1}^N w_m^{(i)} \cdot t_{m+1}^{(i)}}{\sum_{i=1}^N w_m^{(i)}} \quad (3-8)$$

As for the historical database, the PF-based model only needs one database once the length of the candidate is decided because its prediction step is identical. In contrast, the KNN-based model needs several historical databases according to the prediction horizon.

Process of the three models is illustrated in Figure 3.6, where the black dots represent the observed travel time of an object vehicle, whereas the red dots represent the model predictions.



**Figure 3.6 Models used in this study: (a) KNN-based model; (b) PF-based model; (c) Proposed model**

In the proposed model, the historical data provide the prior travel time distributions based on the spatiotemporal relationships between object vehicles and the spatiotemporal relationships between object vehicles and crossing vehicles. Therefore, a prediction can be made whenever there is an observed data sample (i.e., an object or crossing vehicle). In contrast, the KNN-based and the PF-based models can only make a prediction when there are



continuously observed object vehicle data. Because of the use of information from crossing vehicles to make predictions, the proposed model is expected to achieve a higher coverage rate than the other two models. As mentioned before, to make a prediction at one time point, there might be several prediction processes in the proposed model, depending on the number of observed vehicles during the prediction period. Therefore, it was difficult to calculate the computation cost for one prediction in the proposed model because it varied from case to case. However, the proposed model was expected to have higher computation costs than the KNN-based and the PF-based models which only have one prediction process for one prediction.

### **3.3.4 Experiment setting**

One-third of the testing data collected from the target link was selected randomly as travel times for prediction (target data), whereas the remaining two-thirds were treated as travel time that can be observed during the prediction (observed data). Then, some observed data and data from crossing vehicles (crossing vehicle data), were removed randomly to simulate the situation under different penetration rates. In the KNN-based and the PF-based models, historical samples with a similar pattern of the current traffic condition were used to predict the travel time for a certain link. Because different links have different characteristics, historical samples from one link can only be used to predict the travel time for that link. That is the reason why KNN-based and PF-based models cannot use the crossing vehicle data to predict the travel time for the target link. However, the correlation between adjacent links can help improve the accuracy of travel time prediction so many researchers applied the crossing

vehicle data to the link travel time prediction as introduced in chapter 2.2. Because the accuracy improvement is not the interest of this paper, the crossing vehicle data were not used in the KNN-based and the PF-based models in this study. On the other hand, the proposed model without considering crossing vehicles, namely proposed model  $\beta$  ( $PM_\beta$ ), should be tested to evaluate the influence of crossing vehicles because the travel time of the target link can be predicted directly from the crossing vehicle data in the proposed model.

As for the parameter calibration in the KNN-based model, Robinson and Polak (2005) pointed out that attention should be paid to determining the optimal value of  $k$  which depends on the database size. In their research, no matter how the database size changed, the optimal value of  $k$  was less than 300 and the difference between the mean absolute percentage error (MAPE) with a different value of  $k$  within that range was within 5%. A similar pattern of the relationship between the MAPE and the value of  $k$  as that in Robinson and Polak's (2005) research can be found in other KNN-based studies (Cai et al., 2016; Habtemichael and Cetin, 2016) and the difference between the MAPE with a different value of  $k$  in these studies was also within 5%. Although the value of  $k$  affects the accuracy of the KNN-based model, it is reasonable to fix the value of  $k$  without calibration because a 5% difference is at an acceptable level when the main concern of this chapter is not improving the accuracy. Consequently, the value of  $k$  in (3-5) was set as 4 and the sample length  $n$  was set as 2 to ensure the database size is large enough in advance.

For the parameter calibration in the PF-based model, Chen and Rakha's (2014) work showed that although the number of candidates and the value of the resampling rate

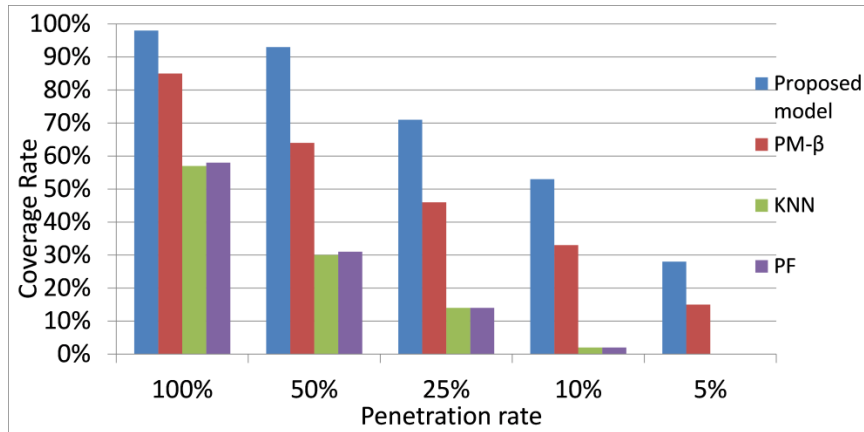
influenced the accuracy of the PF-based model, the difference in the MAPE was also within 5%. For the same reason, the number of candidates and the value of the resampling rate were fixed in advance without calibration. The number of candidates  $N$  in (3-8) was set as 100. The resampling rate was set as 50%, which means that the latter 50 candidates were replaced by the former 50 candidates according to their weight.

This chapter aims to maintain the accuracy of the proposed model at the same level as that of the KNN-based and the PF-based models, so some parameters in the proposed model were set as the same value as those in the two comparison models in advance. The value of  $k$  in (3-2) was set as 4, the number of candidates was set as 100, and the resampling rate was set as 50%. The discussion of  $\alpha$  is out of the scale of this chapter so it remains as future work. In addition, the time interval and prediction horizon for the three models were the same.

### **3.3.5 Results and discussion**

Although the number of target data was the same under different penetration rates, not all of them can be predicted. The proportion of target data that can be predicted was referred to as coverage rate. The coverage rate under different penetration rates for different models is shown in Figure. 3.7. The coverage rate for each model shrank when the penetration rate decreased, but the coverage rate of the proposed model was always higher than that of other models. Its coverage rate was also stable if the penetration rate was more than 50%. Even when the penetration rate was less than 50%, the proposed model could cover over half of the

points in time if the penetration rate was no less than 10%. The coverage rate may be influenced by the length of the time interval and the traffic volume, but these variables were set in advance in the simulation. Therefore, the analysis of the relationship between the coverage rate and these variables will remain as future work.



**Figure 3.7 Coverage rate under different penetration rates**

The same historical database was used for different penetration rates because it was assumed that sufficient data can be collected, regardless of the penetration rate, if the period of data collection was reasonably long. Consequently, the spatiotemporal relationships and signal timing estimation were the same for different penetration rates. The estimation of the green phase was 65 s with a 5-s variance, whereas the estimation of the red phase was 55 s with a 5-s variance. The estimation was comparatively close to the setting, which was 60 s for both phases.

Two indices were used to measure the accuracy of the proposed model. One was the MAPE and the other was the root mean squared error (RMSE), defined as follows:

$$MAPE = \frac{1}{N} \sum_i^N \frac{|t_i - \hat{t}_i|}{t_i} \quad (3-9)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (t_i - \hat{t}_i)^2} \quad (3-10)$$

where  $N$  is the total number of predictions. At time point  $i$ ,  $t_i$  is the true value of the travel time, whereas  $\hat{t}_i$  is its prediction.

The accuracy of the proposed model under different penetration rates is shown in Table 3.4.

**Table 3.4 Accuracy for the proposed model under different penetration rates**

<b>Penetration rate (%)</b>	<b>100</b>	<b>50</b>	<b>25</b>	<b>10</b>	<b>5</b>
Proposed model MAPE (%)	19.3	25.6	26.2	26.5	33.8
Proposed model RMSE	19.7	24.4	29.2	27.3	30.7
Average value MAPE (%)	71.9	65.5	57.5	58.9	54.0
Average value RMSE	43.1	42.6	44.3	39.8	34.2

The accuracy deteriorated slightly when the penetration rate decreased, but it remained at a stable level. If the average value of travel time in the historical database was used as the prediction, the accuracy increased when the penetration rate decreased. This is because, when the penetration rate decreased, the traffic conditions where predictions can be made were probably in congestion, and the average travel times in the historical database were relatively long. Nevertheless, the proposed model had higher accuracy than just using the average value under different penetration rates.

Because the coverage rate of each model under different penetration rates varied and the proposed model can make predictions at more points in time, it was unreasonable to compare the MAPE or RMSE directly. Therefore, the following adjusted measurements of accuracy

were used in this chapter:

$$Model - diff. MAPE = \frac{1}{N_a} \left( \sum_i^{N_a} \frac{|t_i - \hat{t}_i^{PM}|}{t_i} - \sum_i^{N_a} \frac{|t_i - \hat{t}_i^M|}{t_i} \right) \quad (3-11)$$

$$Model - diff. RMSE = \sqrt{\frac{1}{N_a} \sum_i^{N_a} (t_i - \hat{t}_i^{PM})^2} - \sqrt{\frac{1}{N_a} \sum_i^{N_a} (t_i - \hat{t}_i^M)^2} \quad (3-12)$$

where *Model* represents the name of the comparison method and  $N_a$  is the total number of predictions it can make. At time point  $i$ ,  $\hat{t}_i^{PM}$  is the prediction of the proposed model, and  $\hat{t}_i^M$  is the prediction of the comparison method.

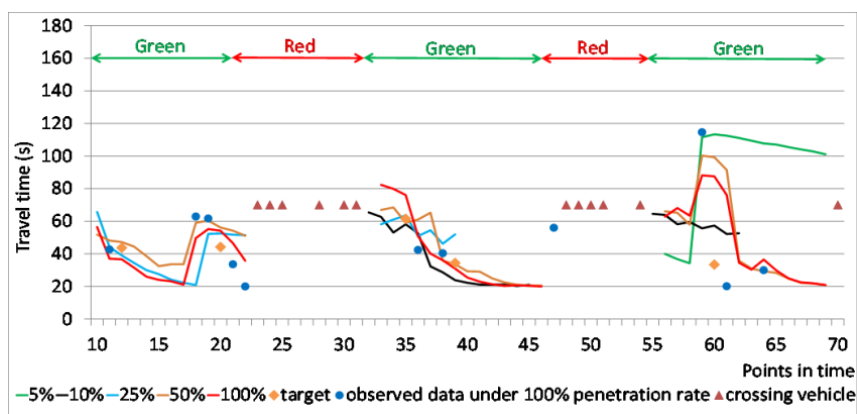
**Table 3.5 Studies on the impact of the car sharing**

<b>Penetration rate (%)</b>	<b>100</b>	<b>50</b>	<b>25</b>	<b>10</b>	<b>5</b>
kNN-diff.MAPE (%)	3.2	-1.0	-8.0	-	-
kNN-diff.RMSE	2.0	-1.0	-5.0	-	-
PF-diff.MAPE (%)	-12	-12	-27	-	-
PF-diff.RMSE	-9.0	-9.0	-15	-	-
$PM_\beta$ -diff.MAPE(%)	0.0	2.0	0.0	-1.0	-1.0
$PM_\beta$ -diff.RMSE	-3.0	1.0	2.0	0.0	-6.0

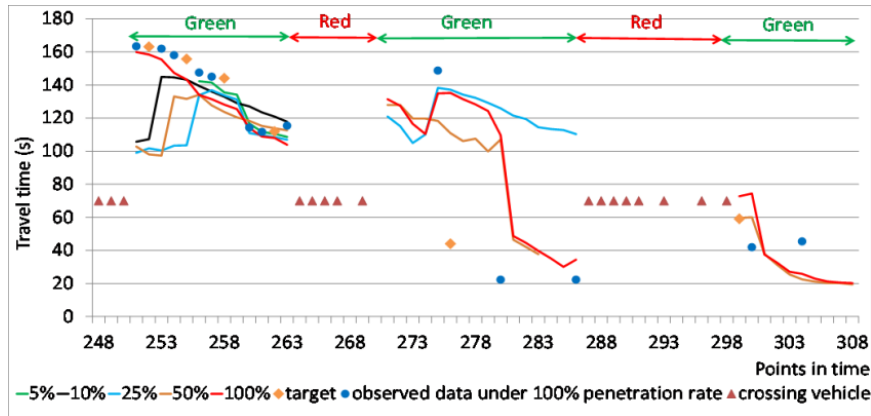
The accuracy for comparing methods under different penetration rates is shown in Table 3.5. Because the coverage rates of the KNN-based and the PF-based models were remarkably low when the penetration rate was no more than 10%, results are not shown. The KNN-based model outperformed the proposed model when the penetration rate was remarkably high, but this advantage was not obvious. When the penetration rate decreased, the accuracy of the proposed model became better than that of the KNN-based and the PF-based model. Considering the coverage rate, the performance of the proposed model was better than the KNN-based and the PF-based models under different penetration rates. As for  $PM_\beta$ , there was little difference between the proposed model and  $PM_\beta$ , so crossing vehicles did not

influence the accuracy when data from the object vehicle were plentiful. Nevertheless, if the object vehicle was unavailable, crossing vehicles could provide information to make predictions at a similar level of accuracy, so it is still necessary to consider crossing vehicles to achieve a high coverage rate.

Some examples of predictions by the proposed model under different penetration rates are shown in Figure 3.8. The orange dots (target) represent the travel times for prediction, whereas the blue dots represent observed data during prediction under a 100% penetration rate. For simplicity, the travel time of the crossing vehicle was changed to 70 s because the value of travel time does not affect the prediction. Because there is no vehicle passing through the intersection when the signal is red, there is no prediction during the red phase where the crossing vehicles appear successively. If the prediction during the red phase is needed, for example, for the route choice task, the prediction at the beginning of the following green phase can be used because it includes the stopping time at the intersection.



(a)



(b)

**Figure 3.8 Predictions of the proposed model under different penetration rates: (a) Time points from 10 to 70; (b) Time points from 248 to 308**

The proposed model could make predictions under most traffic conditions if the penetration rate was no less than 25%, but it might not be useful under unsaturated traffic conditions if the penetration rate was lower than 25%. For example, when congestion was disappearing, the proposed model could not work at the low penetration rate in Figure 3.8 (b). The proposed model could reflect the fact that link travel time decreases after the signal turns green and the increase in travel time at the beginning of the green phase due to the stop at the intersection, especially under unsaturated traffic conditions, as shown in Figure 3.8 (a). When the penetration rate was high, the proposed model could trace the change in travel time resulting from congestion using its resampling process. However, the proposed model could not react to the sudden change in traffic conditions promptly unless there were data observed after the change. For example, in Figure 3.8 (b), from time points 273 to 283, the congestion started to disappear, but the proposed model did not reflect the drop in travel time until new data were observed. However, the resampling process sometimes made the prediction overreact to some unexpected change in travel time, as shown in time points 55 to 60 in



Figure 3.8 (a).

### **3.4 Summary**

Generally speaking, the proposed model had a high coverage rate and stable performance under different penetration rates because it used the information from both object vehicles and crossing vehicles as well as a resampling process to trace the change in travel time due to unexpected events. In this study, although the crossing vehicle data did not contribute to the accuracy improvement, they can make travel time prediction at the same level as using the data from the target link. Therefore, the proposed model can significantly enhance the coverage rate when applying to the urban networks where the penetration rate of probe vehicles is low. Since the distribution of probe vehicles is not uniform in the real world, most researches have difficulty in practical application because they have limited coverage rates. However, the proposed method can predict link travel time in the form of distribution on the whole urban network for searching the most reliable route wherever the penetration rate of probe vehicles is low. Furthermore, the travel time distribution can provide more information than a weighted summation in practical applications such as travel time reliability analysis.

# **CHAPTER 4 Applying Dynamic Time Warping to Expand the Prediction Horizon using Disaggregate Probe Data**

## **4.1 Introduction**

In the last chapter, we enhanced the coverage rate when predicting travel time at urban networks, but the proposed model can only predict travel time within a signal cycle, so it is necessary to expand the prediction horizon for real-world application. As introduced in chapter 2, the pattern recognition model is more suitable than other types of models for practical application at urban networks. However, most pattern recognition models have to use aggregated data because they compare the alignment between two time-sequential groups of data, which must be sorted in exactly the same time interval and have exactly the same length.

To avoid the constraints of data aggregation, which were stated in chapter 1, this chapter developed a new pattern recognition model based on DTW to predict the short-term link travel time at urban networks using disaggregate probe data. The proposed model aimed to expand the prediction horizon and reflect temporal changes in travel time, so the forecasting step was based on the traffic signal cycle instead of a fixed time interval. To show the capability of the proposed model in practical application, it was compared with the classic KNN model and the naïve model under two real-world cases. In real-world cases, probe data were collected from the taxi in Nagoya, Japan within two months. In addition, the computer simulation under the same settings as in chapter 3 was also used to test the three models under different penetration rates.

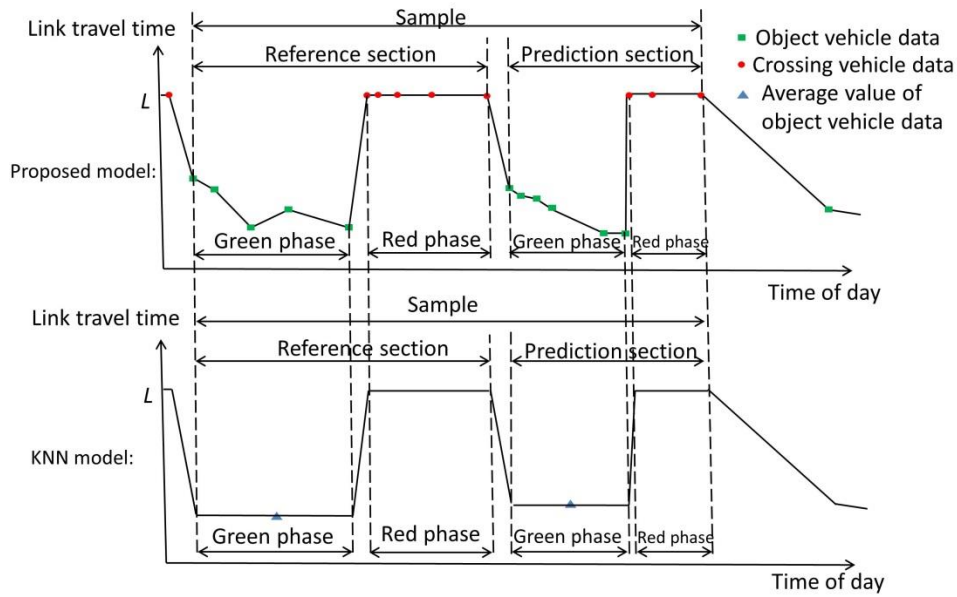
## **4.2 Methodology**

DTW was originally developed to classify nonlinear time-sequential sequences with different lengths and has long been used in speech pattern recognition (Muller, 2007). DTW has been applied successfully to various fields such as data mining (Berndt and Clifford, 1994) and bioinformatics analysis (Aach and Church, 2001). In the field of transportation, DTW was employed for out-of-sequence traffic classification by comparing two traffic flows with different lengths (Yan et al., 2013). Another application was to estimate the vehicle speed with a distorted magnetic signature due to behaviors such as acceleration and deceleration within the monitoring distance (Zhang et al., 2017).

Like the last chapter, this chapter was still focused on object vehicles that go straight through the intersection and the urban link was defined by the segment of a roadway between two adjacent signalized intersections. Consequently, the link travel time consists of the running time on the link and the stopping time at the downstream signalized intersection. To reflect the influence of traffic signals, crossing vehicles were also recorded. The time stamp is defined as the time when vehicles leave the link at the downstream signalized intersection.

### **4.2.1 Sample selection**

Similar to the KNN model, the proposed model compares samples (i.e., link travel time sequences) from the historical database with the real-time sample. Figure 4.1 shows an example of a sample in the proposed model, along with the corresponding example in the KNN model.



**Figure 4.1 Examples of samples with two signal cycles**

One sample consists of two sections: a reference section and a prediction section. The reference section is used to make a comparison between the historical sample and the real-time sample, whereas the prediction section is used to produce a prediction. In this chapter, the length of each section is measured by the number of signal cycles. A signal cycle is divided into two phases: a green phase when the object vehicle can move through the intersection and a red phase when crossing vehicles can move through the intersection. The continuous object vehicle data stream is used to represent the green phase, whereas the continuous crossing vehicle data stream is used to represent the red phase. Because the value of the crossing vehicle is not of interest, it is set as  $L$ , which is significantly greater than any possible value of the object vehicle data.

In the proposed model, the travel times from both object vehicles and crossing vehicles are directly recorded in the same link travel time sequence to represent the sample, whereas average travel times of the object vehicles are used in the KNN model. For both sections, the

start point is the first observation in the continuous object vehicle data stream, and the end point is the last observation in the continuous crossing vehicle data stream. A reference section must contain more than two object vehicle observations to avoid a significant difference in length (Muller, 2007). There are no other observations between the two sections in the same sample. Two conditions for collecting the samples follow:

1. If the current sampling time is in the red phase, the current crossing vehicle observations are at the end of the sample; the observations before it should be recorded until the predefined length of each section is reached.
2. If the current sampling time is in the green phase, one crossing vehicle observation is added at the end of the sample, and the observations before it are recorded in the same way as in condition 1.

For the historical sample, the prediction section contains only one signal cycle, whereas the reference section can contain successive signal cycles. For comparison with the historical sample, the reference section in the real-time sample should have the same length as in the historical sample. On the other hand, to simulate the multi-step prediction, the prediction section in the real-time sample contains successive signal cycles.

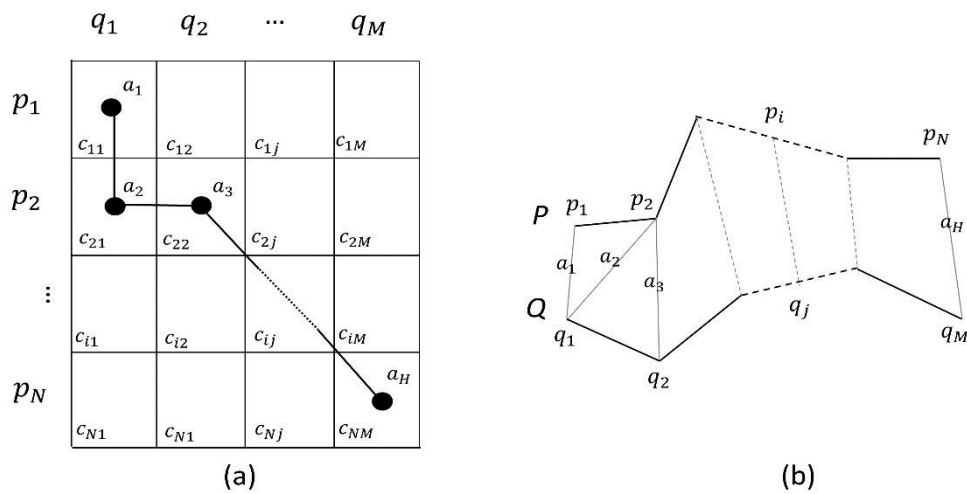
#### **4.2.2 DTW algorithm in the proposed model**

In the proposed model, the classic DTW algorithm is modified to compare reference sections of the historical sample and the real-time sample, e.g.,  $P = (p_1, p_2, \dots, p_N)$  and

$Q = (q_1, q_2, \dots, q_M)$ . To compare the different travel times between the two sections, a local cost measure is needed. There are various methods for calculating the local cost (Muller, 2007), such as the Manhattan distance and the Euclidean distance. Eq. (4-1) is used in the proposed model to calculate the local cost.

$$c_{ij} = (p_i - q_j)^2, \quad i \in N, j \in M \quad (4-1)$$

A local cost matrix, in which each element represents the local cost  $c_{ij}$  between the points  $p_i$  and  $q_j$ , is built, and several optional alignments between  $P$  and  $Q$  can be found. An optional alignment is represented by a warping path  $A = (a_1, a_2, \dots, a_H)$  with  $a_h = (i, j)$  for  $h \in [1, H]$ , which is shown in Figure 4.2.



**Figure 4.2 Illustrations of a warping path**

There are two basic constraints on the warping path selection (Muller, 2007):

1. Boundary constraint:  $a_1 = (1,1)$  and  $a_H = (N, M)$ .
2. Unit step-size constraint: For any  $h \in [1, H - 1]$ ,  $a_{k+1} - a_k \in \{(1,0), (0,1), (1,1)\}$ .

The boundary constraint makes sure that the sequences are compared from the beginning to the end. The unit step-size constraint makes sure that all elements in the sequence are checked so that each element in the warping path is unique. The unit step-size constraint gives rise to another constraint called monotonicity. In other words, for any  $h \in [1, H - 1]$ , if  $a_{h+1} = (i, j)$  and  $a_h = (i', j')$ , then  $i \geq i'$  and  $j \geq j'$ . The monotonicity constraint ensures that there is no step back while searching for the warping path.

The purpose of the DTW is to ascertain the shortest warping path between the sections  $P$  and  $Q$ . Each path has its own accumulated cost, which is the summation of each local cost corresponding to each path obtained by using Eq. (4-2).

$$C_A(P, Q) = \sum_{h=1}^H \alpha_h c_{ij}(a_h), \quad a_h \in A \quad (4-2)$$

$$\alpha_h = \begin{cases} 1, & a_h - a_{h-1} \in \{(1,0), (0,1)\} \\ 2, & a_h - a_{h-1} = (1,1) \end{cases}, \quad a_0 = (0,0) \quad (4-3)$$

The coefficient  $\alpha_h$  is calculated by Eq. (4-3) to avoid the preference of choosing the diagonal path. The shortest warping path is the one with the minimum accumulated cost, i.e.,  $DTW(P, Q) = \min\{C_A(P, Q) | A \text{ is a warping path between sequences } P \text{ and } Q\}$ , which represents the similarity between the two sections.

If the accumulated cost of each possible warping path is calculated, the shortest path can be found. However, it is computationally expensive to do so, particularly when the local cost matrix is large. However, there is an existing method to find the shortest warping path without calculating the costs of all the possible paths (Muller, 2007; Zhang et al., 2017). The minimum accumulated cost can be calculated using the algorithm listed in Table 4.1. First, a

local cost matrix between the sections  $P$  and  $Q$  is built (Lines 1–2). Then, an additional column and an additional row are added to the original local cost matrix (Lines 3–5). Finally, the cost of the shortest warping path between the sections  $P$  and  $Q$  is calculated using Eq. (4-4) (Lines 6–13).

**Table 4.1 Modified classic DTW algorithm**

---

1.	Build an $N$ -by- $M$ local cost matrix $C$ between sequences $P = (p_1, p_2, \dots, p_N)$ and		
2.	$Q = (q_1, q_2, \dots, q_M)$		
3.	Add an additional row, where $c_{i0} = \infty, i > 0$		
4.	Add an additional column, where $c_{0j} = \infty, j > 0$		
5.	$c_{00} = 0$		
6.	For $i = 1$ to $N$		
7.	For $j = 1$ to $M$		
8.			
9.		$DTW(p_i, q_j) = \min \begin{cases} DTW(p_{i-1}, q_j) + c_{ij} \\ DTW(p_i, q_{j-1}) + c_{ij} \\ DTW(p_{i-1}, q_{j-1}) + 2c_{ij} \end{cases} \quad (4-4)$	(4-4)
10.			
11.	End For		
12.	End For		
13.	$DTW(P, Q) = DTW(p_N, q_M)$		

---

Table 4.2 presents an example of searching for the shortest warping path between the reference sections of two samples:  $(70, 20, 15, L, L, 90, 88, L)$  and  $(68, 60, 26, 10, L, L, L, 82, 77, 69, L, L)$ . The reference sections of both samples contain two signal cycles. The shaded portion from the upper left to the lower right represents the shortest warping path. In Table 4.2, elements represent the local costs calculated using Eq. (4-1), except for those in the first column and the first row. The local cost matrix can be divided into several sub-matrices using the crossing vehicle data  $L$  as long as the value of  $L$  is sufficiently high. The sub-matrices, through which the shortest warping path traverses, represent the local cost matrices between the travel time sequences in the corresponding



signal phases. When the penetration rate is high, it is necessary to record the crossing vehicle data so that the travel time sub-sequence in one sample can be compared with the one in the corresponding signal phase in another sample if there is more than one signal cycle in the reference section. When the penetration rate is low, the crossing vehicle might be misleading. Therefore, the shortest warping path between the reference sections of two samples without recording the crossing vehicle should also be found.

**Table 4.2 Example of searching the shortest warping path in the local cost matrix**

	70	20	15	$L$	$L$	90	88	$L$
68	4	2304	2809	$L^2$	$L^2$	484	400	$L^2$
60	100	1600	2025	$L^2$	$L^2$	900	784	$L^2$
26	1936	36	121	$L^2$	$L^2$	4096	3844	$L^2$
10	3600	100	25	$L^2$	$L^2$	6400	6084	$L^2$
$L$	$L^2$	$L^2$	$L^2$	0	0	$L^2$	$L^2$	0
$L$	$L^2$	$L^2$	$L^2$	0	0	$L^2$	$L^2$	0
$L$	$L^2$	$L^2$	$L^2$	0	0	$L^2$	$L^2$	0
82	144	3844	4489	$L^2$	$L^2$	64	36	$L^2$
77	49	3249	3844	$L^2$	$L^2$	169	121	$L^2$
69	1	2401	2916	$L^2$	$L^2$	441	361	$L^2$
$L$	$L^2$	$L^2$	$L^2$	0	0	$L^2$	$L^2$	0
$L$	$L^2$	$L^2$	$L^2$	0	0	$L^2$	$L^2$	0

### 4.2.3 Prediction

In the classical KNN model, the average travel time for object vehicles in each signal cycle is used. The prediction  $\tilde{t}$  is given by the weighted summation of  $\tilde{t}_k$  from the first  $K$  historical samples with the highest weight, as shown in Eq. (4-5) and Eq. (4-6):

$$W_k = 1 / \sqrt{\sum_r^{N_r} (\bar{T}_o^r - \bar{T}_k^r)^2} \quad (4-5)$$

$$\tilde{t} = \sum_k^K \tilde{t}_k W_k / \sum_k^K W_k \quad (4-6)$$

where  $W_k$  is the weight,  $\bar{T}_k^r$  is the average travel time for the  $r$ th signal cycle in the reference section of the historical sample,  $\bar{T}_o^r$  is the average travel time for the  $r$ th signal cycle of the real-time sample,  $Nr$  is the number of signal cycles in the reference section, and  $\tilde{t}_k$  is the average travel time for the prediction section of the historical sample.

Some pieces of information are lost when using the average travel time in a signal cycle. For example, the link travel time tends to decrease after the traffic signal turns green because vehicles leaving in the subsequent green phase might pass through the intersection without a stop. In the proposed model, the prediction is based on the link travel time sequence instead of one single value. Because the exact length and start time point of a signal phase is unknown, it is unreasonable to merge the travel time sequences from the prediction section of different historical samples. Therefore, the prediction section of the historical sample with the highest similarity is used directly for the prediction. The similarity is calculated by Eq. (4-7):

$$S_r = 1/\sqrt{\min\{DTW(P, Q)/l, DTW(P', Q')/l'\}} \quad (4-7)$$

where  $P$  is the reference section of the historical sample with the crossing vehicle data,  $Q$  is the reference section of the real-time sample with the crossing vehicle data,  $l$  is the length of the shortest warping path between  $P$  and  $Q$ ,  $P'$  is the reference section of the historical sample without the crossing vehicle data,  $Q'$  is the reference section of the real-time sample without the crossing vehicle data, and  $l'$  is the length of the shortest warping path between  $P'$  and  $Q'$ . As mentioned above, the prediction section of the historical data contains only one signal

cycle, implying that predictions can be made only in the next signal cycle. For the multi-step prediction, a new reference section of the real-time sample is made at each step by removing the first signal cycle of the reference section and adding the current prediction to the end of the reference section. Then, the prediction at the next step can be made by comparing the historical samples and the real-time sample with the new reference section.

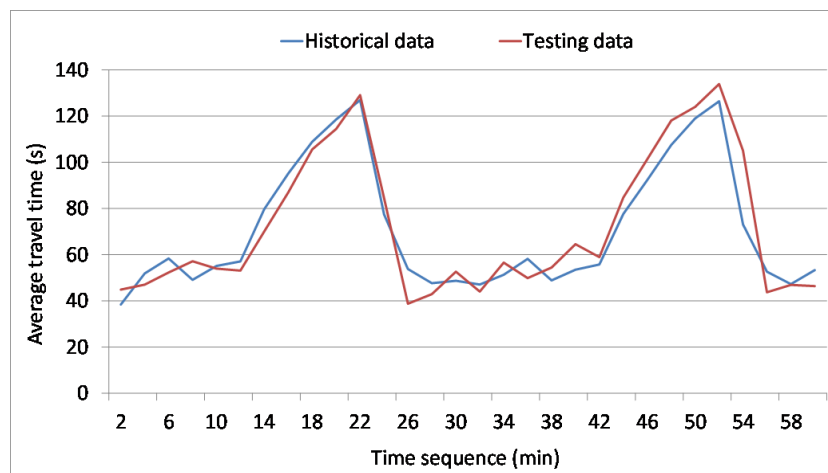
## **4.3 Experiments**

### **4.3.1 Data description**

Currently, probe data are usually collected from special vehicles (e.g., taxis and buses), so the penetration rate of vehicles that can send probe data is usually low. This chapter used probe data collected from taxis, which means that it also suffers from the problem of low penetration rate. Therefore, the computer simulation under the same settings as in the last chapter was employed to evaluate the proposed model under different penetration rates.

However, in this chapter, five trials were used as the testing data and the remaining 25 trials made up the historical database. The amount of data in the testing database was 1,475, whereas the amount of data in the historical database was 7,388. The average link travel time in the test database was 82.0 s, whereas that in the historical database was 86.6 s. Figure 3.3 and Figure 4.3 show the distribution of travel time and the changes in travel time on the target link. Fusco et al. (2016) showed that the connection between the traffic signals and the individual vehicle speeds can be inferred by the distribution of individual vehicle speed. For the same

reason, the connection between the traffic signals and the travel times can be inferred by the distribution of travel time. For example, the peak at approximately 60 s represents data from vehicles having a stop at the intersection, whereas the peak at approximately 110 s represents data from vehicles having at least one stop at the intersection. In this manner, the traffic condition was defined as congested when the average travel time was greater than 80 s in the simulation. Congestion was observed from the 14th min to the 24th min and from the 44th min to the 54th min.

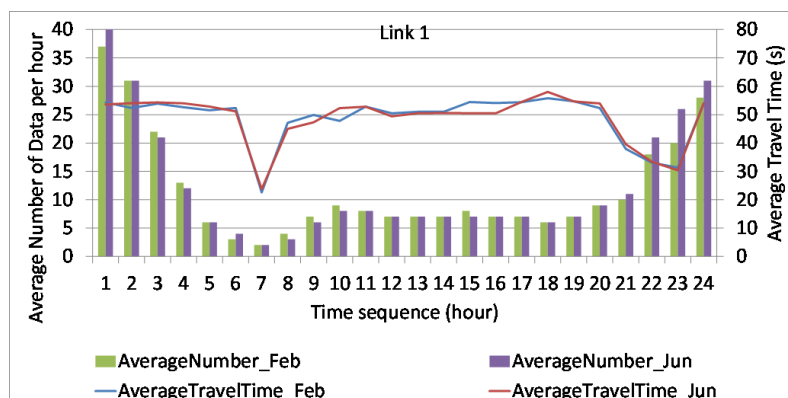


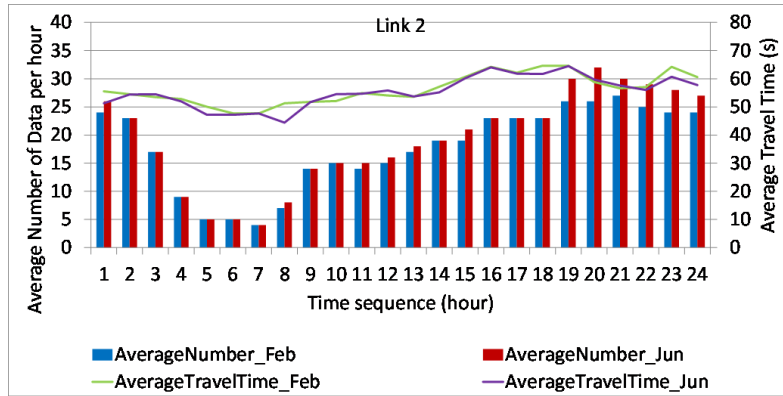
**Figure 4.3 Changes in travel time on the target link in the computer simulation**

The probe data used in the real-world cases were obtained from taxis in Nagoya, Japan, in February and June 2015 (58 days in total). Each data record represented a taxi that passed through a link and included information such as the times the taxi entered and exited the link, the ID and length of the link, the ID of the next link, and the latitude and longitude of the endpoints for each link. Data pertaining to the stopping of the taxi on the link to pick up or deliver passengers were discarded. Because taxis only accounted for a small portion of all vehicles on the road, some of the links had insufficient data. Consequently, two links with a relatively large amount of data were selected from the entire network as target links. One link

(Link 1) was a 209-m-long west-east link on Hirokoji Street, which connects the city center and a suburban area. The other one (Link 2) was a 223-m-long north-south link on Ootsu Street at the city center. Both of the links connect signalized intersections.

The amount of data for Link 1 was 23,815, among which the number of data of taxis moving straight was 17,478. The amount of data for Link 2 was 47,001, and the number for taxis going straight was 26,586. The number of data of taxis going straight for both links was overwhelming compared to all possible turning choices. Link 1 consists of a right-turn lane, a left/straight optional lane and two straight lanes. Link 2 consists of a right-turn lane, a left-turn lane and two straight lanes. Therefore, vehicles going straight in the real-case received little or no influence from vehicles with other turning behaviors. For simplicity, it is assumed that vehicles going straight on the left/straight optional lane have the same characteristics as vehicles on the straight lane, so it is reasonable to focus on predicting travel time of vehicles going straight as in the simulation. Figure 4.4 shows the changes in the average number and travel time of taxis going straight per hour during one day for both links in February and June.





**Figure 4.4 Changes in average number and travel time of taxis going straight during one day.**

The situations in February and June were similar for both links. Although both links had more data than most other links in the network, the amount of data was still insufficient because of the low penetration rate. The average number of taxis going straight per hour for Link 2 was approximately 19, compared to approximately 300 in the simulation. Therefore, the penetration rate of Link 2 was approximately 6%, and this rate was even worse for Link 1. The average travel time for Link 1 was 49.1 s, whereas that for Link 2 was 57.4 s.

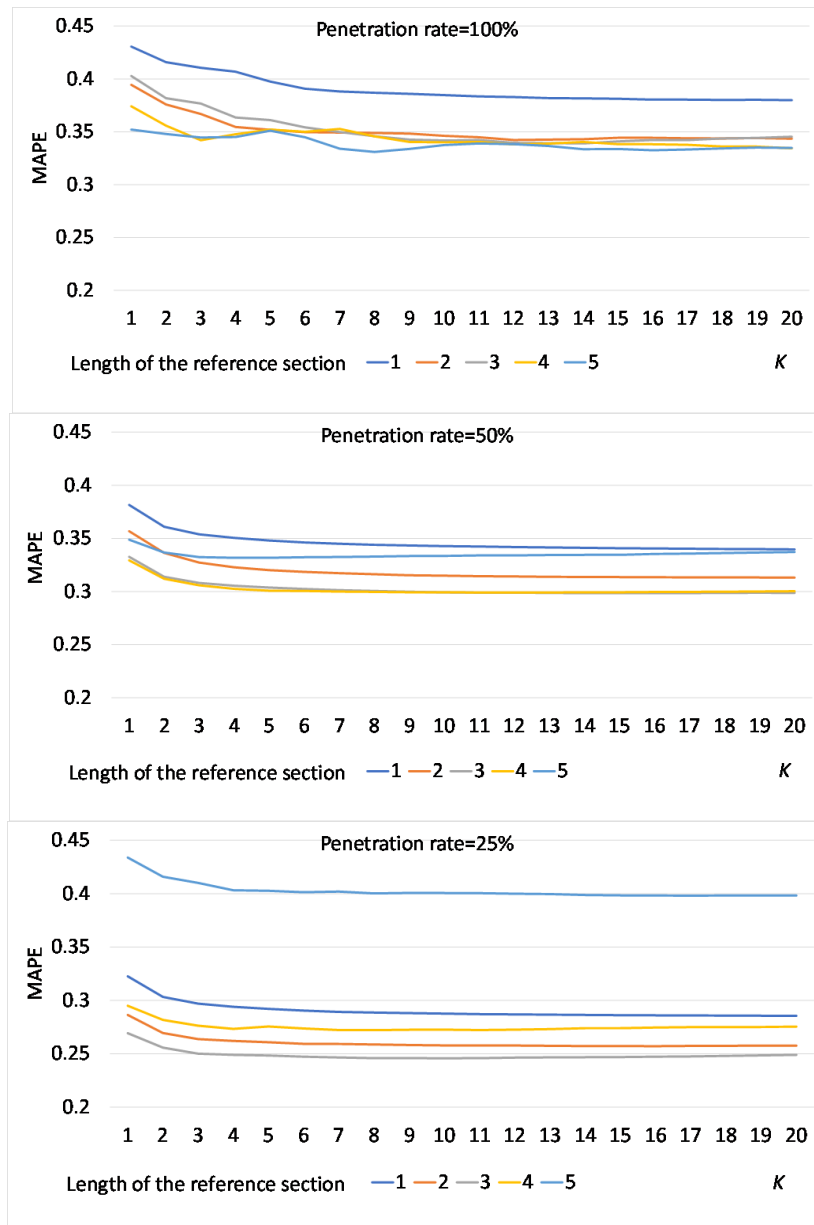
### 4.3.2 Parameter calibration

As mentioned previously, the prediction in the proposed model is based on the travel time sequence instead of one single value. This makes it possible to reflect the changes in travel time during a signal cycle. Because the time interval and sample length of the forecasting result differ from the target, DTW is also used to compare the forecasting result and the target. As shown in Figure 4.2, the travel time in one sequence can be matched with the travel time in another according to the warping path. For example, in Figure 4.2, if  $P$  represents the forecasting result and  $Q$  represents the target, then  $p_1$ ,  $p_2$  refers to  $q_1$ , and  $p_2$  refers to  $q_2$ . Therefore, according to the shortest warping path, the travel time in the target can be represented by the average value of the corresponding elements in the

forecasting result. For example, if the forecasting result is  $(68, 60, 26, 10)$  and the target is  $(70, 20, 15)$ , the forecasting result can be transformed into  $(64, 26, 10)$ . Aside from the KNN model, the naïve model, which uses the average value of the historical data as the forecasting result, was also used to compare with the proposed model. The average value used in the KNN model or in the naïve model was considered as a forecasting result containing only one element. The MAPE and RMSE were used to measure the accuracy of the proposed model.

The simulation data were used to calibrate the parameters. The proposed model and the KNN model were tested under different penetration rates in the simulation. If more than 75% of the data were reduced, the number of effective samples would be insufficient for parameter calibration. Therefore, the data were randomly reduced by 50% and 75% to simulate conditions of 50% and 25% penetration rates, respectively. When calibrating the parameters, the prediction horizon was fixed as one signal cycle.

For the KNN model, two parameters required calibration: the length of the reference section, which influences the sample size and the information provided by the sample, and the value of  $K$ , which is the number of historical samples used for prediction. Figure 4.5 shows the MAPE of the KNN model with different combinations of reference section length and  $K$  under different penetration rates.



**Figure 4.5 MAPE of the KNN model with different combinations of reference section length and K under different penetration rates**

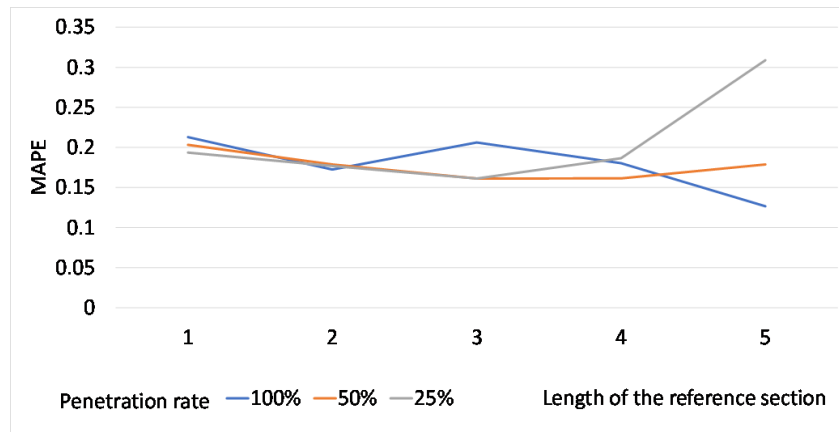
If the reference section is too short, there might not be enough information for comparison. On the other hand, if it is too long, the information might be redundant, making it difficult to collect effective samples. Therefore, when the penetration rate is not 100%, the MAPE for a length of 1 or 5 is not as good as that for a length ranging from 2 to 4. The length was set as 3 because this length results in stable performance under different penetration rates. When  $K$  increases, the MAPE decreases and then becomes stable.  $K$  was set as 8 because the



MAPE is best under this condition, irrespective of the penetration rate.

For the proposed model, only the length of the reference section needs to be determined.

Figure 4.6 shows the MAPE of the proposed model with different reference section lengths under different penetration rates.

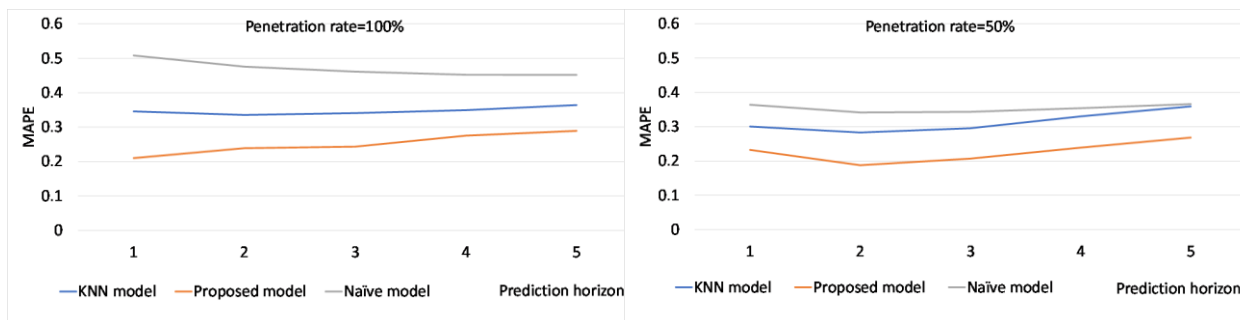


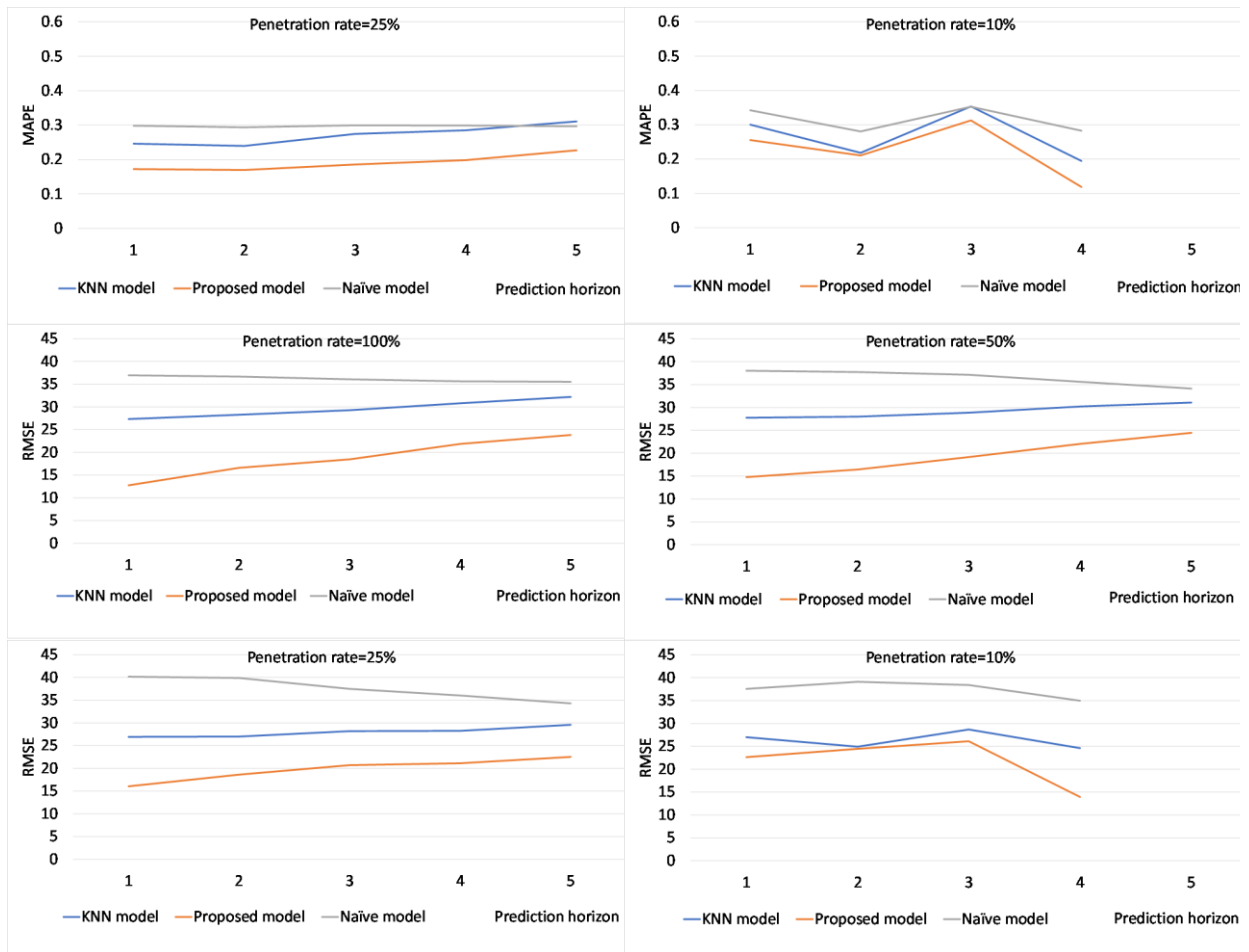
**Figure 4.6 MAPE of the proposed model with different reference section lengths under different penetration rates**

The MAPE fluctuates slightly when the penetration rate is 100%. The MAPE decreases at first and then increases under the other two penetration rates. The MAPE decreases because the additional length is expected to help provide more information. The increase in the MAPE is due to the lack of crossing vehicle data, which can separate the signal cycles under low penetration rates. If the data are not separated into corresponding signal cycles, more data lead to more redundancy. This can also explain why the increase in the MAPE under the 25% penetration rate is faster than that under the 50% penetration rate. Therefore, to ensure sufficient samples and that the proposed model can attain a stable accuracy, the length was set as 2.

### 4.3.3 Results and discussion

The proposed model, along with the KNN and naïve models, was tested against the simulation and real-world cases. Because the length of the reference section of the real-time samples used in the proposed model was shorter than that in the KNN model, the number of real-time samples in the proposed model was larger. Therefore, when testing the proposed model, the same real-time samples as in the KNN model were used to make sure that the real-time sample size was the same, but only the last two signal cycles were used when comparing the reference sections. In the simulation, multi-step predictions were made using the models under different penetration rates. As mentioned previously, there must be more than two object vehicle observations in each signal cycle of a sample. However, when the penetration rate is 10%, each signal cycle of the samples contains at least two object vehicle observations to make sure there are enough samples. Despite this, there is no qualified sample with a prediction horizon of five when the penetration rate is 10%. Figure 4.7 shows the accuracy of the three models under different penetration rates in terms of MAPE and RMSE.





**Figure 4.7 Accuracy of different models under different penetration rates in the simulation**

The proposed model outperforms the other two models irrespective of the penetration rate because it can reflect the changes in travel time during a signal cycle. When the prediction horizon increases, the accuracy of the proposed model deteriorates. This occurs because it cannot catch up with the change in traffic time in the long term if there are no new data to update the current traffic condition. However, when the penetration rate is 10%, the accuracy fluctuates. The accuracy of the naïve model is relatively stable under different prediction horizons, except for the case of the 10% penetration rate. Therefore, it is reasonable to conclude that the fluctuation comes from the bias of samples under the low penetration rate.

When the penetration rate decreases, the MAPE values of all three models tend to decrease because the sample diversity decreases and the predictable range shrinks. We now consider samples with a prediction horizon of one as an example, as listed in Table 4.3.

**Table 4.3 Sample size and composition**

Penetration rate (%)	100	50	25	10
Sample size	600	235	70	26
Ratio of samples in congestion (%)	46	89	96	100

Because each sample must contain four successive signal cycles and each signal cycle must contain at least two object vehicle observations according to the previous setting, if the penetration rate decreases, the sample size also decreases. For the same reason, successive signal cycles that contain sufficient object data tend to appear congested under low penetration rates. Therefore, both the historical and real-time samples tend to be collected when the traffic is congested under low penetration rates, implying that the proposed and KNN models can hardly be applied to an uncongested traffic condition. If the samples are taken from similar traffic conditions, it is likely that they exhibit similar patterns and have similar travel times, making it easy to achieve accurate predictions under low penetration rates. Moreover, because the samples are mainly obtained from congested conditions in which the travel time is considerable, the MAPE tends to be low. Concerning the computation cost, the program was built by C++ and run by a computer with a 2.5 GHz Intel Core i7 processor and 16 GB of 1600 MHz DDR3 memory. The running time for one prediction in the proposed model was approximately 0.45 s. In the KNN model, it was approximately 0.02 s.

As for the real cases, the parameter setting was the same as that in the simulation. Because the penetration rate was extremely low, the prediction horizon was fixed as one signal cycle and the length of the reference section was set as 2 to collect sufficient real-time samples. For the proposed model, the sample sizes in Links 1 and 2 were 246 and 115, respectively, whereas for the KNN model, the sample sizes in Links 1 and 2 were 66 and 9, respectively. Therefore, data were used more effectively in the proposed model than in the KNN model. Although the amount of data in Link 1 was less than that in Link 2, the sample size in Link 1 was greater than that in Link 2. This occurred because most of the data in Link 1 were distributed during the nighttime, whereas in Link 2, the data were distributed more evenly, as shown in Figure 4.4. Leave-one-out cross-validation was performed to test the different models in Links 1 and 2, as listed in Tables 4.4 and 4.5.

**Table 4.4 Accuracy of different models in Link 1**

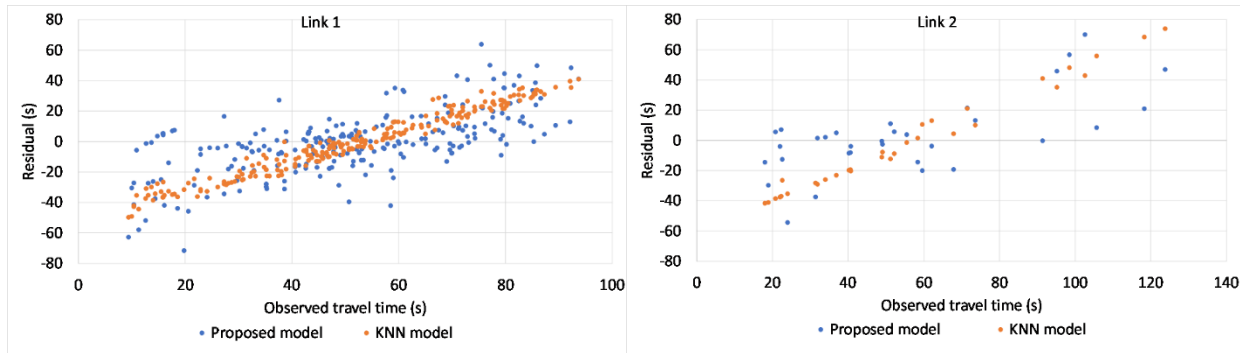
	Proposed model	KNN model	Naïve model
MAPE (%)	45.3	56.1	52.9
RMSE	18.8	20.7	20.5

**Table 4.5 Accuracy of different models in Link 2**

	Proposed model	KNN model	Naïve model
MAPE (%)	36.7	66.4	62.0
RMSE	22.2	28.9	27.1

The proposed model still outperformed the KNN model and the naïve model under extremely low penetration rates. The accuracy of the KNN model is lower than that of the naïve model because the length of the signal cycle changes for the real cases, implying that the time interval when collecting samples in the KNN model is not equal. However, the

proposed model is free from the equal time interval constraint, thus exhibiting a higher accuracy. To compare the proposed model and the KNN model further, a residual analysis was performed and the results are shown in Figure 4.8.



**Figure 4.8 Results of the residual analysis**

The residual is defined as the difference between the observed travel time and its prediction. The distribution of residuals in the KNN model tends to be linear because it uses the weighted average value as the prediction. The distribution of residuals in the proposed model shows more randomness than that in the KNN model because its prediction is in the form of a travel time sequence, which can reflect changes in travel time. However, when the observed travel time is too short or too long, the residuals of the proposed model also have a bias. This occurs because the penetration rate is so low that the proposed model cannot find similar patterns in the past to make predictions. On the whole, it is reasonable to conclude that the residual has a more desirable random distribution in the proposed model than in the KNN model.

## 4.4 Summary

In this study, a pattern recognition model based on the DTW was developed to expand the prediction horizon using disaggregate probe data when predicting short-term urban link travel time. The DTW is a method used to compare two sets of the time sequential data and is free from constraints of the equal time interval and equal sample length. In the proposed model, crossing vehicle data were introduced to divide the object data into different traffic signal cycles instead of identifying the exact signal patterns, so the changes in travel time caused by the traffic signal can be reflected. The proposed model predicted the travel time sequence in a signal cycle instead of the average travel time. The longest prediction horizon was five signal cycles, which corresponds to ten minutes. Because the proposed model took advantage of disaggregate data and because it was not required to estimate the exact signal pattern, the method can use data more effectively to achieve higher accuracy and had wider applicability than the classical KNN model and the naïve model.

# **CHAPTER 5 Intersection Priority Control for Urban Traffic Management to Reduce Traffic Congestion**

## **5.1 Introduction**

In the last two chapters, effort was devoted to expanding short-term travel time prediction to urban networks. Based on the predicted travel time, the ITS-based traffic control and management can react to traffic condition that might happen in a short future. Since the SODTA approach that solves the congestion problem at the scale of the whole network is difficult to implement in the real-world application, it is necessary to consider solving the congestion at a small scale. Therefore, this chapter proposed a LTM-based model to reduce the congestion by managing the incoming traffic flows at each intersection which is the main potential bottleneck on the urban network.

This model combined a local linear programming formulation with the LTM to minimize the congestion on the incoming links by optimizing their priority in the condition that route choices are determined in advance. The reason to use the LTM is that it is computationally efficient to manage the transition traffic flows between links connected by the intersection. The main advantage of the proposed model is that it can be added to other SODTA approaches that have different global objectives like minimizing the total emission. Moreover, the proposed model is easy to be extended from one application to another because it has no



pre-defined assumption and it is not constrained by the type of intersection. In this chapter, a computer simulation was employed to compare the proposed model with other models using different intersection priority strategies at the Sioux Falls network.

## 5.2 Methodology

### 5.2.1 Classic LTM

The LTM network consists of homogeneous links and different types of nodes, including inhomogeneous node, origin node, destination node, merge node, diverge node, and crossing node. The main reason why the LTM is more computationally efficient than the CTM is that it only focuses on the update of the cumulative number of vehicles  $N(x,t)$  at the upstream and downstream boundaries of link  $i$  which are denoted as  $x_i^0$  and  $x_i^L$  respectively.

The LTM consists of the link model and the node model. In the link model, two variables are defined—the sending flow  $S_i(t)$  and the receiving flow  $R_j(t)$ . During the time interval  $[t, t + \Delta t]$ ,  $S_i(t)$  represents the maximum number of vehicles that can potentially leave the downstream boundary of link  $i$ , whereas  $R_j(t)$  represents the maximum number of vehicles that can be received from the upstream boundary of link  $j$ . They are defined as:

$$S_i(t) = \min\{[N(x_i^0, t + \Delta t - L_i/v_{f,i}) - N(x_i^L, t)], q_{D,i}\Delta t\} \quad (5-1)$$

$$R_j(t) = \min\{[N(x_j^L, t + \Delta t + L_j/w_j) + k_j^{jam}L_j - N(x_j^0, t)], q_{U,j}\Delta t\} \quad (5-2)$$

where,

$L_i, L_j$  : length of link  $i$  and  $j$  respectively,

$v_{f,i}$  : free-flow speed of link  $i$ ,

$w_j$  : negative maximum spillback wave speed of link  $j$ ,

$k_j^{jam}$  : jam density of link  $j$ ,

$q_{D,i}$  : capacity of link  $i$  at the downstream boundary, and

$q_{U,j}$  : capacity of link  $j$  at the upstream boundary.

In the node model, three variables are defined—the turning fraction  $\beta_{ij}(t)$ , the priority fraction  $\varphi_{ij}(t)$ , and the transition flow  $G_{ij}(t)$  ( $i \in I_n, j \in J_n$ ).  $I_n$  represents the assemblage of incoming links of node  $n$ , whereas  $J_n$  represents the assemblage of outgoing links of node  $n$ .  $\beta_{ij}(t)$  represents the proportion of vehicles leaving the same incoming link  $i$  for different outgoing links, whereas  $\varphi_{ij}(t)$  represents the proportion of vehicles entering the same outgoing link  $j$  from different incoming links.  $G_{ij}(t)$  represents the maximum number of vehicles that can actually transfer from incoming link  $i$  to outgoing link  $j$  through node  $n$  during the time interval  $[t, t + \Delta t]$ . The main process of the LTM, which is to update the cumulative number of vehicles, is shown in Table 5.1.

**Table 5.1 Algorithm of updating the cumulative number of vehicles**

---

For each time step  $t$ :

- Using the link model to determine  $S_i(t)$  and  $R_j(t)$  for each link.
  - Using the node model to determine  $\beta_{ij}(t)$ ,  $\varphi_{ij}(t)$ , and  $G_{ij}(t)$  for each node.
  - For each incoming link  $i$  at node  $n$ ,  $N(x_i^L, t + \Delta t) = N(x_i^L, t) + \sum_j^{J_n} G_{ij}(t)$
  - For each outgoing link  $j$  at node  $n$ ,  $N(x_j^0, t + \Delta t) = N(x_j^0, t) + \sum_i^{I_n} G_{ij}(t)$
-

The definition of  $G_{ij}(t)$  differs according to the type of node. For the inhomogeneous node which connects one incoming link to one outgoing link,  $G_{ij}(t)$  is intuitively defined as

$$G_{ij}(t) = \min\{S_i(t), R_j(t)\} \quad (5-3)$$

For the diverge node which connects only one incoming link to two or more outgoing links, the sending flow of the incoming link is decomposed into several sub-flows denoted by  $S_{ij}(t)$  according to  $\beta_{ij}(t)$ . As mentioned before, most LTM-based SODTA approaches focused on solving the routing problem which determines the turning fraction. Similarly, in this chapter,  $\beta_{ij}(t)$  was determined by the route search according to the dynamic user optimal assignment in advance. It is assumed that vehicles at the intersection obey the first-in-first-out (FIFO) discipline, so the transition flow for one outgoing link is constrained not only by the receiving flow of this link but also other outgoing links. Consequently,  $G_{ij}(t)$  for the diverge node is defined as

$$S_{ij}(t) = \beta_{ij}(t)S_i(t) \quad (5-4)$$

$$G_{ij}(t) = \min_{j' \in J_n} \left\{ S_{ij}(t), \frac{S_{ij}(t)}{S_{ij'}(t)} R_{j'}(t) \right\} \quad (5-5)$$

For the merge node which connects two or more incoming links to only one outgoing link, the receiving flow of the outgoing link is allocated to incoming links according to  $\varphi_{ij}(t)$ . Thus,  $G_{ij}(t)$  for the merge node is defined as

$$G_{ij}(t) = \min\{S_{ij}(t), \varphi_{ij}(t)R_j(t)\} \quad (5-6)$$

Daganzo (1995) provided another method to calculate  $G_{ij}(t)$ , but it is preferred when there are only two incoming links (Hajiahmadi *et al.*, 2013), so it is not discussed here. There

are several methods for calculating  $\varphi_{ij}(t)$ , for example, many researchers (Lebacque, 1996; Gentile, 2010; Van de Weg et al., 2016; Nezamuddin and Boyles, 2014) used the fixed fraction which is proportional to the capacity of each incoming link. Except for the capacity, Jin and Zhang (2003) used fixed fraction which is proportional to the demand of each incoming link. In this chapter, a new method is proposed to calculate  $\varphi_{ij}(t)$  so as to reduce the congestion on networks.

For the crossing node which connects two or more incoming links to two or more outgoing links, it can be treated as the combination of merge and diverge nodes. Therefore,  $G_{ij}(t)$  for the crossing node is defined as

$$G_{ij}(t) = \min_{j' \in J_n} \left\{ S_{ij}(t), \varphi_{ij'}(t) \frac{S_{ij}(t)}{S_{ij'}(t)} R_{j'}(t) \right\} \quad (5-7)$$

For the origin node, it is assumed that there is one dummy incoming link which has no length but infinite capacity. The sending flow of its dummy incoming link is defined as

$$S_i(t) = N_o(t + \Delta t) - N(x_i^0, t) \quad (5-8)$$

where,

$N_o$  : cumulative traffic demand at origin  $o$ .

Consequently, the origin node can be treated as the diverge node. Similarly, it is assumed that the destination node connects to one dummy outgoing link which has no length but infinite capacity. Thus, the destination node can be treated as the merge node which can receive all flows from incoming links. Readers who need more details about the classic LTM can refer to

Yperman's (2007) Ph.D. dissertation.

### 5.2.2 Local linear formulation to reduce the congestion

According to Eq. (5-7), there is no guarantee that the sub-sending-flow  $S_{ij}(t)$  equals to the corresponding transition flow  $G_{ij}(t)$ . The difference between  $S_{ij}(t)$  and  $G_{ij}(t)$  results from the gap between demand and supply, and the FIFO behavior. This difference implies that there are vehicles remaining at the link which may cause the congestion. Therefore, a local linear programming formulation is proposed to reduce the congestion on the network. It minimizes the difference between the sub-sending-flow and the corresponding transition flow for each node (intersection) at each time step. The objective function is defined as:

$$\min \sum_i^{I_n} \sum_j^{J_n} [S_{ij}(t) - G_{ij}(t)] \quad (5-9)$$

Because the crossing node is a mix of merge and diverge nodes, and the origin and destination nodes can be viewed as diverge and merge nodes respectively,  $G_{ij}(t)$  for different types of nodes can be generally represented in the form of the crossing node. Therefore, if substituting Eq. (4-7) into Eq. (4-9), the local linear programming formulation can be written as:

$$\begin{aligned} & \min \sum_i^{I_n} \sum_j^{J_n} \left[ S_{ij}(t) - \min_{j' \in J_n} \left\{ S_{ij'}(t), \varphi_{ij'}(t) \frac{S_{ij}(t)}{S_{ij'}(t)} R_{j'}(t) \right\} \right] \\ & = \min \sum_i^{I_n} \sum_j^{J_n} S_{ij}(t) \left\{ 1 - \min_{j' \in J_n} \left[ 1, \frac{R_{j'}(t)}{S_{ij'}(t)} \varphi_{ij'}(t) \right] \right\} \\ & = \min \sum_i^{I_n} S_i(t) \max_{j' \in J_n} \left[ 0, 1 - \frac{R_{j'}(t)}{S_{ij'}(t)} \varphi_{ij'}(t) \right] \end{aligned} \quad (5-10)$$

subject to,

$$\sum_i^{I_n} \varphi_{ij}(t) = 1$$

Since the turning fraction is fixed by solving the routing problem in advance, this linear programming problem results in optimizing the combination of  $\varphi_{ij}(t)$  to reduce the congestion at each incoming link at the node. It can be further reformulated as a standard form:

$$\begin{aligned} & \min \sum_i^{I_n} S_i(t) \left[ 1 - \frac{R_{j^*}(t)}{S_{ij^*}(t)} \varphi_{ij^*}(t) \right] + \sum_j^{J_n} R_j(t) \left[ 1 - \sum_i^{I_n} \varphi_{ij}(t) \right] \\ \Leftrightarrow & \max \sum_i^{I_n} \frac{S_i(t) R_{j^*}(t)}{S_{ij^*}(t)} \varphi_{ij^*}(t) + \sum_i^{I_n} \sum_j^{J_n} R_j(t) \varphi_{ij}(t) \end{aligned} \quad (5-11)$$

subject to,

$$0 \leq \varphi_{ij}(t) \leq S_{ij}(t)/R_j(t) \quad (5-12)$$

$$\exists j^* \in J_n, \varphi_{ij^*}(t) R_{j^*}(t) / S_{ij^*}(t) - \varphi_{ij}(t) R_j(t) / S_{ij}(t) \leq 0 \quad (5-13)$$

$$\sum_i^{I_n} \varphi_{ij}(t) \leq 1$$

The first term of the reformulated objective function (5-11) is to make sure that the throughput of the transition flow which is most congested at the intersection is as large as possible. The second term of the reformulated objective function (5-11) is to make sure that the total available capacity of outgoing links is as large as possible. In a word, the reformulated objective function (5-11) is to maximize the throughput of the intersection. The constraint (5-12) is to make sure that the available capacity of the outgoing link is larger than the number of vehicles from the incoming link. The constraint (5-13) is to find out the most congested transition flow.

### 5.3 Experiments

In this chapter, a model that combined the classic LTM and the local linear programming formulation mentioned above is proposed to reduce the congestion on networks. To evaluate the effect of the congestion reduction, the average congestion index (ACI) is introduced. The ACI is positively related to the congestion, which means higher the ACI is, heavier the congestion is on the whole network (Sun *et al.*, 2014). Besides the ACI, the total travel time (TTT) was also used to evaluate the performance of the models. The travel time consisted of the running time and waiting time on the link. Since the LTM algorithm calculated cumulative vehicle numbers on discrete time steps, travel time was estimated based on an interpolation procedure which was explained in Yperman's (2007) Ph.D. thesis. The ACI is defined as:

$$ACI = \sum_i^N \left( \frac{tt_i^T - tt_i^0}{tt_i^0} f_i^T \right) / \sum_i^N f_i^T \quad (5-14)$$

where,

$N$  : assemblage of links (without dummy links) on analyzed network,

$tt_i^T$  : actual travel time of link  $i$  during time period  $T$ ,

$tt_i^0$  : free-flow travel time of link  $i$ ,

$f_i^T$  : traffic flow of link  $i$  during time period  $T$ .

Three other LTM-based models with different definitions of priority fraction are used to compare with the proposed model. Their names and definitions of priority fraction are shown in Table 5.2.

**Table 5.2 Models for comparison**

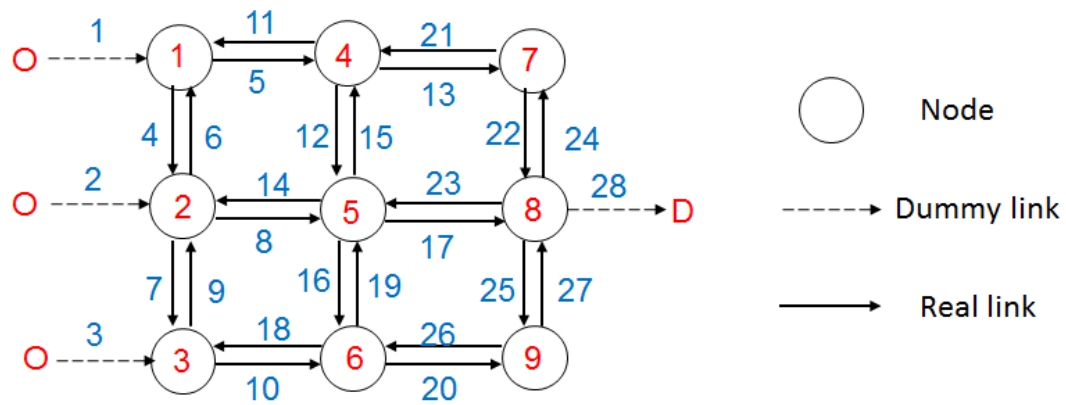
	Definition of priority fraction
Fairness Model	$\varphi_{ij}(t) = 1/I_n$
Capacity Model	$\varphi_{ij}(t) = Q_i / \sum_i^{I_n} Q_i$ ( $Q_i$ : Capacity at the downstream boundary of link $i$ )
Demand Model	$\varphi_{ij}(t) = S_i(t) / \sum_i^{I_n} S_i(t)$

In the fairness model, each incoming link has the same priority fraction. In the capacity model, the priority fraction is proportional to the capacity at the downstream boundary of the incoming link. In the demand model, the priority fraction is proportional to the demand of the incoming link. In this section, simulations were run at two networks to test the four models. One was a simple grid network which was aim to compare the performances of the four models at a single crossing intersection. The other one was the Sioux Falls network which was originally from the real-world urban network so that the potential of practical application for the four models can be evaluated.

### 5.3.1 Simulation at a simple grid network

The topology of the simple grid network is shown in Figure 5.1. There are three origins  $O$  and 1 destination  $D$  which are connected with nodes at the network by the dummy link. The blue number refers to the link number, whereas the red number refers to the node number. Each link only has one lane. Attributes of the simple grid network are shown in Table 5.3.





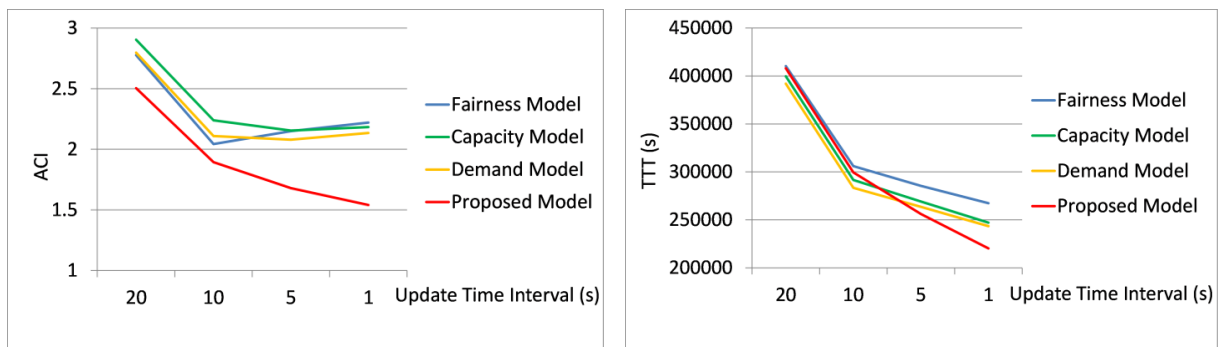
**Figure 5.1 The simple grid network**

**Table 5.3 Attributes of the simple grid network.**

Link number	Length (m)	Capacity (veh/s)	Jam density (veh/m)	Free-flow speed(m/s)
1,2,3,28	0	10000	1000	30
4,6,7,9	600	0.5	0.1	20
5,8,10,11,14,17,18,23	600	1	0.2	30
12,15,16,19,22,24,25,27	600	0.5	0.1	30

The assignment period was 300 seconds. The traffic demand for each origin during the assignment period was the same, which was 1 vehicle/s. Since vehicles cannot go through a link within one update time interval in the LTM, the update time interval should be no more than the minimum free-flow travel time. In this section, different update time intervals, which were 1s, 5s, 10s, and 20s, were tested. The simulation continued after the assignment period until all vehicles reached the destination. To reproduce the supply uncertainty, a noise that followed the standard normal distribution was added to the link capacity for both upstream and downstream boundaries. 30 sets of random seeds were tested for each model under different update time intervals. Last but not least, the route for a vehicle was decided based on the current traffic condition using the Method of Successive Average (MSA) before vehicles enter

the network, which means the turning choice at each intersection is determined. It is possible for a vehicle to search the route at each intersection based on the current traffic condition. The dynamic priority control and the route choice may affect each other but this chapter focused on the influence of the priority control on reducing congestion so the application of the route searching at each intersection will remain as a future work.



**Figure 5.2 ACI and TTT for models under different update time intervals**

Results of both the ACI and the TTT for each model under different update time intervals are shown in Figure. 5.2. When the update time interval increased, the ACI and TTT became larger because simulation with a shorter time interval can capture more changes in the traffic flow. The traffic congestion level in the proposed model was lower than the other three models, especially when the update time interval was short. However, there was no obvious difference between the travel costs of these models when the update time interval was long. To exclude the influence of stochasticity from link boundary capacity on the results, a t-test is conducted between the proposed model and other LTM-based models.

**Table 5.4 Results of the t-test between the proposed model and other LTM-based models.**

t value for the ACI				
Update time interval (s)	20	10	5	1
Fairness Model	-2.36	-1.65	-7.27	-15.63
Capacity Model	-3.55	-3.95	-7.69	-18.12

Demand Model	-3.01	-2.55	-6.53	-16.64
P value for the ACI				
Update time interval (s)	20	10	5	1
Fairness Model	0.01	0.05	0.00	0.00
Capacity Model	0.00	0.00	0.00	0.00
Demand Model	0.00	0.01	0.00	0.00
t value for the TTT				
Update time interval (s)	20	10	5	1
Fairness Model	-0.19	-0.82	-6.36	-22.08
Capacity Model	0.64	0.95	-2.75	-13.60
Demand Model	1.30	2.04	-1.78	-10.49
P value for the TTT				
Update time interval (s)	20	10	5	1
Fairness Model	0.43	0.21	0.00	0.00
Capacity Model	0.26	0.17	0.00	0.00
Demand Model	0.10	0.02	0.04	0.00

$\alpha=0.1$ , sample size=30

According to the results of the t-test in Table 5.4, the proposed model had a distinctly lower congestion level than the other three models regardless of the update time interval. The proposed model also had lower travel costs than the other three models, but when the update time interval increased, the difference in the TTT became insubstantial. It can be concluded that the proposed model has the potential of reducing traffic congestion in the real world. However, the grid network is too simple to describe the real-world urban network, so simulation at the real-world based network is a must.

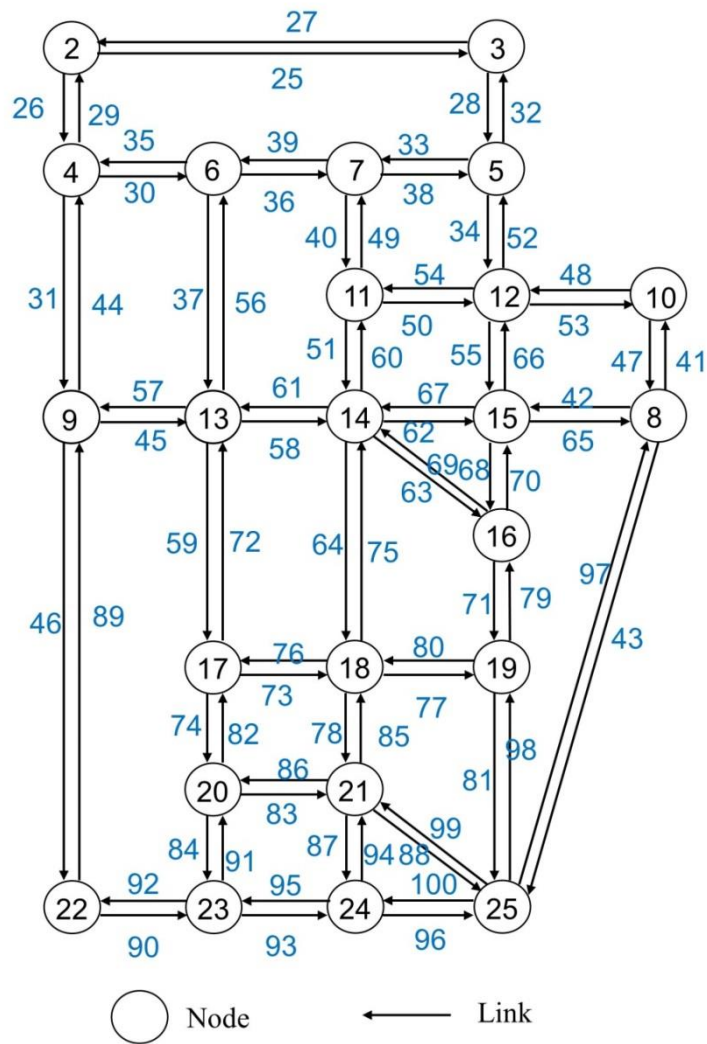
### 5.3.2 Simulation at the Sioux Falls network

To further evaluate the proposed model, the Sioux Falls network, which is widely used as a substitute for the real-world network in urban transportation research, was employed in this

section. Attributes of the Sioux Falls network are shown in Table 5.5 and its topology is shown in Figure 5.3.

**Table 5.5 Attributes of the Sioux Falls network.**

Link number	Length (m)	Capacity (veh/s)	Jam density (veh/m)	Free-flow speed(m/s)
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124	0	10000	10000	30
30,33,35,36,38,39,42,45,48,50,53,54,56,58,61,62,65,67,73,76,77,80,83,86,90,92,93,95,96,100	600	0.5	0.15	30
26,28,29,32,40,41,47,49,51,55,60,66,68,70,71,74,78,79,82,84,85,87,91,94	600	1	0.2	30
34,52	600	1.5	0.2	30
63,69,88,99	840	1	0.2	30
31,44	1200	1.5	0.2	30
37,57,59,64,72,75,81,98	1200	1	0.2	30
25,27	1800	0.5	0.15	30
46,89	2400	1	0.2	30
43,97	2460	1	0.2	30



**Figure 5.3 The Sioux Falls network**

The origin node whose node number is 1, the destination node and dummy links are necessary for iterating the LTM, but they do not actually exist, so they are not shown in Figure 5.3. There are 24 0-meter-long dummy links connecting the origin node with each node at the network so that each node can generate vehicles. Similarly, there are 24 0-meter-long dummy links connecting the destination node with each node at the network so that each node can receive vehicles. The blue number refers to the link number, whereas the black number refers to the node number. Each link only has one lane.

In this simulation, a vehicle was generated based on the Poisson distribution from each node and was randomly assigned to another node. The vehicle determined its route by the MSA before it entered the network as in the simulation of the simple grid network. The assignment period continued for 240 seconds and the vehicle generating rate for each node was 1 vehicle/s. The simulation continued after the assignment period until all vehicles reached the destination. To reproduce the supply uncertainty, a noise that followed the standard normal distribution was added to the link capacity for both upstream and downstream boundaries. For simplicity, the update time interval was fixed as 1 second in this simulation and 30 sets of random seed were tested for each model. The ACI and TTT for each model and the results of the t-test between the proposed model and other LTM-based models are shown in Table 5.6.

**Table 5.6 Results of the simulation on the Sioux Falls network.**

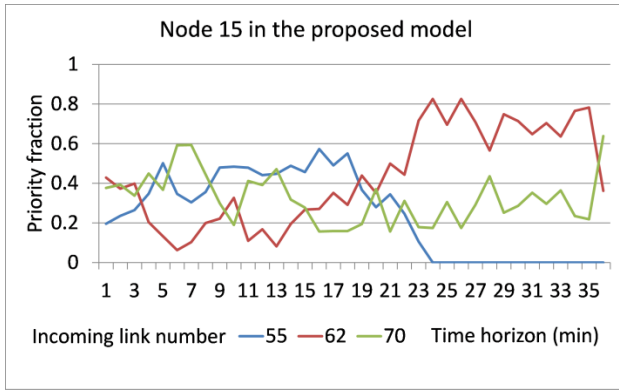
	Fairness Model	Capacity Model	Demand Model	Proposed Model
ACI	3.75	3.62	3.40	3.22
TTT (min)	43117	41354	40362	39355
t value for the ACI	-7.03	-4.19	-2.16	-
P value for the ACI	0.00	0.00	0.02	-
t value for the TTT	-10.55	-5.24	-3.30	-
P value for the TTT	0.00	0.00	0.00	-

$\alpha=0.1$ , sample size=30

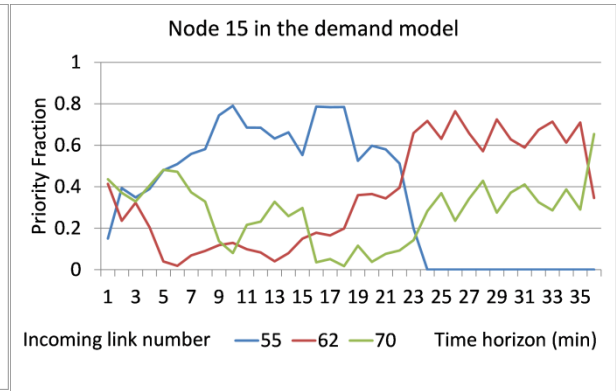
It is obvious that in this simulation the proposed model outperformed other LTM-based models in both the traffic congestion level and the travel cost. Intuitively, at a complicated urban network like the Sioux Falls network, it is better to apply dynamic priority at the intersection than the static priority. Therefore, to further compare the difference between the proposed model and the demand model, nodes that connected more than four links at the

Sioux Falls network, were selected to show the changes in the priority for the incoming links which correspond to their outgoing link. For simplicity, the priority was aggregated in one minute. Due to the paper limitation, part of the results is shown in Figure 5.4 the remaining part is shown in Appendix B.

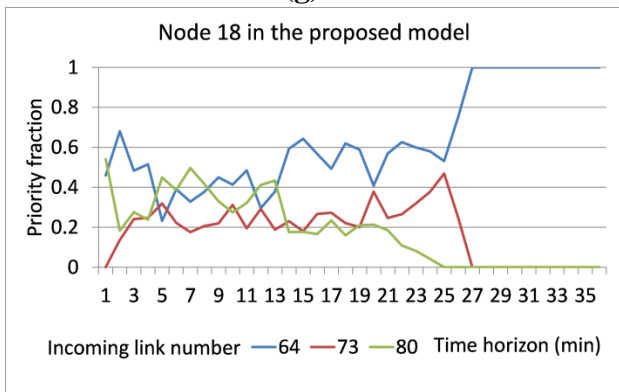




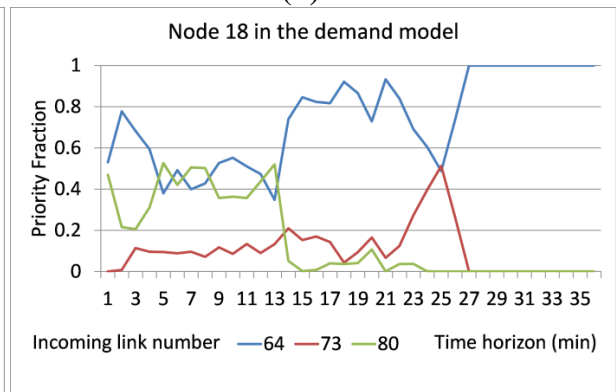
(g)



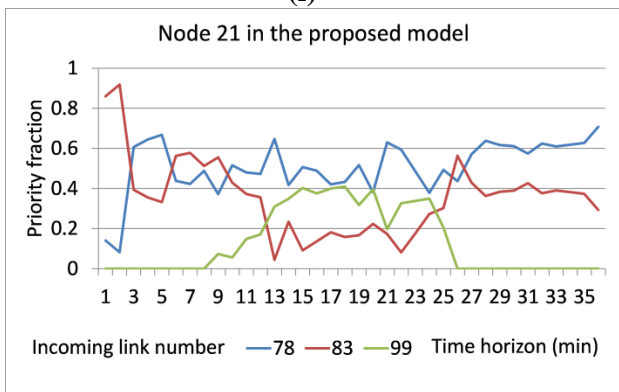
(h)



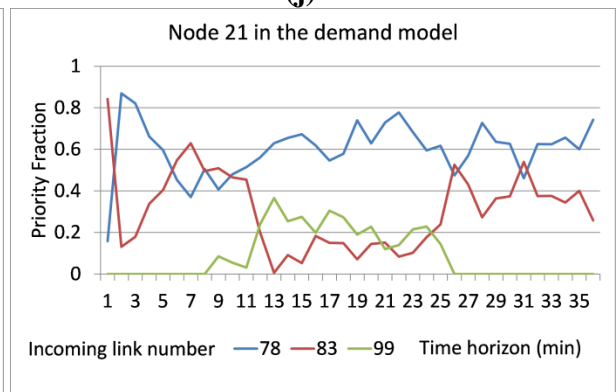
(i)



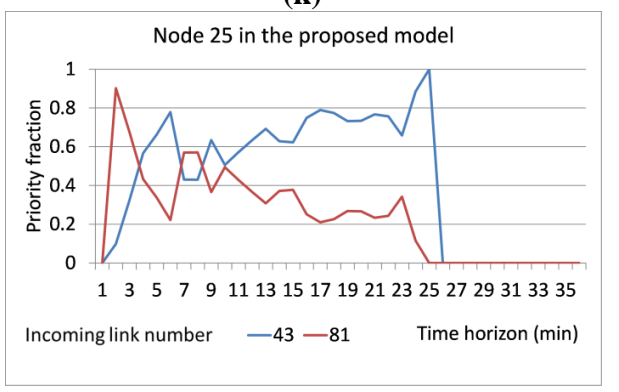
(j)



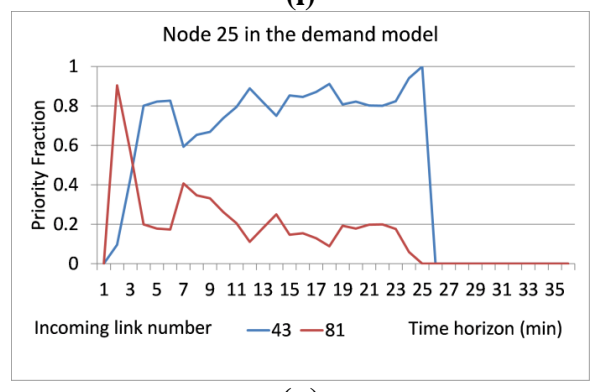
(k)



(l)



(m)





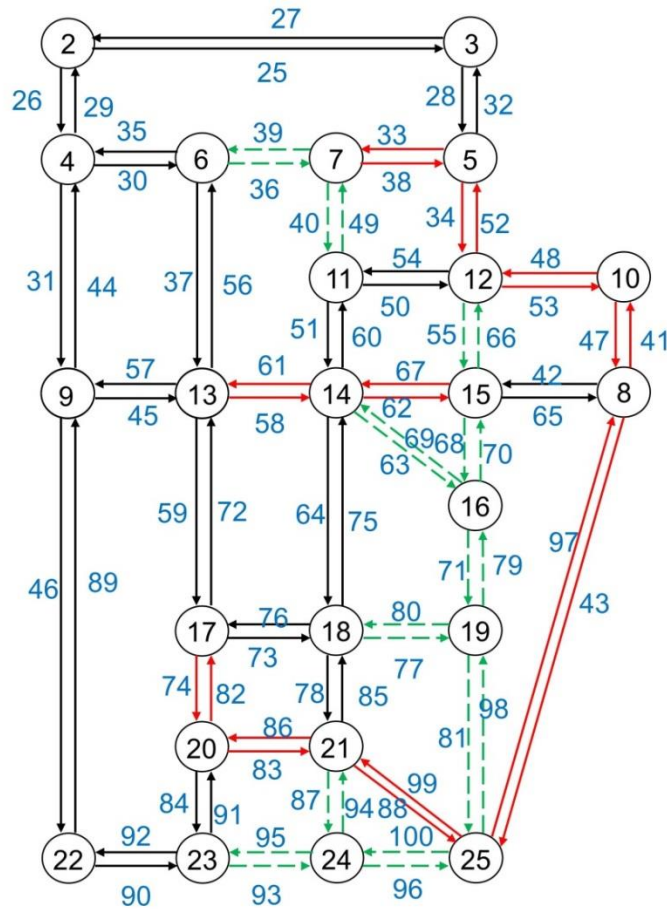
(n)

**Figure 5.4 Changes in the priority in the proposed model and the demand model**



Compared with the demand model, the proposed model tended to give less priority to the incoming link with the largest demand in most cases. In other word, incoming links were treated more equally in the proposed model because the proposed model was intended to make the maximum use of each link. If the significance level of the t-test in Section 5.2.1 was increased to 0.05, the proposed model would have a similar congestion level with the fairness model when the update time interval was 10. Therefore, it could be concluded that the proposed model is somehow fair when allocating the road resource so that vehicles are encouraged to use the whole network instead of some main roads that have large demand. The advantage of encouraging the full use of the whole network is obvious, which is the reduction of congestion level at the network. To make the full use of the network, the proposed model might even give higher priority to the incoming link which had lower demand as shown in Figure 5.4 (e) and (f). Figure 5.4 (e) and (f) also demonstrated that the proposed model can be applied to different types of the intersection. Figure 5.4 (m) and (n) shown that even though there was no vehicle from Link 88 to Link 100, the proposed model can still allocate the capacity of Link 100 to other incoming links. Therefore, the proposed model is flexible to deal with different traffic conditions in the real world.

 Accumulated traffic flow: Proposed model > Demand model in both directions  
 Accumulated traffic flow: Demand model > Proposed model in both directions



**Figure 5.5 The difference of cumulative traffic flow between the proposed model and the demand model**

Although the proposed model cannot decide the route choice, it affects the route searching by controlling the priority of each link. To further find out its influence on the route choice, the difference of cumulative traffic flow between the proposed model and the demand model is shown in Figure 5.5. Roads that were more frequently used in the proposed model were represented by the red solid line, whereas roads that were more frequently used in the demand model were represented by the green dash line. The black solid line represented the road where one direction was more frequently used in the proposed model and the other direction

was more frequently used in the demand model. Vehicles in the proposed model tended to detour whereas vehicles in the demand model tended to use the shortest route.

In the demand model, vehicles on the uncongested link had to give way to vehicles on the congested link. Therefore, even though the running time of the congested link was more than the uncongested road, the waiting time at the intersection of the congested link was less than the uncongested road. When the extra waiting time and the extra detour time were more than the time lost in the congestion, most vehicles in the demand model prefer the shortest road which usually was more congested. On the contrary, in the proposed model, since links were treated more fairly, the influence of the extra waiting time at the intersection was not as significant as the time lost in the congestion. If the extra detour time was less than the time lost in the congestion, vehicles were willing to detour to avoid the congested links. Due to the full use of the network, the congestion level can be reduced in the proposed model. The travel cost might not decrease significantly in the proposed model as shown in Table 5.4 because the extra detour time depends on the topology of the network. To further analyze the influence of the proposed model on the route choice, another simulation which aimed to reproduce the tidal traffic that usually appears during peak hours in the real-world urban network was conducted. Similar results can be found and details are attached in Appendix C. In a word, it can be concluded that the proposed model can reduce the congestion of the network with similar travel cost, compared with other LTM-based models.

## 5.4 Summary

This chapter combined the classic LTM and a local linear programming formulation which optimized the throughput of each incoming link at each intersection to reduce the congestion on networks. The characteristic of the proposed model is that it reduces the congestion when the route choice is determined, so it can be added to other existing traffic problem solutions, such as the routing approach which aims at minimizing the total emission. The proposed model might have other practical applications, for example, the dynamic determination of priority fraction can be converted into the real-time signal control; the proposed model can be applied to the negotiation among vehicles when the car connection and driverless car are realized in the future.

The simulation results showed that compared with other LTM-based models, vehicles in the proposed model were more willing to detour to avoid the congestion if the extra detour time was less than the time lost in the congested link because incoming links at the intersection were assigned with relatively fair priority. It can be concluded that the proposed model can reduce the congestion at the network because of the full use of the network while maintaining a similar or even less travel cost.

## **CHAPTER 6 Conclusions and Future Work**

### **6.1 Conclusions**

The development of data collection technology, especially the probe data, made a large abundant of real-time urban traffic data available so that the ITS solution for urban congestion has become possible. However, there are still some problems restricting the wide application of ITS-based traffic control and management in the real-world. This dissertation focused on the following three issues to contribute to ITS-based traffic control and management strategy which dynamically controls the priority at the intersection and employs short-term urban link travel time prediction so that it can catch up with the frequent change in the urban traffic condition.

Firstly, the low penetration rate of probe vehicles often constrains the wide application of the ITS solution at the urban network. For example, when making travel time prediction which is very important for route searching, the travel time of some links from the route might be unpredictable due to the low penetration rate. To overcome the low penetration rate problem, a non-parametric model based on Bayes' theorem and a resampling process was proposed to predict short-term urban link travel time in chapter 3. Most existing models, such as the KNN-based model, predicted the travel time of the target link with data from that link. To improve accuracy, they used to add the correlation between links to the prediction results

by applying data from the adjacent links. Differently, the proposed model predicted the travel time of the target link by data from its crossing direction when there was no observation on the target link. Compared with the KNN-based and PF-based models, the proposed model had similar prediction accuracy but a higher coverage rate. Results showed that the data from the crossing direction can make a prediction as accurate as using the data from the target link. Therefore, using the data from the crossing direction can significantly enhance the coverage rate and maintain the prediction accuracy in the low penetration rate environment.

Secondly, travelers sometimes need traffic information in the longer future, but the current ITS system usually provides this kind of information in an aggregation form because the data used to provide the information were aggregated. The aggregation information only can provide a general description of the traffic condition so it is difficult to meet different individual needs. For example, some commuters prefer routes with higher travel time reliability (less variety) rather than routes with shorter average travel time. To provide more useful information, a non-parametric model using disaggregate probe data based on the DTW was developed in chapter 4. Instead of a fixed prediction interval, the proposed model can make travel time prediction based on the traffic signal cycle so that it can reflect changes in travel time during a signal cycle. The proposed model was compared with the classic KNN model and the naïve model by both simulation data and real-world data. Results showed that the proposed model outperformed the other two models even under extremely low penetration rates and that it can be applied more widely because it was free from the constraint of the equal time interval and can reflect changes in travel time.

Thirdly, when the study site changes from the highway to the urban network, the traffic condition becomes more complex. It is more doable to solve the congestion problem over a small scale where the traffic condition has a similar characteristic instead of reducing the congestion on the whole network. Intuitively, the intersection which is the main potential bottleneck on the urban network attracts our interest. To reduce congestion at the intersection, a local linear programming formulation was combined with the LTM to dynamically determining the priority of each incoming link in chapter 5. The priority strategy was decided by the traffic condition of the link connecting to the intersection, which means the decision of the priority at each intersection is independent. The proposed priority strategy has the potential to be applied to the urban network in the real world because all intersections on the urban network can be processed simultaneously by this strategy. Results showed that compared with other priority strategies, the proposed priority strategy can reduce congestion at the network because it encouraged individuals to make full use of road resources. In other words, the individual's route choice was affected by the proposed priority strategy. Even though the individual might choose to make a detour, the extra detour time is no more than the time lost in the congestion of the "shortest" route. Otherwise, the individual would still choose the "shortest" route.

## 6.2 Future Work

There is still much room for improving the dissertation. As for the short-term urban link travel time prediction which was explained in chapters 3 and 4, the focus will be concentrated on the following tasks in the future:

- i) Consider different turning behaviors.
- ii) Calibrate parameters of the proposed model in chapter 3 for higher accuracy.
- iii) Analyze the relationship between the coverage rate and variables, such as the length of the time interval and the traffic volume.
- iv) Reduce computation cost and further extend the prediction horizon.
- v) Consider the actuated traffic signal.
- vi) Test the proposed models in the real-world network and realize applications, such as travel time reliability analysis and reliable route searching.

As for the intersection priority control which was explained in chapter 5, the focus will be concentrated on the following tasks in the future:

- i) Apply the route searching at each intersection in the simulation.
- ii) Combine the proposed priority control strategy with other SODTA approaches to realize congestion reduction and other system optimal goals, such as emission reduction, at the same time



Last but not least, the combination of the short-term travel time prediction and the intersection priority control should be tested. However, due to the lack of data, this will also remain as future work. In addition, traffic demand prediction is needed to describe the traffic condition in the future. Using the prediction of travel time and traffic demand as the input, the proposed ITS-based traffic control and management strategy can simulate the traffic condition in the future so that the corresponding intersection priority control can not only reduce the traffic congestion that already exists at the network but also prevent the traffic congestion that will happen in a short future.

## APPENDIX A Algorithm for Signal Timing Estimation

To build a relatively sufficient historical database, it is assumed that the signal pattern remains the same during the same period of the day. In this algorithm, one signal cycle is divided into two phases—the green phase when the object vehicle can move straight through the intersection, and the red phase when the crossing vehicle can move through the intersection. There are two problems in estimation. One is the errors; for example, there might be an object vehicle data appearing in a stream of crossing vehicle data, which is physically impossible. The other one is an overestimation of the signal phase due to the lack of data.

**Table A.1 Algorithm A-I: Signal timing estimation**

---

### Part 1

- 1-1 Measure the period over which object vehicle data appear continuously as the length of the green phase. Intervals between each green phase are treated as red phases.
- 1-2 Change the red phases shorter than 40 s into green phases and remove the first and last phases.

### Part 2

- 2-1 Measure the period over which crossing vehicle data appear continuously as the length of the red phase. Intervals between each red phase are treated as green phases.
- 2-2 Change the green phase shorter than 40 s into red phases and remove the first and last phases.

### Part 3

- 3-1 Replace the red phases longer than 80 s in Part 2 with the corresponding segment in Part 1.
- 3-2 Calculate the average length of the green phase ( $AVE_{green}$ ) and its standard deviation ( $SD_{green}$ ) in part 2 with green phases between 40 s and 80 s long.

$$\text{Length of green phase} = AVE_{green} + \varepsilon_{green} \quad (\text{A-1})$$

---

---


$$\text{where } \varepsilon_{\text{green}} = \begin{cases} SD_{\text{green}}, & \delta \in (SD_{\text{green}}, +\infty) \\ \delta, & \delta \in [-SD_{\text{green}}, SD_{\text{green}}] \\ -SD_{\text{green}}, & \delta \in (-\infty, -SD_{\text{green}}) \end{cases} \quad \text{and } \delta \sim (0,1) \quad (\text{A-2})$$

3-3 Calculate the average length of the red phase ( $AVE_{\text{red}}$ ) and its standard deviation ( $SD_{\text{red}}$ ) in part 2 with red phases between 40 s and 80 s long.

$$\text{Length of red phase} = AVE_{\text{red}} + \varepsilon_{\text{red}} \quad (\text{A-3})$$

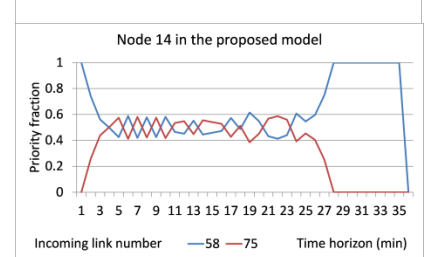
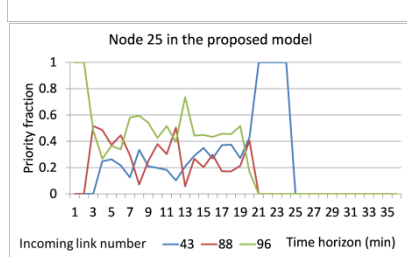
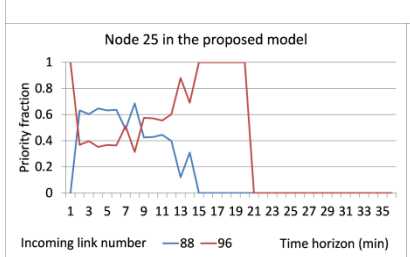
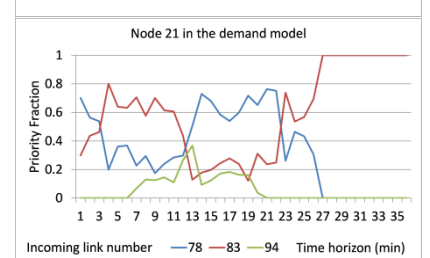
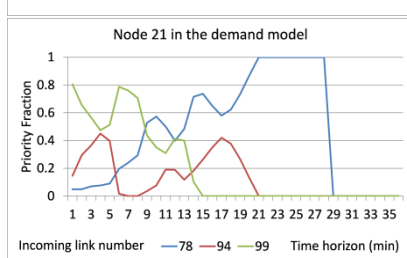
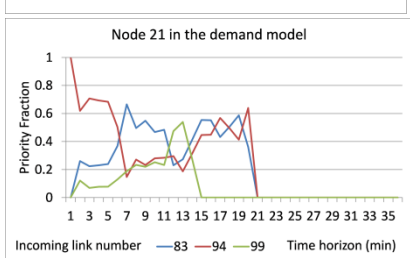
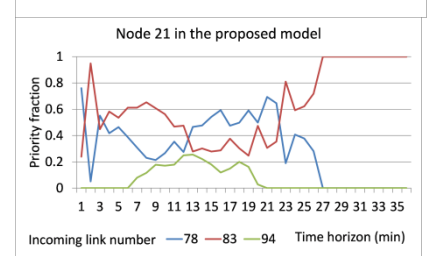
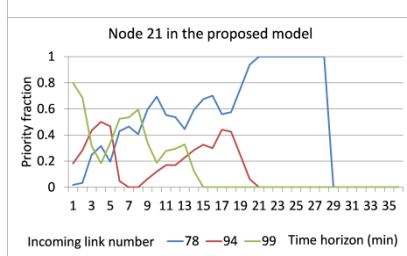
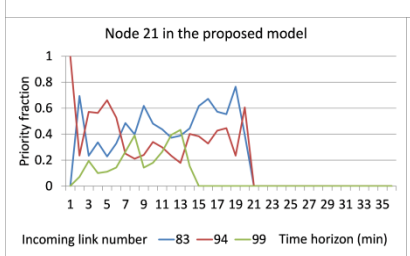
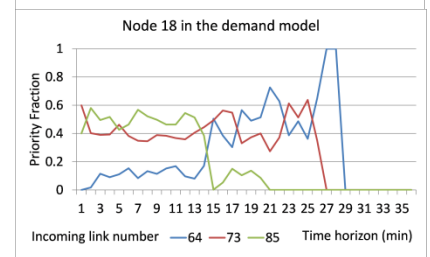
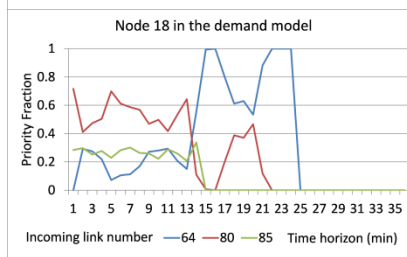
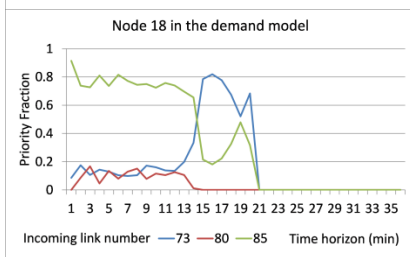
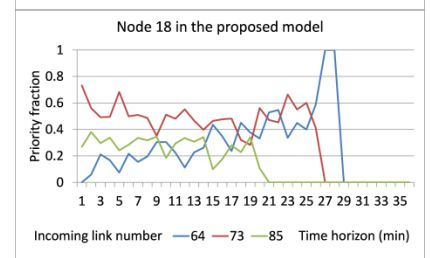
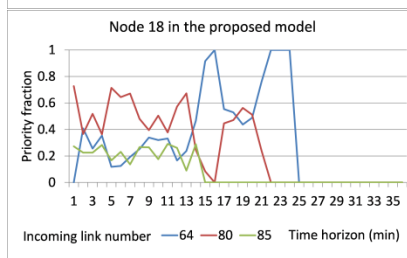
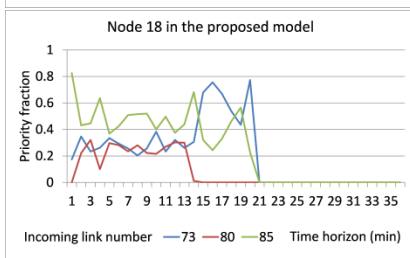
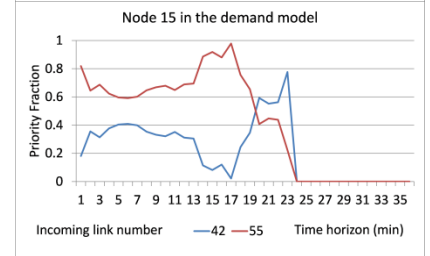
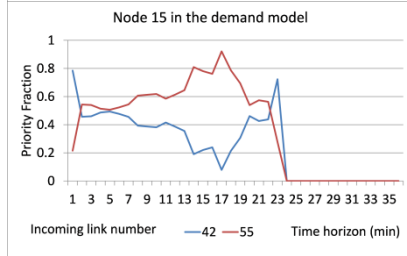
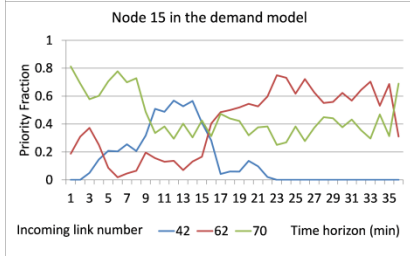
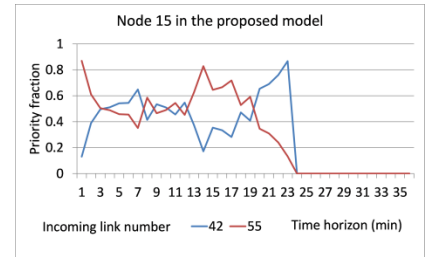
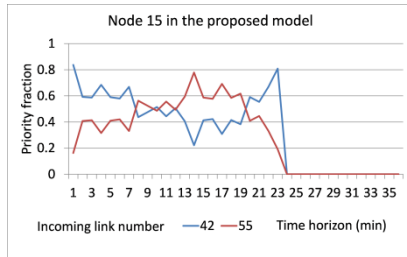
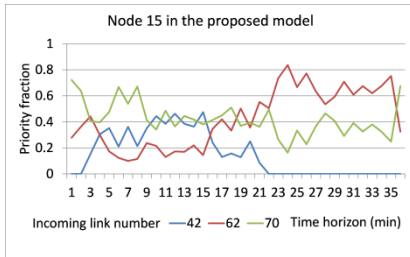
$$\text{where } \varepsilon_{\text{red}} = \begin{cases} SD_{\text{red}}, & \delta \in (SD_{\text{red}}, +\infty) \\ \delta, & \delta \in [-SD_{\text{red}}, SD_{\text{red}}] \\ -SD_{\text{red}}, & \delta \in (-\infty, -SD_{\text{red}}) \end{cases} \quad \text{and } \delta \sim (0,1) \quad (\text{A-4})$$

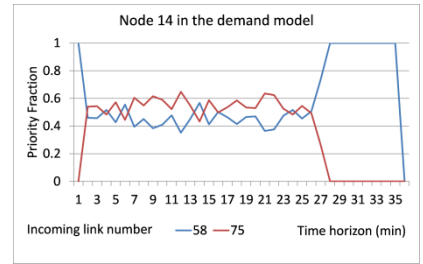
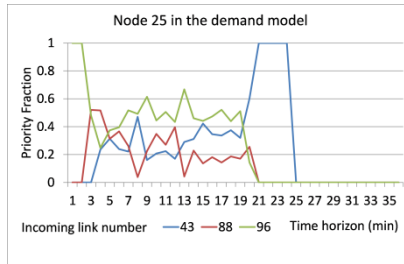
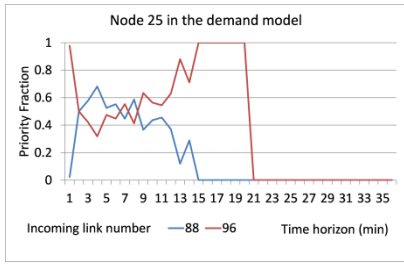

---

Algorithm A-I for signal timing estimation is shown in Table A.1. The historical data of object vehicles and crossing vehicles at the intersection from the same day are sorted as a time sequence. If there are object vehicles going through the intersection, the signal is green; otherwise, the signal is red, and vice versa (1-1 and 2-1). To address the first problem, the phase is changed when its length is under 40 s (1-2 and 2-2). Because traffic signals change successively, the phase at the beginning and end of the time period is removed. To address the second problem, the red phases in Part 2 over 80 s long are replaced by corresponding segments in Part 1 (3-1), and vice versa. Signal phases whose length is over 80 s are also removed from the calculation (Part 3). Here, 40 s and 80 s are used as the thresholds because in Japan, it is common for the average length of a signal phase to be 60 s with a variance of 20 s. Error term  $\delta$  is added to reflect unexpected events, such as a change in signal pattern. It is assumed that  $\delta$  follows a standard Gaussian distribution. Provided the historical data are abundant and precise, Part 1 or Part 2 can be used alone without overestimating the length of the phase.

# APPENDIX B Changes in the Priority in the Proposed Model and the Demand Model







**Figure B.1 Changes in the priority in the proposed model and the demand model**

## **APPENDIX C Simulation of tidal traffic on the Sioux Falls network**

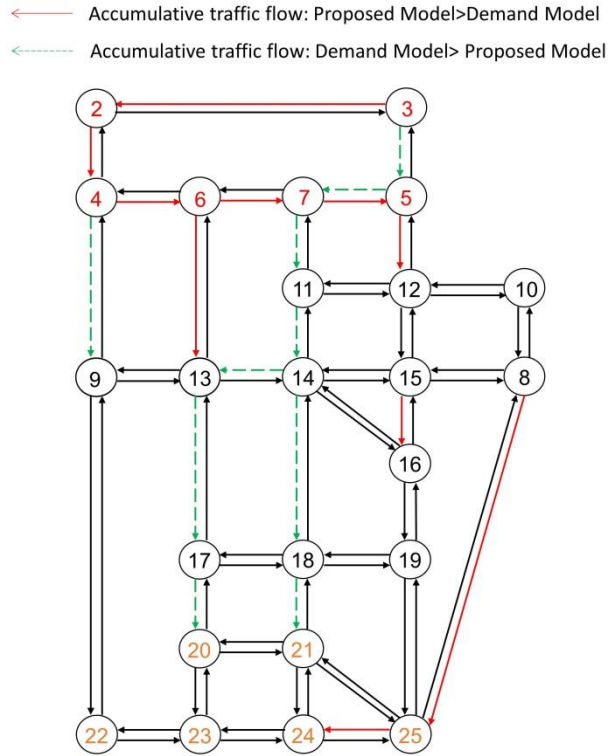
To simulate the tidal traffic, 6 nodes (Node 2-7) located on the top part of the Sioux Falls network were selected as the origin, while 6 nodes (Node 20-25) located on the bottom part of the Sioux Falls network were selected as the destination. OD pairs were randomly selected from these nodes. The assignment period was extended to 300 seconds. Other simulation settings were as same as in section 5.3.2. The ACI and ATT for each model and the results of the t-test between the proposed model and other LTM-based models are shown in Table C.1.

**Table C.1 Results of the simulation of tidal traffic on the Sioux Falls network.**

	Fairness Model	Capacity Model	Demand Model	Proposed Model
ACI	2.10	2.05	1.84	1.62
ATT (s)	706.74	682.81	645.16	651.18
t value for the ACI	-15.99	-14.31	-7.60	-
P value for the ACI	0.00	0.00	0.00	-
t value for the ATT	-12.85	-7.10	1.64	-
P value for the ATT	0.00	0.00	0.11	-

$\alpha=0.1$ , sample size=30

It is obvious that the proposed model outperformed the fairness model and the capacity model because the proposed model employed dynamic priority. The ATT of the proposed model was at the same level as that of the demand model, but the proposed model had a distinctly lower congestion level than the demand model. To further compare the proposed model and the demand model which both employed the dynamic priority, the first 10 links sorted by the difference of cumulative traffic flow between the proposed model and the demand model are shown in Figure C.1.



**Figure C.1 First 10 links sorted by the difference of cumulative traffic flow between the proposed model and the demand model**

The difference of cumulative traffic flow between the proposed model and the demand model shows the influence of the priority fraction on the route choice. In Figure C.1, the red links were used more frequently by vehicles in the proposed model, whereas the green links were used more frequently by vehicles in the demand model. It is intuitive that vehicles in the demand model tended to use the shortest path, whereas vehicles in the proposed model tended to detour. It is reasonable that vehicles in the proposed model used less congested roads to compensate for the time lost in detouring. This explains why the congestion level in the proposed model was distinctly lower than the demand model and they experienced similar average travel times at the same time.



## BIBLIOGRAPHY

Aach J., Church G. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17, 495-508.

Alrukaibi F., Alsaleh R., Sayed T. (2018). Real-time travel time estimation in partial network coverage: case study from Kuwait City. *Adv. Transp. Stud.*, 79-94.

Argote-Cabañero J., Christofa E., Skabardonis A. (2015). Connected vehicle penetration rate for estimation of arterial measures of effectiveness. *Transp. Res. Part C*, 60, 298-312.

Axer S., Friedrich B. (2017). Signal timing estimation based on low frequency floating car data. *Transp. Res. Procedia*, 25, 1648–1664.

Barth, M., Boriboonsomsin, K. (2008). Real-World carbon dioxide impacts of traffic congestion. *Transp. Res. Rec.*, 2058(1), 163–171.

Barte, P. (2000). Urban transport in Asia: Problems and prospects for high-density cities. *Asia-Pac. Dev. Monit.*, 2, 33–66.

Bauer, D., Tulic, M, Scherrer, W. (2019). Modelling travel time uncertainty in urban networks based on floating taxi data. *Eur. Transp. Res. Rev.*, 11, 46.

Bellavista P., Caselli F., Corradi A., Foschini L. (2018). Cooperative Vehicular Traffic Monitoring in Realistic Low Penetration Scenarios: The COLOMBO Experience. *Sensors*, 18(3), 822.

Ben-Akiva M., Bierlaire M., Burton D., Koutsopoulos N., Mishalani R. (2001). Network state estimation and prediction for real-time traffic management. *Networks and Spatial Economics*, 1(3-4), 293-318.

Berndt D., Clifford J. (1994). Using dynamic time warping to find patterns in time series. *AAAI-94 Workshop on Knowledge Discovery in Database*, 229-248.

Bliemer M.C.J., Raadsen M.P.H. (2019). Continuous-time general link transmission model with simplified fanning, part I: theory and link model formulation. *Transp. Res. Part B*, 126, 442-470.

Bloomberg L., Dale J. (2000). A comparison of the VISSIM and CORSIM traffic simulation models. *Inst. of Transp. Engineers Annual Meeting*, Nashville, TN, USA., August 6–9.

Bruechner J.K. (2011). *Lectures on Urban Economics*, The MIT Press, Cambridge.

- Bucknell C., Herrera J. C. (2014). A trade-off analysis between penetration rate and sampling frequency of mobile sensors in traffic state estimation. *Transp. Res. Part C*, 46, 132–150.
- Cai P., Wang Y., Lu G., Chen P., Ding C., Sun J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transp. Res. Part C*, 62, 21–34.
- Cao P., Miwa T., Morikawa T. (2014). Modeling distribution of travel time in signalized road section using truncated distribution. *Proc.-Social Behav. Sci.*, 138, 137-147.
- Carrion C., Levinson D. (2012). Value of travel time reliability: a review of current evidence. *Transp. Res. Part A*, 46, 720-741.
- Chakraborty S., Rey D., Moylan E., Waller S.T. (2018). Link transmission model based linear programming formulation for network design. *97th Annual Meeting of the Transp. Res. Board*, Washington, DC, U.S..
- Chen B. Y., Shi C., Zhang J., Lam W. H. K., Li Q., Xiang S. (2016). Most reliable path-finding algorithm for maximizing on-time arrival probability. *Transp. B: Transp. Dyn.*, 5(3), 248-264.
- Chen H., Rakha H. A. (2014). Real-time travel time prediction using particle filtering with a non-explicit state-transition model. *Transp. Res. Part C*, 43(1), 112–126.
- Cheu R. L., Xie C., Lee D. (2002). Probe Vehicles population and Sample Size for arterial speed Estimation. *Comput.-Aided Civil Infrastruct. Eng.*, 17(1), 53-60.
- Daganzo C.F. (1994). The cell transmission model: a dynamic representation of highway traffic consistent with hydrodynamic theory. *Transp. Res. Part B*, 28, 269-287.
- Daganzo C.F. (1995). The cell transmission model, Part II: network traffic. *Transp. Res. Part B*, 29, 79-94.
- Dhivyabharathi B., Hima E. S., Vanajakshi L. (2016). Stream travel time prediction using particle filtering approach. *Transp. Lett.*, 1–8.
- Elhenawy M., Chen H., Rakha H. A. (2014). Dynamic travel time prediction using data clustering and genetic programming. *Transp. Res. Part C*, 42, 82-98.
- Fayazi S. A., Vahidi A., Mahler G., Winckler A. (2015). Traffic signal phase and timing estimation from low-frequency transit bus data. *IEEE Trans. Intell. Transp. Syst.*, 16(1), 19–28.

Fei X., Lu C., Liu K. (2011). A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. *Transp. Res. Part C*, 19(6), 1306-1318.

Fellendorf M., Vortisch P. (2001). Validation of the microscopic traffic flow model VISSIM in different real-world situations. *80th Annual Meeting of the Transp. Res. Board*, Washington, DC, U.S..

Feng Y., Hourdos J., Davis G. A. (2014). Probe vehicle based real-time traffic monitoring on urban roadways. *Transp. Res. Part C*, 40, 160–178.

Flötteröd G., Osorio C. (2017). Stochastic network link transmission model. *Transp. Res. Part B*, 102, 180-209.

Fusco G., Colombaroni C., Isaenko N. (2016). Short-term speed predictions exploiting big data on large urban road networks. *Transp. Res. Part C*, 73, 183–201.

Fusco G., Gori S. (1995). The use of artificial neural networks in advanced traveler information and traffic management systems. *Appl. Adv. Technol. Transport. Eng.: ASCE*, 341-345.

García-Nieto J., Alba E., Olivera A.C. (2012). Swarm intelligence for traffic light scheduling: Application to real urban areas. *Eng. Appl. Artif. Intell.*, 25 (2), 274–283.

Gentile G. (2010). The general link transmission model for dynamic network loading and a comparison with the DUE algorithm. *New Developments in Transport Planning: Advances in Dynamic Traffic Assignment*, 153-178.

Gentile G. (2015). Using the general link transmission model in a dynamic traffic assignment to simulate congestion on urban networks. *Transportation Research Procedia*, 5, 66-81.

Gentile G. (2017). Formulation of the transit link transmission model. *Transportation Research Procedia*, 27, 889-896.

Guo J., Huang W., Williams B. M. (2014). Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. Part C*, 43(1), 50–64.

Habtemichael F. G., Cetin M. (2016). Short-term traffic flow rate forecasting based on identifying similar traffic patterns. *Transp. Res. Part C*, 66, 61–78.

Hajiahmadi H., Corthout R., Tampère C., De Schutter B., Hellendoorn H. (2013). Variable

speed limit control based on extended link transmission model. *Transp. Res. Rec.*, 2390, 11-19.

Himpe W., Corthout R., Tampère C.M.J. (2016). An efficient iterative link transmission model. *Transp. Res. Part B*, 92(B), 170-190.

Hofleitner A., Herring R., Bayen A. (2012). Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning. *Transp. Res. Part B, Methodol.*, 46(9), 1097-1122.

Hook W., Replogle M. (1996). Motorization and non-motorized transport in Asia: Transport system evolution in China, Japan and Indonesia. *Land Use Policy*, 13(1), 69-84.

Jayasooriya S. A. C. S., Bandara Y. M. M. S. (2017). Measuring the economic costs of traffic congestion. *2017 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, 141-146.

Jenelius E., Koutsopoulos H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. *Transp. Res. Part B*, 53, 64–81.

Jenelius E., Koutsopoulos H. N. (2018). Urban network travel time prediction based on a probabilistic principal component analysis model of probe data. *IEEE Trans. Intell. Transp. Syst.*, 19(2), 436-445.

Jiang G., Zhang R. (2001). Travel-time prediction for urban arterial road: a case on China. *IVEC2001. Proceedings of the IEEE International Vehicle Electronics Conference 2001. IVEC 2001 (Cat. No.01EX522)*, Tottori, Japan, 255-260.

Jin W.L., Zhang H.M. (2003). On the distribution schemes for determining flows through a merge. *Transp. Res. Part B*, 37(6), 521-540.

Kerper M., Wewetzer C., Sasse A., Mauve M. (2012). Learning traffic light phase schedules from velocity profiles in the cloud. *Proc. 5th Int. Conf. NTMS*, 1–5.

Lebacque J.P. (1996). The Godunov scheme and what it means for first order traffic flow models. *13th International Symposium on Transportation and Traffic Theory*.

Levin M.W. (2017). Congestion-aware system optimal route choice for shared autonomous vehicles. *Transp. Res. Part C*, 82, 229-247.

Li Z., Hensher D. A., Rose J. M. (2010). Willingness to pay for travel time reliability in passenger transport: A review and some new empirical evidence. *Transp. Res. Part E*, 46,

384-403.

Li Z., Kluger R., Hu X., Wu Y., Zhu X. (2018). Reconstructing vehicle trajectories to support travel time estimation. *Transport. Res. Rec.: J. Transport. Res. Board*, 2672(42), 148-158.

Lint van J. W. C. (2008). Online learning solutions for freeway travel time prediction. *IEEE Trans. Intell. Transp. Syst.*, 9(1), 38-47.

Lint van J. W. C., Hoogendoorn S. P., Zuynen van H. J. (2005). Accurate freeway travel time prediction with state-space neural networks under missing data. *Transp. Res. Part C*, 13(5-6), 347-369.

Long J., Chen J., Szeto W.Y., Shi Q. (2018). Link-based system optimum dynamic traffic assignment problems with environmental objectives. *Transp. Res. Part D*, 60, 56-75.

Lu L., Li X., Zheng P., Wang K. (2018). Determining the required probe vehicle size for real-time travel time estimation on signalized arterial. *IEEE Access*, 7, 4546-4554.

Mori U., Mendiburu A., Álvarez M., Lozano J. A. (2015). A review of travel time estimation and forecasting for advanced traveller information systems. *Transp. A: Transp. Sci.*, 11 (2), 119-157.

Muller M. (2007). *Information Retrieval for Music and Motion*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., pp. 69-84.

Newell G.F. (1993). A simplified theory of kinematic waves in highway traffic, part I: general theory; part II: queuing at freeway bottlenecks; part III: multi-destination flows. *Transp. Res. Part B*, 27, 281-313.

Nezamuddin N., Boyles S.D. (2014). A continuous DUE algorithm using the link transmission model. *Networks and Spatial Economics*, 15 (3), 465-483.

Osorio C., Flötteröd G. (2014). Capturing dependency among link boundaries in a stochastic dynamic network loading model. *Transportation Science*, 49 (2), 420-431.

Park B., Messer C., Urbanik T. (2000). Enhanced genetic algorithm for signal-timing optimization of overstaturated intersections. *Proc. Transp. Res. Board 79th Annu. Meeting*, 1661-1680.

Peeta S., Ziliaskopoulos A.K. (2001). Foundations of dynamic traffic assignment: the past, the present, and the future. *Netw Spat Econ I*, 233-265.

- Petersen N. C., Rodrigues F., Pereira F. C. (2019). Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Syst. Appl.*, 120, 426-435.
- Polson N. G., Sokolov V. O. (2017). Deep learning for short-term traffic flow prediction. *Transp. Res. Part C*, 79, 1-17.
- Richards M.G. (2006). *Congestion charging in London: the policy and the politics*. Palgrave Macmillan UK, London.
- Robinson S., Polak J. (2005). Modeling urban link travel time with inductive loop detector data by using the k-NN method. *Transp. Res. Rec. J. Transp. Res. Board*, 47-56.
- Sanaullah I., Quddus M., Enoch M. (2016). Developing travel time estimation methods using sparse GPS data. *J. Intell. Transp. Syst.*, 20(6), 532–544.
- Shi C., Chen B. Y., Li Q. (2017). Estimation of travel time distributions in urban road networks using low-frequency floating Car data. *ISPRS Int. J. Geo-Inf.*, 6, 253.
- Smith B. L., Demetsky M. J. (1994). Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2, 1706-1709.
- Smith B. L., Williams B. M., Oswald R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transp. Res. Part C*, 10(4), 303–321.
- Srinivasan K. K., Jovanis P. P. (1996). Determination of number of probe vehicles required for reliable travel time measurement in urban networks. *Transport. Res. Rec.: J. Transport. Res. Board*, 1537(1), 15–22.
- Sumalee A., Zhong R., Pan T.L., Szeto W.Y. (2011). Stochastic cell transmission model (SCTM): a stochastic dynamic traffic model for traffic state surveillance and assignment. *Transp. Res. Part B*, 45(3), 507-533.
- Texas A&M Transp. Inst., INRIX Inc. (2015). *2015 Urban mobility scorecard*. TX, USA.
- Tri-State Transportation Campaign (2018). *A way forward for New York city: road pricing in London, Stockholm, and Singapore*. New York, USA.
- Sun D.J., Liu X., Ni A., Peng C. (2014). Traffic congestion evaluation method for urban arterials: case study of Changzhou, China. *Transp. Res. Rec.: J. Transp. Res. Board*, 2461, 9-15.

U.S. Department of Transportation (2017). 2017 National Household Travel Survey.

Van der Gun J.P.T., Pel A.J., Van Arem B. (2017). Extending the link transmission model with non-triangular fundamental diagrams and capacity drops. *Transp. Res. Part B*, 98, 154-178.

Van de Weg G.S., Keyvan-Ekbatani M., Hegyi A., Hoogendoorn S.P. (2016). Urban network throughput optimization via model predictive control using the Link Transmission Model. *95th Annual Meeting of the Transp. Res. Board*, Washington, DC, U.S..

Van Hinsbergen C. P. I. J., Lint van J. W. C., Sanders F. M. (2007). Short term traffic prediction models. *Proceedings of the 14<sup>th</sup> ITS World Congress*.

Vlahogianni E. I., Karlaftis M. G., Golias J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transp. Res. Part C*, 43(1), 3–19.

Wan N., Vahidi A., Luckow A. (2016). Reconstructing maximum likelihood trajectory of probe vehicles between sparse updates. *Transp. Res. Part C*, 65, 16-30.

Wang J., Shi Q. (2013). Short-term traffic speed forecasting hybrid model based on Chaos-Wavelet analysis-support vector machine theory. *Transp. Res. Part C*, 27, 219-232.

Wang J., Tsapakis I., Zhong C. (2016). A space-time delay neural network model for travel time prediction. *Eng. Appl. Artif. Intell.*, 52, 145-160.

Wang Y., Szeto W.Y., Han K., Friesz T.L. (2018). Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications. *Transp. Res. Part B*, 111, 370-394.

Wiering M. (2000). Multi-agent reinforcement learning for traffic light control. *Proc. 17th Int. Conf. Mach. Learn. (ICML)*, 1151-1158.

Xia J., Chen M., Huang W. (2011). A multistep corridor travel-time prediction method using presence-type vehicle detector data. *J. Intell. Transp. Syst.*, 15(2), 104–113.

Xia J., Huang W., Guo J. (2012). A clustering approach to online freeway traffic state identification using ITS data. *KSCE J. Civ. Eng.*, 16, 426-432.

Xu W., Zhou H., Cheng N., Lyu F., Shi W., Chen J., Shen X. (2018). Internet of vehicles in big data era. *IEEE/CAA J. of Autom. Sinica*, 5(1), 19-35.

Yan J., Yun X., Luo H., Wu Z., Zhang S. (2013). Out-of- sequence traffic classification based on improved dynamic time warping. *IEICE Trans. Inform. Syst.*, 96(11), 2354-2364.

Yperman I. (2007). The Link Transmission Model for dynamic network loading. *PhD Thesis*, Katholieke Universiteit Leuven, Belgium.

Yu J., Lu P. (2016). Learning traffic signal phase and timing information from low-sampling rate taxi GPS trajectories. *Knowl.-Based Syst.*, 110, 275–292.

Zhang Z., Zhao T., Ao X., Yuan H. (2017). A vehicle speed estimation algorithm based on dynamic time warping approach. *IEEE Sensors Journal*, 17(8), 2456-2463.

Zheng F., Zuylen H. V. (2013). Urban link travel time estimation based on sparse probe vehicle data. *Transp. Res. Part C*, 31, 145–157.