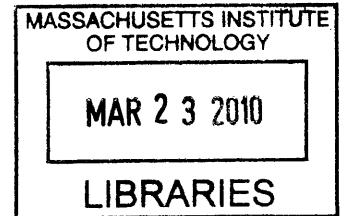


**Critical Enhancements of a Dynamic Traffic
Assignment Model for Highly Congested, Complex
Urban Network**

by

Zheng Wei

Bachelor of Science in Automation (2008)
Tsinghua University, Beijing, China



Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Civil and Environmental Engineering

at the

ARCHIVES

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010

© Massachusetts Institute of Technology 2010. All rights reserved.

Author
Department of Civil and Environmental Engineering
January 15, 2010

Certified by
Moshe E. Ben-Akiva
Edmund K. Turner Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by
Daniele Veneziano
Chairman, Departmental Committee for Graduate Students

Critical Enhancements of a Dynamic Traffic Assignment Model for Highly Congested, Complex Urban Network

by

Zheng Wei

Submitted to the Department of Civil and Environmental Engineering
on January 15, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Civil and Environmental Engineering

Abstract

To accurately replicate the highly congested traffic situation of a complex urban network, significant challenges are posed to current simulation-based dynamic traffic assignment (DTA) models. This thesis discusses these challenges and corresponding solutions with consideration of model accuracy and computational efficiency. DynaMIT-P, an off-line mesoscopic DTA model is enhanced. Model success is achieved by several critical enhancements aimed to better capture the traffic characteristics in urban networks. A Path-Size Logit route choice model is implemented to address the overlapping routes problem. The explicit representation of lane-groups accounts for traffic delays and queues at intersections. A modified treatment of acceptance capacity is required to deal with the large number of short links in the urban network. The network coding is revised to maintain enough loader access capacity in order to avoid artificial bottlenecks. In addition, the impacts of bicycles and pedestrians on automobile traffic is modeled by calibrating dynamic road segment capacities.

The enhanced model is calibrated and applied to a case study network extracted from the city of Beijing, China. Data used in the calibration include sensor counts and floating car travel time. The improvements of the model performance are indicated by promising results from validation tests.

Thesis Supervisor: Moshe E. Ben-Akiva

Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Acknowledgments

First of all, I would like to express my heartfelt thanks to my research supervisor, Prof. Moshe Ben-Akiva for his constant encouragement and invaluable guidance during my graduate study at MIT. It has been a privilege to work with him. I am deeply grateful to Prof. Song Gao, University of Massachusetts and Dr. Yang Wen, Oracle Corp. for guiding me and being constant source of knowledge and inspiration. They have spent countless hours going through every detail of my work, giving me feedback on my research.

I would also like to thank my academic advisor Prof. Cynthia Barnhart, MST program director Prof. Nigel H.M. Wilson, Kris kipp, Patty Glidden, and Tina Xue for all their help.

Thanks to Beijing Transportation Reseach Center (BTRC) and Liyun Zhu for their data and financial support. I am also thankful to the people who have worked on this study, my labmate Lang Yang, Fan Wei, and visiting students Hongliang Ma, Li Qu from Tsinghua University, Nale Zhao from Beijing Jiaotong University. A special thanks goes to Dr. Yuelong Su from Tsinghua University for his help and interest in my research.

During my studies in the U.S., I have acquainted myself with a lot of new friends in and out of MIT. I would like to express my thanks to them for making my life wonderful and memorable. They are: Tina Xue, Eric Huang, Maya Abou-Zeid, Cristian Angelo Guevara, Sujith Rapolu, Shunan Xu, Ying Zhu, Vikrant Vaze, Xin Zhang, Xiaoning Wang, Yinyin Zhu. . .

Finally, my deepest gratitude goes to my family, my parents and my wife Guannan Wang, for their love, understanding, constant support and encouragment. This work is dedicated to them.

Contents

Abstract	3
Acknowledgements	5
Contents	6
List of Figures	10
List of Tables	13
1 Introduction	15
1.1 Motivation and Problem Statement	15
1.2 Thesis Contribution	17
1.3 Thesis Outline	18
2 Literature Review	19
2.1 General Background of DTA Approach	19
2.2 Simulation-Based DTA	21
2.2.1 Macroscopic Models	21
2.2.2 Microscopic Models	22
2.2.3 Mesoscopic Models	23
2.3 Congested Large-Scale Network Simulation	24
2.4 Calibration Algorithm	27
2.5 DTA Model Enhancement	28

3	Modeling Challenges and Solutions	31
3.1	Revised Route Choice Model	32
3.1.1	Logit route choice model	32
3.1.2	C-Logit model	33
3.1.3	Path Size Logit Model (PSL)	33
3.1.4	Result	36
3.2	Lane-Group-Based Queuing Feature	38
3.3	Treatments for Short Links	43
3.3.1	Increase the Minimum Speed	44
3.3.2	Dynamic Acceptance Capacity	46
3.3.3	Result	48
3.4	Centroid (Loader) Access Capacity	51
3.4.1	Adding Centroid Connector	52
3.4.2	Result	55
3.5	Variable Output Capacity	57
4	Off-line Model Calibration	59
4.1	Need for DTA Calibration	59
4.2	Calibration Methodology	61
4.3	Problem Formulation and Characteristics	61
4.4	Solution Algorithm	63
4.4.1	Algorithm Feature	63
4.4.2	Choice of the Gain Sequences	64
4.4.3	Algorithm Process Description	65
5	Case Study : City of Beijing	69
5.1	Introduction to DynaMIT-P	69
5.2	Network and Data	70
5.2.1	Surveillance Data	73
5.2.2	Sensor Data Analysis	76
5.3	Calibration	78

5.3.1	Calibration Variables	78
5.3.2	Methodology	78
5.3.3	Results	79
5.4	Application Analysis	84
6	Conclusion	89
6.1	Summary	89
6.2	Future Research Directions	91
A	MATLAB Code for SPSA Algorithm	93
B	Fit-To-Counts for Every Simulation Interval	105
	Bibliography	109

List of Figures

1-1	Congestion Growth Trend in the U.S. (Source: Schrank and Lomax, 2009)	15
2-1	Structure of Generic DTA (Source: Balakrishna, 2006)	20
3-1	The Overlapping Path Problem	34
3-2	Choice Probabilities for the Overlapping Path Network	35
3-3	Path set for OD pair: 1616 - 1584	37
3-4	Deterministic Queuing Model	38
3-5	Left-turn lane block the through and right turn traffic	40
3-6	Off-ramp block the through traffic on the freeway	41
3-7	Group of lanes according to turning movement	42
3-8	A complicated interchange with short links	44
3-9	A Typical Freeway Short Link #372	49
3-10	Density Change on Short Link #372	50
3-11	Speed Change on Short Link #372	50
3-12	Flow Change on Short Link #372	51
3-13	Real world network of the area of interest	53
3-14	Computer representation of the area of interest	53
3-15	Simulation result of the artificial bottleneck	54
3-16	Compare of the network with-and-without connector	55
3-17	Density change on link 419 (ring road freeway)	56
3-18	Density change on link 420 (off ramp)	56
3-19	Severe interferences from non-motorized	57

4-1	Off-line calibration framework (Source: Balakrishna R. , 2006)	60
4-2	SPSA Algorithm Workflow	67
5-1	Network V/C ratios of morning peak on a weekday, 2007	71
5-2	Study area (within the black rectangle)	72
5-3	Frequent on- and off-ramps	72
5-4	Deployment of RTMS detectors	74
5-5	Coverage of FCD system	74
5-6	Network of the study area	75
5-7	Locations of inconsistent RTMS detectors (1)	76
5-8	Locations of inconsistent RTMS detectors (2)	77
5-9	Trend of objective value for 530 iterations	81
5-10	Calibration results for the peak periods	82
5-11	Fit to counts analysis	83
5-12	Fit to link travel time analysis	83
5-13	Fit to path travel time analysis	85
5-14	Application analysis in TianNingSi Area, 7:00AM	86
5-15	Application analysis in XiZhiMen Area, 7:15AM	87
B-1	Fit-To-Counts for 6:30 to 7:00	105
B-2	Fit-To-Counts for 7:00 to 8:00	106
B-3	Fit-To-Counts for 8:00 to 9:00	107
B-4	Fit-To-Counts for 9:00 to 10:00	108

List of Tables

3.1	Path Choice Probabilities Before/After PSL Enhancement	36
3.2	Typical Lane Groups for Analysis(Source: HCM 2000)	43
5.1	Inconsistent Observed Counts from RTMS detectors	78
5.2	Starting values of model fit	80
5.3	Overall calibration results	81
5.4	Fit-to-counts by time interval	84
5.5	Analysis of Rotating No-Driving Day Restriction	87

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Demand for urban trips continues to grow as population increases, particularly in metropolitan areas. According to the *Traffic Congestion and Reliability Report* (FHWA, 2005), congestion has grown substantially throughout the U.S. over the past 20 years, regardless of city size. The 2009 Urban Mobility Report (Schrank and Lomax, 2009) estimates that, “ In 2007, congestion caused urban Americans to travel 4.2 billion hours more, resulting an extra 2.8 billion gallons of fuel for a congestion cost of \$87.2 billion, an increase of more than 50% over the previous decade ”

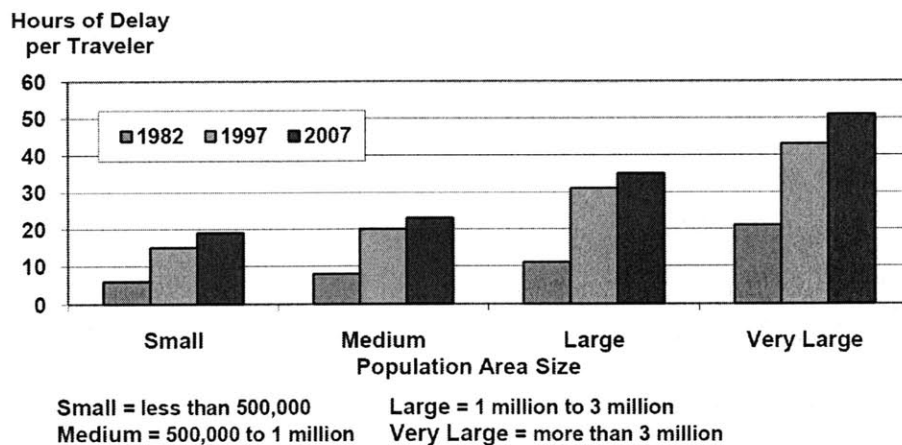


Figure 1-1: Congestion Growth Trend in the U.S. (Source: Schrank and Lomax, 2009)

Traffic congestion poses a number of environmental, social and economic challenges for major cities around the world. The construction of new roads to increase capacity has not kept pace with the continual growth in vehicle ownership and travel demand due to physical and economic constraints. Therefore, Advanced Traffic Management Systems (ATMS) and Intelligent Transportation Systems (ITS) are attracting increasing attention. The design, operation and management of these traffic systems in metropolitan areas call for traffic network model that can address large-scale congested urban traffic networks. A large-scale congested urban traffic network is generally marked by the following characteristics:

- a) Large number of directional links, usually in the thousands or more;
- b) Large number of relatively short links, at-grade intersections and separate-grade interchanges;
- c) Closely-spaced on- and off-ramps connecting elevated roadways and surface roads;
- d) Severe congestion with queues and spillbacks frequently occurring in the network;
- e) Complex mixed traffic interaction. Motorized vehicles receive significant interference from non-motorized traffic at intersections.

Each of these characteristics may pose extra challenges that only sophisticated models, retaining certain corresponding feature can handle.

While rough estimates of traffic impacts from a conventional static traffic assignment model might be adequate for the analysis of major infrastructure changes, models that can capture the dynamic nature of both the travel demand and traffic flow are required to evaluate demand management and traffic control strategies, such as high-occupancy-vehicle (HOV) or high-occupancy-toll (HOT) lanes and congestion pricing. One of the critical components of such models is known as Dynamic Traffic Assignment (DTA). Unfortunately, until recently, real-world DTA applications have

been limited in highway networks, which are relatively simple with relatively small number of links and low congestion level. There have been some attempts to deal with either congestion problems (Barcelo and Casas, 2003; Dittberner and Kerns, 2002; Ido and Prashker, 2009; Kaysi et al., 2003; Tong and Wong, 2000; Varia and Dhingra, 2004; Xu, 2009a) or large-scale networks (Caliper Corporation, 2009; Mahut, 2009; Wen, 2009). But few of them are tested and validated on networks that are both large-scale and heavily congested (Ben-Akiva et al., 2010).

This has motivated the attempt to better understand and to solve the problems that prevent current DTA models from replicating the urban traffic situation realistically within reasonable running time. Several key elements and enhancements that are critical to DTA applications in large-scale congested urban network were identified and implemented in this thesis.

A case study on a traffic network in Beijing, China was used to demonstrate the modelling process and to validate the enhancements identified in this thesis. The Beijing network is large and heavily congested. Without model enhancements, attempts to calibrate the model in order to replicate the real-world traffic situation failed because of the emergence of non-existing gridlock in the simulation. After implementing the discussed enhancements, the gridlocking problem was resolved and the enhanced model was successfully calibrated using the observed data.

1.2 Thesis Contribution

The main contribution of this thesis is in the identification of an array of important modeling features that are required for the application of DTA models in large-scale congested urban networks, which include:

- i) An enhanced route choice model which can account for overlapping routes to address the identified route choice bias on freeways.
- ii) Explicit representations of lane groups and lane based queuing models to properly model traffic queues and spillbacks;

- iii) The ability to handle a large number of short links without the artificial modification of their geometric representation;
- iv) Revised network coding to avoid artificial bottlenecks by maintaining enough centroid (loader) access capacity;
- v) The impacts of bicycles and pedestrians on auto traffic, modeled by calibrated dynamic road segment capacities.

Although, DynaMIT-P was used (as described in Section 5.1 and Appendix A of Balakrishna (2006)) and enhanced for the modeling of the case study, the features aforementioned are generally applicable to most current simulation-based DTA frameworks. The methodologies and conclusions in the discussion are by no means specific to DynaMIT-P, and can be generalized to a wider range of DTA models.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 presents a detailed review of simulation-based DTA and applications on real world networks. Then the problems encountered during the simulation of a highly congested large-scale urban network and the corresponding enhancement solutions are discussed in Chapter 3¹. The model calibration methodology is presented in Chapter 4. Chapter 5 is a case study in the city of Beijing, China, where DynaMIT-P, a state-of-the-art simulation-based DTA system with a mesoscopic traffic (supply) simulator, is calibrated successfully by applying the model enhancements discussed here. A traffic management case analysis is presented after using the calibrated model. Finally conclusions and direction for future research are discussed.

¹Chapter 3,4,5 are based on the research in paper: Ben-Akiva, M., S. Gao, Z. Wei and Y. Wen. "A Dynamic Traffic Assignment Model for a Highly Congested Network". In Preparation

Chapter 2

Literature Review

This chapter reviews current simulation-based DTA models and their applications, with focus on congested large-scale networks. The general background of DTA systems will also be briefly reviewed.

2.1 General Background of DTA Approach

Enabled advancements in technology and computational power, DTA is a high-fidelity model designed to replicate and predict complex traffic phenomena. Peeta and Ziliaskopoulos (2001) offers a comprehensive review of the research on DTA models. Figure 2-1 presents a generic DTA model structure (Balakrishna, 2006), which consists of two main components (see, e.g., Cascetta, 2001; Florian et al., 2001):

- 1) A demand model with the ability to determine the path-dependent flow rates of paths in a network. This model estimates and predicts OD flows and drivers' decision, and subsequently converts the aggregate OD flows into individual vehicles (a.k.a. packets), which are used as input for the supply model.

- 2) A supply model capable of performing network loading. This model explicitly simulates the interaction between demand and the network, reproducing measurements such as time-dependent flow, travels times, and queue lengths.

DTA captures complex traffic dynamics and replicates traffic phenomena using the interactions between the demand and supply models.

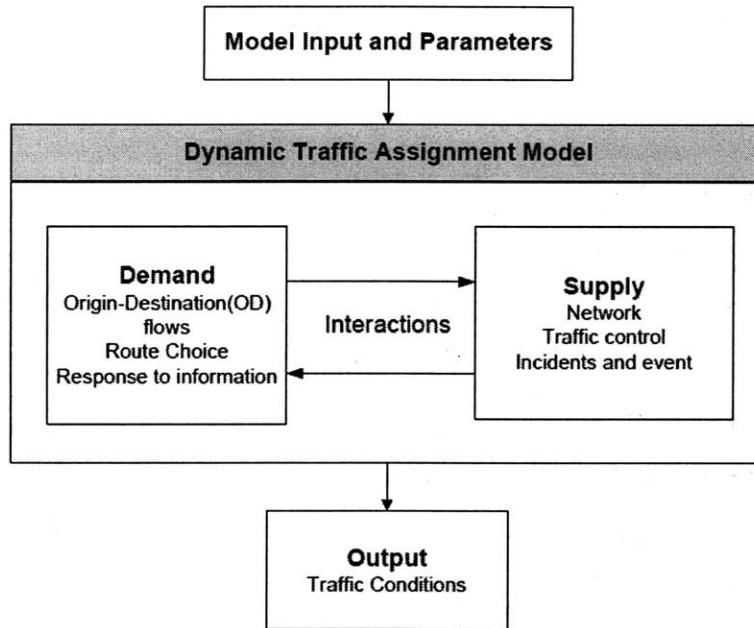


Figure 2-1: Structure of Generic DTA (Source: Balakrishna, 2006)

All DTA approaches developed up to this point can be categorized into four classes based on the nature of the modeling and solution frameworks:

- (1) Mathematical programming;
- (2) Optimal control;
- (3) Variational inequality (VI);
- (4) Simulation-based.

Models in groups one through three are considered analytical models, which generally contain mathematically proven properties that provide insights into their capabilities. While analytical models are mathematically rigorous, they rely on simplified assumptions to account for traffic dynamics, making them unsuitable for large-scale real-world applications focused on traffic congestion and management. (see, e.g., Balakrishna, 2006; Peeta and Ziliaskopoulos, 2001; Wen, 2009; Ziliaskopoulos et al., 2004)

Effective large-scale, real-world applications require models that capture the stochastic characteristics of traffic dynamics in detail, by estimating and predicting OD flows, travel times, queues and spill-backs. These capabilities are generally beyond that of

existing analytical models, and therefore, simulation is required. The rest of Chapter 2 reviews state-of-art simulation-based dynamic traffic assignment models. Their applications are also discussed.

2.2 Simulation-Based DTA

There are three types of simulation-based DTA models, which are distinguished, based on the level of detail with which they represent the studied system. They range from Macroscopic to Microscopic as follows:

- Macroscopic (Low Fidelity);
- Mesoscopic (Middle Fidelity);
- Microscopic (High Fidelity);

To begin with, macroscopic models treat traffic in an aggregate manner, such as a uniform or homogeneous flow, without considering each constituent part (individual vehicle). They propagate approximate flows throughout the network using physical concepts or analytical methods. Microscopic models depict individual entities, decisions and interactions with a high degree of detail. Each vehicle maneuvers at a specific simulation time step based on estimates derived from a list of driving behavior models, such as car following, lane changing, merging and yielding models. Mesoscopic models combine elements from microscopic and macroscopic approaches. While they represent individual vehicles with a high degree of detail, mesoscopic models depict the activities and interactions of each vehicle with less detail, in comparison to a microscopic model.

2.2.1 Macroscopic Models

Early macroscopic DTA models, such as the LWR model (Lighthill and Whitham, 1955; Richards, 1956) applied physical methodologies such as fluids theory in depicting traffic. The evolution of traffic over time and space is represented by a set of

differential equations. While analytical methods are usually used to solve these equations, they lack the ability to evaluate the interaction between vehicles and roads, leading some macroscopic DTA models to include simulations.

The common way of macroscopic simulation-based DTA is to discretised road section into cells. In Daganzo's Cell Transmission Model (Daganzo, 1994, 1995), the volume of vehicles that can cross cell boundaries at each time step is calculated using a density function on the upstream and downstream cell. The simulation tracks vehicles in each cell, recording their behavior at each time step. Another macroscopic simulation model, METACOR, which is an analytical continuum model, developed by Payne (1971) employs a similar method to describe the propagation of vehicle flows (Elloumi and et al., 1994; Papageorgiou and et al., 1989).

These macroscopic models have become increasingly popular, thanks to their efficient and easy implementation with other traffic measurement systems already installed in major urban areas and motorways.

2.2.2 Microscopic Models

Microscopic models depict traffic at a high degree of detail, explicitly modeling individual vehicles and their interactions with each other and the road infrastructure. Thus the models' dynamic variables, such as the position and velocity of a single vehicle, represent microscopic properties. Models that govern vehicle behavior included, but are not limited to, a car-following model, a lane changing model, and a route-choice model. In addition, traffic controls, including signal operation, location and traffic detectors, are also modeled in detail. However, microscopic models often require a large amount of data and the detailed calibration of model parameters, which is very time consuming, and have high computational resource demands. Examples of microscopic models include VISSIM (PTV, 2005), AIMSUN/2 (Barcelo and Casas, 2003), Paramics (Smith and S. Druitt, 1994), and MITSIMLab (Ben-Akiva et al., 2002b), etc.

2.2.3 Mesoscopic Models

Mesoscopic models generally describe traffic entities at a high level of detail, but describe the behavior and interactions of these entities at a lower level of detail.

These models can vary in form depending on how they simulate traffic. One way is to group vehicles into packets, and then route those packets through the network (CONTRAM, Leonard et al., 1989). The packet of vehicles acts as one entity, sharing the speed derived from the speed density function defined for each link, and the density on the link at the moment of entry.

Another popular mesoscopic paradigm is the queue-server approach used by some state-of-the-art models, such as : DynaMIT (Ben-Akiva et al., 1997), DYNASMART (Mahmassani et al., 2004), FASTLANE (Gawron., 1998), and DTASQ (Mahut, 2001). In this approach, road segments are modeled with a queuing and a moving part. The vehicles travel through the moving part with the speed calculated using a macroscopic speed-density function. As the vehicle moves through the segment, a queue-server transfers the vehicles downstream to another segment, or forms queues on the current segment, representing congestion. This approach combines the advantages of dynamic disaggregate traffic modeling (since the vehicles are modeled individually), with easy calibration and the use of macroscopic speed-density relationships. Signal controlled intersections can be modeled by adding gates at the end of segment that open and close according to the phase of the signal control. Adaptive signals can also be modeled (Tian, 2002). Another advantage of representing individual vehicles is the ability to model disaggregate route-choice, which is important for modeling en-route route choice and the use of traffic information and guidance in ITS systems to help drivers make plans.

Boxill and Yu (2000) presents a discussion of various traffic simulators.

Simulation-based models were first developed for the evaluation of ITS as a planning tool to generate and test scenarios, optimize control, and forecast network behavior at the operational level. As rapid technological advancements, simulation-based DTA models have become increasingly popular as a tool for real-world transporta-

tion planning. An advantage to these models is the testing of new management strategies without interrupting real-world traffic. Examples of real-world applications of simulation-based DTA models include: DynaMIT (Balakrishna et al., 2008; Ben-Akiva and Ramming, 1998; Wen et al., 2006a), DYNASMART (Mahmassani and Hawas, 1997); (Mahmassani et al., 2004), VISTA (Ziliaskopoulos et al., 2004), DynusT (Chiu et al., 2008), Dynameq (Florian et al., 2005, 2006), AIMSUN (Barcelo and Casas, 2002, 2006) TRANSCAD (Caliper Corporation, 2009), INTEGRATION (Aerde et al., 1996), METROPOLIS (de Palma and Marchal, 2002).

2.3 Congested Large-Scale Network Simulation

Over the past decade, congestion mitigation has become an important urban development concern, prompting research in simulation-based DTA to focus on the traffic modelling of congested situation. The following presents a detailed list of studies.

Tong and Wong (2000), employed a simulation-based approach to develop a method for bottleneck identification in congested capacity-constrained networks. They give a numerical example of a simple network with 6 traffic zones and 44 nodes to test the effectiveness of their approach.

Varia and Dhingra (2004) developed a supply and demand focused simulation-based DTA model for congested urban road networks that was validated on a network with 14 nodes and 38 links.

Barcelo and Casas (2003), presents AIMSUN, a microscopic simulation model, for the Borough of Amara in the city of San Sebastian. The network studied is comprised of 365 road sections with 100 junctions and intersections. 24 of the 100 junctions and intersections are signalized. There are also 13 centroids defining 135 O-D pairs, and 15 traffic detectors tracking traffic flows. The evening peak period from 18:00-20:00 was used as the simulation horizon.

Dittberner and Kerns (2002) present a simulation project for the Pennsylvania Avenue area in Washington D.C., USA, with a discussion on the key methods applied in the simulation process. The study area includes 72 intersections and the model is

calibrated using speeds and counts.

Kaysi et al. (2003), applies DYNASMART to model the network of the greater Beirut area (GBA), Lebanon, in order to evaluate the impacts of Hot Spot Management (HSM).

Ido and Prashker (2009) developed a DTA model with the capability of evaluating the travel-time and travel-path impacts of moving bottlenecks on network performance.

Xu (2009a), applied DynaMIT-P to analyze the dynamic congestion pricing model in the Lower Westchester County, NY network, using morning and evening peak hours as the simulation period.

These studies on congested situations all share a low-level network complexity. The models are tested on networks with comparatively few links and nodes. While there have been successfully deployed simulation-based DTA models on large urban network, such as TransDNA on Columbus, IN (8811 links) (Caliper Corporation, 2009), Dynameq on Tel Aviv (3700 links, 200,000 car demand/3 hours) and Ljubljana networks (8500 links, 180000 car demand/3 hours) (Mahut, 2009), these cases do not present high congestion levels. To the best of our knowledge, there have been no successful DTA applications in highly congested, large-scale urban networks.

There are three major difficulties in applying DTA systems to analyze congested large-scale networks : Computational Efficiency, Scalability, and Model Accuracy. The main challenge is the trade-off between Computational Efficiency and Model Accuracy. The processing time of a DTA system is highly sensitive to the scale of the problem, which is affected by the amount of links, nodes and demand. Studies in this field contain a wide range of approaches, from identifying computational bottlenecks for the overall DTA approach (Wen, 2009; Ziliaskopoulos et al., 2004), designing more efficient data structures and algorithms (Wen et al., 2006b; Ziliaskopoulos et al., 2004) to using distributed computing resources (Wen, 2009). However, the application of DTA systems in real-world urban networks, generally requires high-fidelity between the network representation and the actual network, which is often characterized by many short links and paths that necessitate frequent turning maneuvers, and a

simulator that can handle dense traffic volumes.

Furthermore, there is a dilemma between high- and low-fidelity network representations. Low-fidelity models are more cost-efficient in terms of computational demands, developmental effort and maintenance. However, their representation of real-world traffic dynamics may also be less accurately, and even inadequate. High-fidelity simulation models, on the other hand, are usually more accurate, but are very computationally demanding. In addition, the potential level of accuracy is not always realized because of the complexity of the system, which results in the large amount of parameters that require calibration.

As a common solution, researchers of DTA systems will make various levels of trade-offs between representational realism and computational efficiency in the demand and supply models. This allows researchers to have an adequately realistic representation of traffic dynamics and the computational ability to analyze large-scale networks. Several major DTA model studies (Ben-Akiva et al., 2001a; Mahmassani, 2001; Mahut, 2001; Taylor, 2003) have applied mesoscopic supply simulation models, which represent single vehicle movements using aggregate traffic flow relationships, instead of microscopic models.

A second major difficulty in applying DTA for large-scale congested urban networks is the realistic representation of traffic congestion. As mentioned before, many characteristics of urban networks such as complicated intersections, short links, and route choice are challenging to model. For example, non-motorized traffic, such as bicycles and pedestrians, often create significant interferences at intersections, especially in developing countries. If a DTA model does not account for this phenomenon, congestion levels at intersections are likely to be misrepresented and underestimated.

Vice versa, if short links and segments (the link length is comparable to that of a car) are inadequately represented, false queues and spillbacks may result. In addition, frequent on- and off-ramps in a network create a large amount of weaving sections. Models that do not distinguish between lane-based movements are likely to forecast non-existent traffic jams. Furthermore, when a slight overestimation of traffic flows on a route could transition the simulated traffic from a stable to an unstable stage,

any error in route choice modeling can also lead to non-existent congestion in the simulation. Therefore, in order to eliminate unrealistic gridlocks from occurring in a DTA model, these complications need to be addressed (Ben-Akiva et al., 2001a; Hughes, 2002; Ziliaskopoulos et al., 2004).

2.4 Calibration Algorithm

To accurately represent congestion, the model calibration requires real data. Usually, model inputs such as Origin-Destination (O-D) flows and a priori parameters (e.g., socio-economic characteristics, speed-density relationships for the segments or links) for different networks have different values, and are obtained from archived data. This process is referred to as the calibration of DTA models. Peeta and Ziliaskopoulos (2001) showed that the estimation and prediction of time-dependent O-D demand is one of the most difficult tasks when trying to apply DTA as a planning tool.

This thesis applies a calibration algorithm in addressing a complex congested traffic network that is highly non-linear, stochastic, and large-scaled. There are no explicit analytical relationship between the objective function with the parameter to be calibrated, making analytical derivatives with respect to the parameters difficult to achieve.

Balakrishna (2006) summarized the optimization algorithms that are potentially suitable for large-scale, non-linear problems with implicit expressions into three categories: path search methods, pattern search methods, and random search methods.

The main difference between these three is how to decide the search direction. The path search method searches along a path. In each moving step, the decision vector moves forward by a certain distance in a particular direction. Usually, the function gradient can be derived as the direction of movement. If analytical derivatives are not available, certain measurements can be used to estimate the derivatives before the subsequent step. Path search methods include: SNOBFIT (Huyer and Neumaier, 2006), FDSA (Kiefer and Wolfowitz, 1952) and SPSA (Spall, 1992, 1999, 1998), etc.

In pattern search methods, the comparison of the function evaluation is used to

determine the direction of movement for each step. It is suitable for problems in which the analytical derivative is hard to achieve because of the existence of implicit relationships. The effectiveness of pattern search algorithms are very sensitive to the efficiency of the objective function evaluation. Usually, the objective function needs to be evaluated several times before achieving a new pattern, making the process time consuming and inappropriate for large-scale problems. The Box-Complex method (Box, 1965), Nelder-Mead method (Nelder and Mead, 1965) and Hooke and Jeeves method (Hooke and Jeeves, 1961) are three typical pattern search methods.

In the random search methods, the search is performed using the entire feasible space. The decision vector is updated using probabilistic mechanisms that try to achieve optimality. Examples of random search methods include: Simulated annealing (Corana et al., 1987; Metropolis et al., 1953) and Genetic algorithms (GA) (Goldberg, 1989; Holland, 1975).

Xu (2009a) presents a detailed review and comparison of these optimization algorithms. While traffic assignment model calibration itself has drawn more attention (see, e.g., Balakrishna, 2006; Kunde, 2002; Mahut et al., 2004), few researchers have focused on specific issues in large-scale congested urban networks.

Although using the appropriate calibration method is crucial for modeling success, it is more critical to enhancement traffic models with the capability to handle specific issue since a model without the necessary features to address the full complexity of urban networks could potentially cause additional difficulties during the calibration.

2.5 DTA Model Enhancement

Boxill and Yu (2000) and Smart Project (1997) have identified limitations within the ITS framework for traffic simulation. These include the inability to appropriately simulate the congestion, environmental impact and safety, as well as computational inefficiency, etc.

Over the past decade, traffic assignment models have been greatly enhanced to meet practical requirements. A comprehensive example is INTRAS, a microscopic,

stochastic simulation model, which was reprogrammed and enhanced to include geometric and operational capabilities such as lane type, ramp metering, surveillance system, driver habits and warning signs, etc. and employs car following and lane-changing models. The new version is renamed as FRESIM and has the ability to realistically simulate freeway traffic and operations features (FHWA, 2005).

DYNASMART (Dynamic Network Assignment simulation Model for Advanced Road Telematics), a traffic assignment and optimization tool was enhanced and renamed DYNASMART-X. In DYNASMART-X, advanced network algorithms were combined with the modeling of the trip-maker's behavioral response to guidance generated in a simulation (Mahmassani et al., 2004). The most recent updates in DYNASMART-X improved the network editing interface, and its ability to capture large scale networks. Finally, a GIS importer was also incorporated into the simulator.

MITSIMLab (Ben-Akiva et al., 2002b), a microscopic traffic simulator developed by the MIT ITS program, has enhanced urban modeling capabilities for unsignalized intersections and roundabouts, drivers path awareness, and signal control and public transportation (Ben-Akiva et al., 2001b). The enhancements were validated using a case study conducted in Stockholm, Sweden.

Together with advancements in computational power, the simulation can run faster than before on a single CPU. However, it still lagged far behind in terms of the increase in network scale and traffic demand. As a result of the increased demand in effective congestion mitigation tools, researches in efficient traffic assignment model and model scalability have drawn increasing attention. Several traffic simulators, such as DynaMIT, TRANSIMS, AIMSUN and PARAMICS have been enhanced through parallel implementations. Wen (2009) provides a comprehensive review.

The traffic assignment model were developed with the main purpose of replicating real-world traffic situations for better traffic management, such as the accurate prediction and simulation of congestion and queues formulation in order to identify bottlenecks. Many of the enhancements that have been incorporated into current traffic assignment models to deal exclusively with congestion involve route choice model, queuing model, capacity model, etc. The challenges in modeling congestion include

how to handle queue spillbacks, and the interaction between queuing vehicles and moving vehicles within a single road segment. Chiu et al. (2008) implemented a lane group structure in AMS, a mesoscopic simulator, to account for the spillback from the downstream segments, which artificially impede traffic under highly congested situations.

Gentile et al. (2007) proposed a new DTA model for road networks, that explicitly addresses queue spillovers. The proposed model combines spillback congestion into an existing DTA formulation.

Another way to enhance model capability is to incorporate congestion pricing models into the existing traffic assignment model framework. Using a discrete choice framework, Xu (2009a) developed a dynamic congestion pricing model, which was implemented in DynaMIT-P and subsequently improved the behavioral choice models. The model enhancement was validated on the Lower Westchester network in New York.

In summary, while the demand for tools to address traffic congestion have increased as a result of accelerated urban development, traffic assignment models that can adequately meet the needs of a large-scale, high-density urban network are still lacking. This thesis presents modeling processes, and a subsequent enhanced model that can replicate the traffic dynamics in a highly congested complex urban network.

Chapter 3

Modeling Challenges and Solutions

In a recent study on dynamic traffic assignment models and traffic simulation using DynaMIT-P for the city of Beijing (Ben-Akiva et al., 2010), researchers encountered gridlocking problem when the traffic demand was increased to reflect reality. Non-existent queues formed in lanes along the freeways and arterials. Few cars could get to their destination, and most of the sensor counts reported in the simulator were close to 0.

The study area in Beijing is a large urban network in an area of around 35 square miles, consisting of freeways, ramps, and signalized arterial roads. These roads are modeled as 3180 directed links. Some of them are unusually short - as short as 20 feet, which caused further complication in the simulation.

The traffic load for this network is also very heavy. On a typical weekday, the model reproduces roughly 630,000 vehicles traveling within the study network between 6:00 AM to 10:00 AM. Severe congestion and spillbacks frequently occur throughout the network. During peak hours, the traffic density in many areas approach jam density, resulting in extremely low speeds. Modeling such conditions is difficult because the congestion is so severe that a small over-estimation in demand or a small under-estimation in supply would result in a grid lock.

Moreover, the mixed traffic conditions in Beijing are unparalleled by previous studies. Most urban residents commute via bicycle to and from work, accounting for a large portion of the traffic. In 2007, the total volume of bicycles in Beijing was

approximately 50.94 million (Sun, 2009). The mix and interaction of different transportation modes result in capacity reductions and intersection delays. This thesis is motivated by the need for a model that can properly simulate such a network and address the current model limitation for better traffic management. The remainder of this chapter discusses the identified problems identified and the corresponding enhancement solutions for these specific network modeling problems. Furthermore, generalizations of these challenges and solutions applicable to all traffic network modeling cases with similar large-scale, highly congested characteristics are presented.

3.1 Revised Route Choice Model

DynaMIT-P was used for the modelling of the Beijing case study. Model input includes the traffic network of the study area, dynamic OD flows, Speed-Density relationship parameters for each segment and the route choice parameter. From the initial simulation results, which were obtained before any model enhancements, we observed excessive congestion on the freeways (ring roads) but little flow on the parallel roads. The links on freeway usually experienced jam density and low speed near to 0, while most of the arterial roads had little traffic on them. A further analysis showed that route choices were biased toward the ring roads, and the adjustments to the route choice model parameters had little effect in mitigating the bias. An investigation on the current route choice model had to be done before any enhancements.

3.1.1 Logit route choice model

A Logit route choice model was initially adopted to calculate path choice probabilities in DynaMIT-P. This model is simple but has a critical limitation in terms of the assumption that the error terms are identically and independently distributed (i.i.d.). Under the route choice context, this assumption requires that all alternatives have to be independent. A complex urban network like the Beijing network does not abide by this assumption because there are a large number of overlapping paths that share the same freeway links.

3.1.2 C-Logit model

In order to overcome such a problem, previous researchers proposed the addition of a deterministic correction term to the utility function for overlapping paths. The first attempt was made by Cascetta et al. (1996). A “commonality factor” (CF) term was added to the deterministic part of the utility that captures the degree of similarity between alternatives in the choice set. $P_n(i)$ - the probability of user n choosing path i among his/her individual choice set $C_n(i)$ - is defined in Equation 3.1:

$$P_n(i) = \frac{e^{V_{in}-CF_{in}}}{\sum_{j \in C_n} e^{V_{jn}-CF_{jn}}} \quad (3.1)$$

3.1.3 Path Size Logit Model (PSL)

Motivated by the C-Logit model, Ben-Akiva and Ramming (1998) proposed the Path Size Logit Model (PSL) (see also Ben-Akiva and Bierlaire, 2003), which has a similar form but uses a “Path Size” (PS) attribute to correct for the utility for overlapping paths. The PS attribute was originally derived from the discrete choice theory for aggregate alternatives (Ben-Akiva and Lerman, 1985). Ben-Akiva and Ramming (1998) defined the correction term PS_{in} as the “size” of path i (see Equation 3.2):

$$PS_{in} = \sum_{a \in \Gamma_i} \frac{l_a}{L_i} \frac{1}{\sum_{j \in C_n} \delta_{aj}} \quad (3.2)$$

where Γ_i is the set of links on path i , l_a the travel time on link a , L_i the total travel time on path i , C_n is the choice set of paths for individual n , and δ_{aj} a binary variable which equals 1 if link a is a part of path j and 0 otherwise. For a path not overlapping with any other path, the path size is 1, and the systematic utility is not adjusted. For a path partially overlapping with other paths, the path size is less than 1, and the systematic utility is adjusted downwards. For a path completely overlapping with $J - 1$ other paths (J being the size of the choice set C_n), the path size is $1/J$. Once the PS attribute is defined, the utility associated with path i for individual n is adjusted as $V_{in} + \ln PS_{in}$, with the probability $P_n(i)$ computed as in

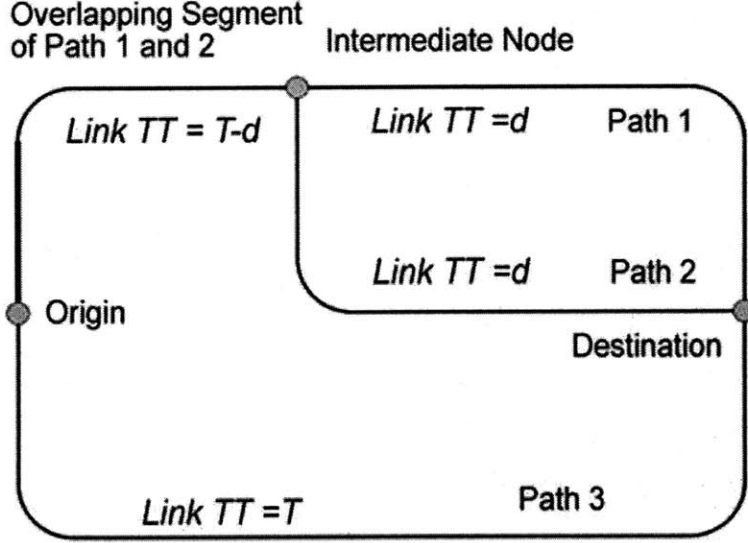


Figure 3-1: The Overlapping Path Problem

Equation 3.3:

$$P_n(i) = \frac{e^{V_{in} + \ln PS_{in}}}{\sum_{j \in C_n} e^{V_{jn} + \ln PS_{jn}}} \quad (3.3)$$

$P_n(i)$, C_n and V_{in} are similarly defined as in Equation 3.1.

The example shown in Figure 3-1 was used by Ramming (2002) to illustrate the overlapping path problem and how the PSL choice compares against other model types, especially with MNL model. There are three paths with the same total travel time T . Path 1 and 2 overlap from the Origin Node to the Intermediate Node with an overlapping travel time of $T - d$. The MNL model predicts equal shares for the three paths, one third each. This is correct only when $d = T$, when there is no overlap.

The choice probabilities for the overlapping path (Path 1 or 2) are presented in Figure 3-2. The horizontal axis is the fraction of T that d represents. When the proportion approaches 0, that is $d = T$, Path 1 and 2 are the same physical path with two separate "labels". In this case, we expect the combined choice probabilities for Path 1 and 2 to be 50%, and Path 3, the other physical path, should have a choice

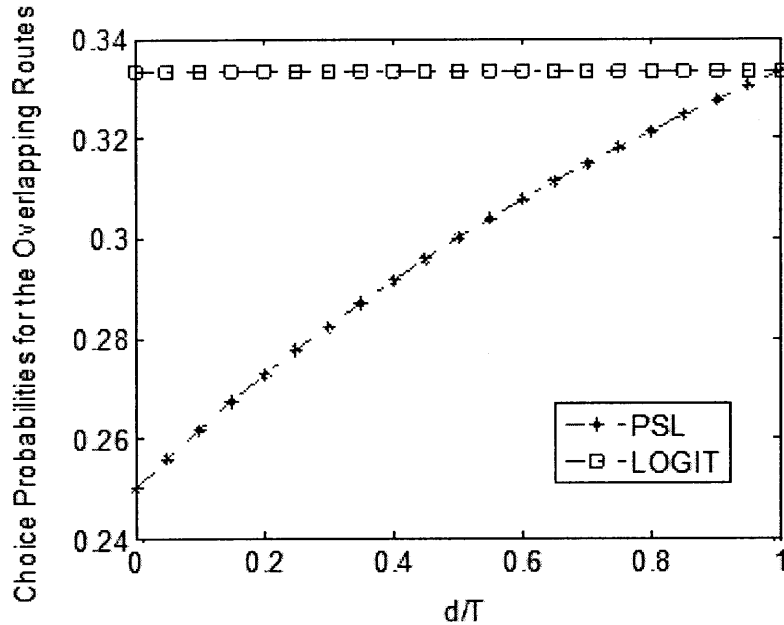


Figure 3-2: Choice Probabilities for the Overlapping Path Network

probability of 50%. The PSL reflects our expectation, but the MNL model gives a flat choice probability of 33% which means that it is not sensitive to the overlap.

Now apply the above example in complicated urban networks. Consider a traveler is planning his/her route choice from home to work. The overlapping segment can be viewed as the freeway, and there are N different paths to the work place after getting off the freeway. In addition, he/she has the choice to use arterial roads the whole way. In an extreme case, assume all paths have the same total travel time, and the overlapping segment is long enough. We can expect 50% probability that s/he will choose the freeway, which concurs with the prediction generated by the PSL model. However, the MNL model will give a probability of $\frac{N}{N+1}$ to the freeway, and a probability of $\frac{1}{N+1}$ to the arterial. This bias will lead to a non-existent higher demand for the freeway, and cause unrealistic congestion in simulation. N is potentially very large in a dense urban network and the bias could be very serious.

3.1.4 Result

The bias mentioned in the previous section was verified in the simulation of the Beijing network. Figure 3-3 shows the path set generated by DynaMIT-P for a typical OD pair (1616-1584). There are 8 paths in the path set, 7 of which share the ring road (the freeway from north to south), except the 8th Path. Consequently, as discussed above, it is very likely that the MNL route choice model over-predicted the probability of choosing paths containing ring road links, which led to unrealistic congestion along these roads. To prove the hypothesis, the path choice probabilities generated before and after the PSL model was enhanced, were calculated and are shown in Table 3.1 (the results may vary when the route choice parameter changes). Note that, before the enhancement, the probability of choosing a path containing the ring road is 98.74% (the sum of the probabilities of the first 7 paths). The probability of choosing the 8th path is only 1.26%. However, since the difference in average travel time between the 8th path and other paths are not significant enough (781.4s vs. 735.1s) to cause the traveller to refrain from choosing the 8th path, we should expect a much higher probability for choosing the 8th path than the MNL model predicts here.

Table 3.1: Path Choice Probabilities Before/After PSL Enhancement

Path No.	Travel Time(s)	Path Size	Prob. Before	Prob. After
1	719.4	0.149	10.03%	2.61%
2	841.2	0.326	0.17%	0.22%
3	732.3	0.201	6.52%	3.10%
4	665.8	0.329	60.11%	76.49%
5	731.1	0.173	6.79%	2.39%
6	726.8	0.179	7.84%	2.95%
7	729	0.197	7.28%	3.32%
8	781.4	0.774	1.26%	8.93%

To correct for this bias, the route choice model was modified from MNL to PSL. The path choice probabilities after this enhancement are also shown in Table 3.1. The new results show a 8.93% probability of choosing a non-ring road path (the 8th path), and among those who would use the ring road, more people would choose the path with shortest travel time (path No.4).

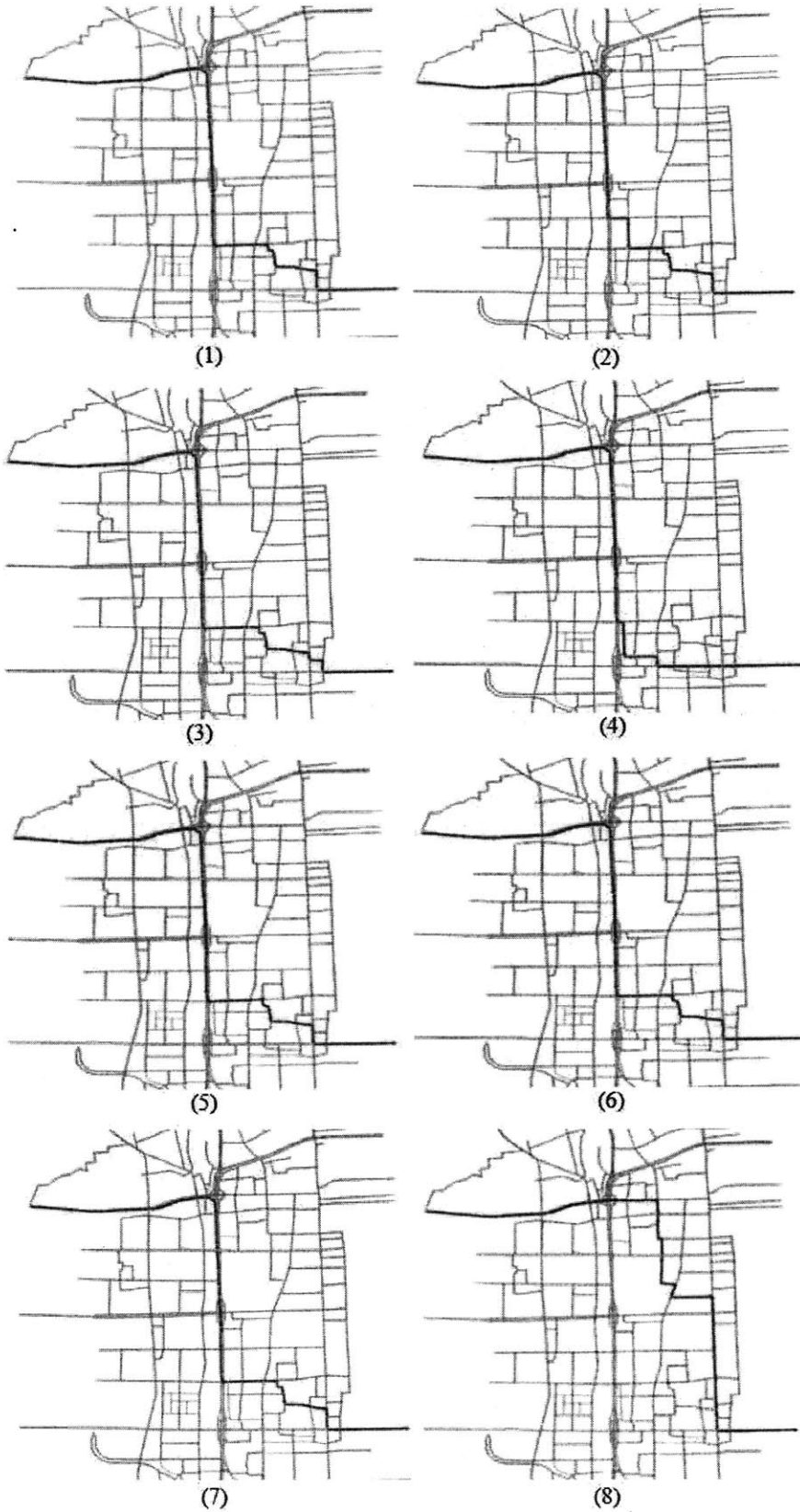


Figure 3-3: Path set for OD pair: 1616 - 1584

After this enhancement the gridlock situation in the Beijing case study was resolved. In addition, we observed more realistic route choice behavior in terms of traffic flow and density distributions on the network.

3.2 Lane-Group-Based Queuing Feature

In the mesoscopic supply models used by some DTA systems (including DynaMIT-P), links are divided into segments in order to capture changing section geometries. Each segment may contain a moving part at the start of the segment and a dynamic queuing part at the end of segment. The length of the moving part and the queuing part may vary as the simulation proceeds. In the moving part, in-flow vehicles move at a certain speed governed by the speed-density relationship, which is propagated till the end of the queue. In the queuing part, (lane- or link-based) queues are formed following a queuing model.

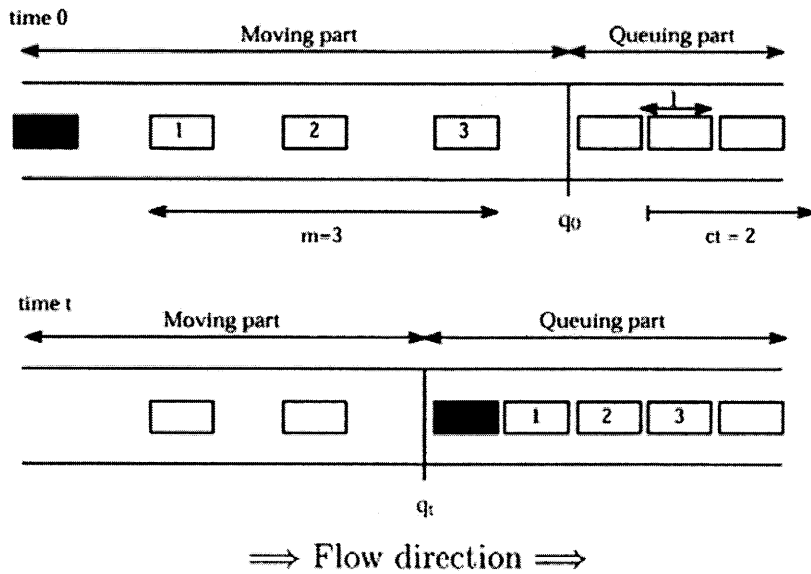


Figure 3-4: Deterministic Queuing Model

A deterministic queuing model is illustrated in Figure 3-4. (Ben-Akiva et al., 2001a) During a period of time t , there are ρt vehicles leaving the queue, where ρ is the output capacity of the segment. At time t , given that there is a vehicle reached

the end of queue, the position of the end of the queue is calculated as:

$$q(t) = q_0 - l(\rho t - m) \quad (3.4)$$

Where q_0 is the position of the end of the queue at time $t = 0$, l is the average length of vehicles(including headways), m is the number of moving vehicles between the vehicle being considered and the end of the queue at time $t = 0$, under the assumption that the position of the end of the upstream segment is 0. Note that the model is relevant only when $0 < q(t) < L$, where L is the link's length. If $q(t) < 0$, then the queue has already dissipated by time t . The explicit spill-back model guarantees that $q(t) \geq L$ will never occur. When a vehicle is approaching the end of the queues in a segment, it will try to merge into a feasible lane that has the shortest queuing length. A set of feasible lanes for a given vehicle is defined as a set of lanes that can reach the downstream segment according to the vehicle's path to the destination.

In most conventional supply models, a dynamic queue is formed at the link level. The lane connections are simplified based solely on link connectivity. In other words, if two links are connected, all lanes on the upstream link are connected to those in the downstream link. DynaMIT was initially designed with the capability for flexible network representation (Ben-Akiva et al., 2001a), which allows lane-based queuing and the organization of lanes going in the same direction into lane groups. However, due to practical reasons, this feature was rarely utilized in previous DynaMIT applications. Most applications employed the link-based queuing model.

Liu et al. (2008) discussed the drawbacks in the link-based queuing model and proposed a set of lane-group-based macroscopic formulations to address them. One of these drawbacks is the blocking effects between different movements. Specifically, queues that spilled back from the downstream link can mistakenly be allowed to form in all lanes in the upstream link. For instance, the left turn queue may block the through and right turn traffic because the model will generate a single queue for the link, regardless of the turning movement (Figure 3-5). Chiu et al. (2008) also

presented the lane group structure in AMS, a mesoscopic simulator. This structure is designed to account for spillbacks from downstream links and ensure that vehicles located in turning bays do not artificially impede through traffic, unlike the link-based queuing model.

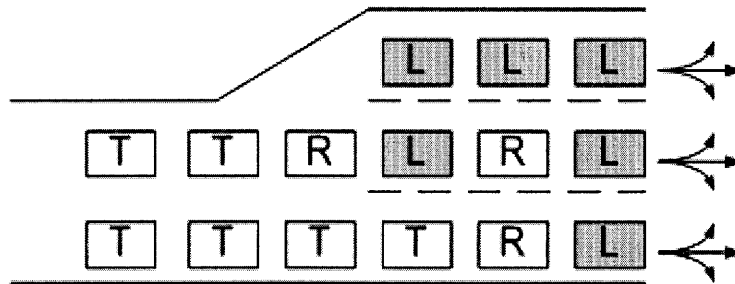


Figure 3-5: Left-turn lane block the through and right turn traffic

The blocking effects may not be significant in typical highway models, where such intersections are rare. However, it can cause severe problems in the simulation of urban networks such as the one in Beijing. In such urban networks, lane restrictions at complicated intersections are commonly seen. Moreover, on- and off-ramps connecting the freeways and side roads are often highly congested. Consequently, lane restrictions have significant impacts on the throughput of those intersections. If such restrictions are not modeled properly, unrealistic congestion caused by exit queues blocking through traffic could result in the simulation, as they did in our initial our case study.

Figure 3-6 demonstrates a typical scene from the early stages of our study before a lane-group-based queuing feature being implemented. The segments with red color denote a jam density, while the green shows a segment with low traffic volumes. Due to the downstream spillback, segment 343 is filled up with queuing vehicles, showing a jam density with flow near to zero. The queue extended to segment 422 and segment 420. In this case, segment 420 is an off-ramp connected to freeway segment 419 with service road segment 422. Segment 419 consists of 3 lanes, while segment 420 only has one lane. The lack of lane groups classifications for segment 419 (only one lane group on 419) resulted in the spillback from segment 420 mistakenly form in all 3 lanes on segment 419. Therefore, the through traffic was blocked from getting

onto the downstream freeway segment 428. An unrealistic bottleneck formed in this area. The queue and congestion accumulated along the freeway and throughout the network. Finally, the traffic on most parts of the 2nd ring road became paralyzed in the simulation.

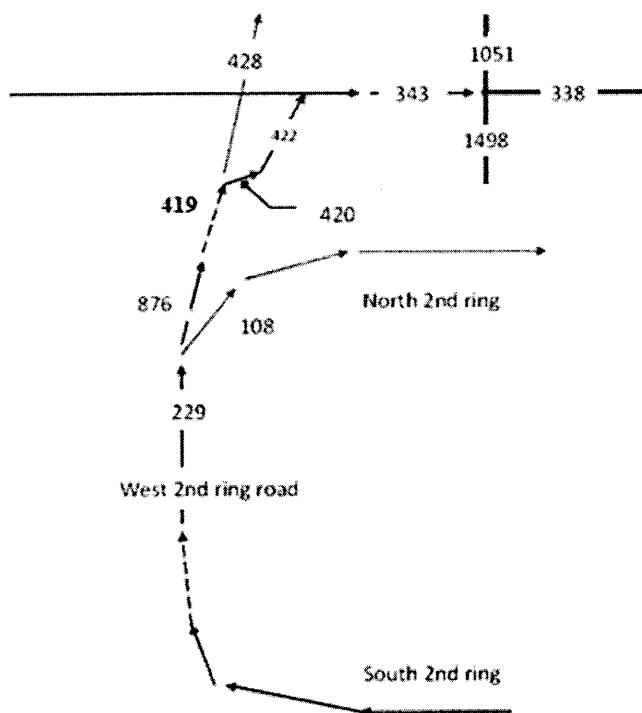


Figure 3-6: Off-ramp block the through traffic on the freeway

To address the aforementioned issue, our initial DTA model was enhanced to capture the lane restrictions through "lane groups". A lane group is defined as a set of lanes established at an intersection approach for separate capacity and level-of-service analysis (Highway Capacity Manual, FHWA (2000)). In a lane group model, lanes are grouped according to their specific turning movements. For example, Figure 3-7 shows an approach to an intersection with a queuing part that is comprised of three lane groups: left only, through, and right only. No lane group is required for the moving part of the link in our model. The advantages of this representation are twofold: 1) the moving part is not over-complicated and capitalizes on the efficiency of mesoscopic traffic simulation; and 2) the queuing part is based on lane groups and thus spillbacks onto one lane group will not mistakenly impact movements on other

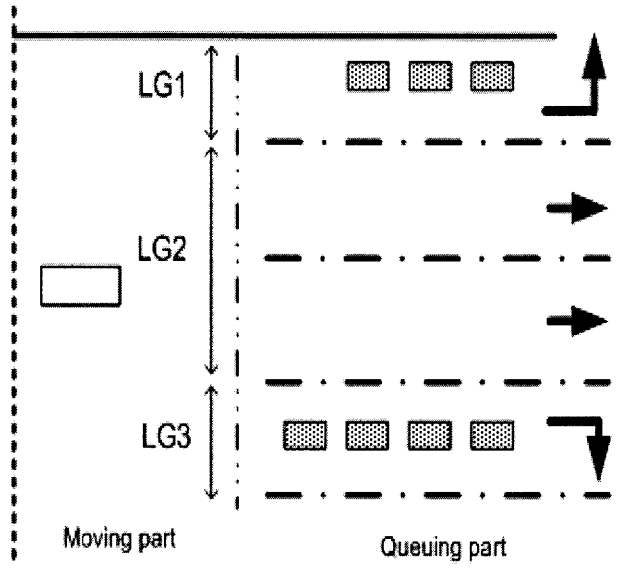


Figure 3-7: Group of lanes according to turning movement



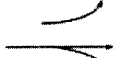

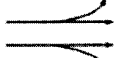
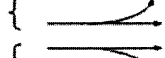

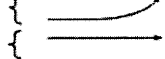
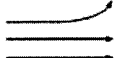
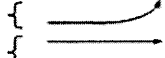
lane groups.

Most existing mesoscopic simulation studies employ link-level implementations of queues, and lane groups or similar constructs are rarely seen. This is probably because networks and lane restrictions for many previous studies were not complicated enough to entail the additional efforts (usually non-trivial) required for the implementation of explicit lane groups, such as determining the boundary of the moving and queuing part, grouping the lanes according to lane restrictions and setting demand capacities for each lane group.

To effectively apply the lane-group-based queuing feature, the construction of lane groups should account for both geometry changes and lane restrictions at intersections. We generated the lane connections following the guideline given by the Highway Capacity Manual (HCM 2000). Table 3.2 shows the typical lane groups for analysis.

For the Beijing study, with a queuing part based on lane groups, the mistakenly formed queues resulting from the limited capacity of another lane group were resolved, and a large number of unrealistic bottlenecks disappeared. This feature combined with the Path Size Logit model contributed significantly in resolving the unrealistic gridlocks generated in the simulations.

Table 3.2: Typical Lane Groups for Analysis(Source: HCM 2000)

Number of Lanes	Movement by lanes	Possible Lane Groups
1		
2		
2		
3		
4		

3.3 Treatments for Short Links

As it is in the Beijing network, short links are not uncommon in the traffic simulation. See Figure 3-8 for an example of a complicated interchange with short links in the Beijing network. Short links are known as links with relatively short length, usually that of a few cars and sometimes even shorter than the length of one car. Short links in traffic simulation can be generally divided into two categories: 1) the short links exist in the real-world network; 2) the short links do not exist in the real-world network. Examples of the first group are links at an at-grade intersection, connecting two same-direction roadways separated only by a divider, or the links connecting different direction roadways to capture turning movement(such as u-turns). In this case, the short links usually occur on intersections and ramps.

Short links that do not exist in the real-world network are usually generated during the network coding process to capture geographic feature, such as the curvature of the roadway. While some state-of-the-art simulators like DynaMIT can represent link curvature with an additional "Bulge" attribute for the link and it is unnecessary

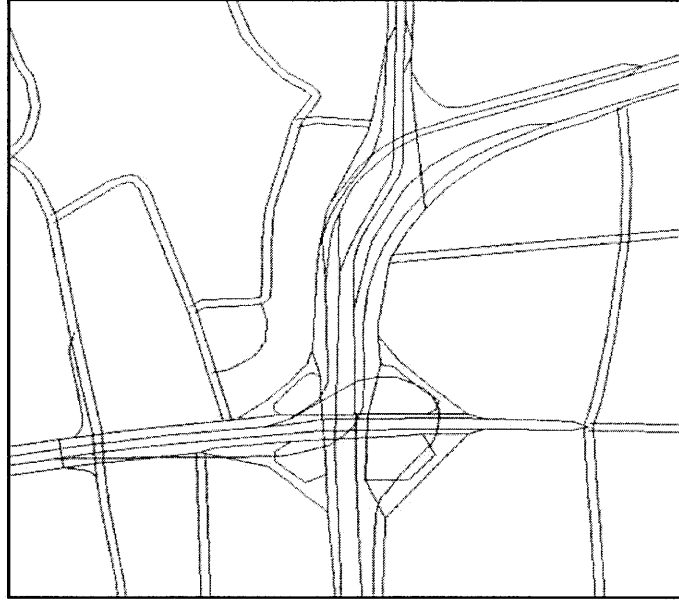


Figure 3-8: A complicated interchange with short links

to convert an arc link into several short links, most simulators and network provider still use short links to represent the curved road/line. One concern is that using polyline will be easier for the model network to be compatible with a GIS database (Geographic Information System) which only employs polylines and nodes to represent networks. For traffic assignment models, these network representations are designed to maintain high fidelity to the real-world road network. Therefore, the short links, whether they exist in reality or not, will be kept in the model. This in turn poses challenges for the model in replicating realistic traffic dynamics. During the early stages of our simulation study in the Beijing network, we observed that abnormal congestion originated from these short links. The congestion resulted from two types of problems: 1) vehicles moving unrealistically slow on short links and 2) vehicles queuing unnecessarily in the upstream of a short link. The causes for the two problems are different, and are discussed separately in the rest of this section.

3.3.1 Increase the Minimum Speed

An abnormal phenomenon we observed was that vehicles moved at a normal/high speeds on the relatively long links, while they decelerated to an unnecessary slow

speed on the short links. Before investigating the cause of the problem, it is necessary to review how the model calculates the speed of a vehicle moving on a road segment.

In mesoscopic traffic simulators of many DTA models, the speed of a vehicle in the moving part of a segment (see section 3.1.2 for definition) is governed by the macroscopic speed-density relationship, which takes a form similar to Equation 3.5(Used in DynaMIT-P):

$$v = \max \left\{ v_{\min}, v_{\max} \left(1 - \left(\frac{\max(k - k_{\min}, 0)}{k_{jam}} \right)^\beta \right)^\alpha \right\} \quad (3.5)$$

where v is the speed to be determined, k the density, v_{\min} the minimum speed on a segment, v_{\max} the maximum speed on a segment, k_{\min} the density below which the speed is fixed at v_{\max} and k_{jam} the jam density. $v_{\min}, v_{\max}, k_{\min}, k_{jam}$ are segment parameters to be calibrated.

In traffic flow theory, the speed-density relationship is assume to be continuous. The density(k) could take an infinite number of values between 0 and k_{jam} so as to capture the traffic dynamic for different segment states (from idle to congested). However, in the simulation, the density of a segment usually takes discrete values derived using Equation 3.6:

$$k = \frac{N}{N_{\max}} = \frac{N \times l_{veh}}{n_{lane} \times L} \quad (3.6)$$

Where the N is the number of vehicles in the moving part of the a segment, N_{\max} is the maximum number of vehicles that the moving part can hold, l_{veh} is the average length of the vehicles. Density(k) is calculated using the total length of moving vehicles divided by the total space of the moving part which is the number of lanes n_{lane} times the length of the moving part L (from the beginning of the segment to the end of the queuing part).

For short links, however, this means that the density will jump to be close or even equal to the jam density (k_{jam}) whenever a car enters the link. For example,

if the link contains one segment and its length equals to that of a typical car, then the segment only has two states: idle or jammed. In addition, the density k can only take two values $\{0, k_{jam}\}$, which results in two possible calculated speeds:

$$v = \begin{cases} v_{\max}, & \text{if } k = 0 \\ v_{\min}, & \text{if } k = k_{\max} \end{cases} \quad (3.7)$$

This implies that the speed on this segment will drop to v_{\min} anytime the segment is occupied, effectively making the average simulated speed equal to v_{\min} . Hence if v_{\min} is lower than the average observed speed, this segment will almost always cause unrealistic congestions.

Since the minimum speed v_{\min} is one of the variables to be calibrated, one possible solution is to increase the starting value of the calibration variable (Minimum speed) to the average observed speed for each of those short links, and to restrict this value from deviating too far from the mean throughout the entire calibration. This method was effective for the Beijing study.

3.3.2 Dynamic Acceptance Capacity

In some mesoscopic traffic simulators, queues are caused by either a limited output capacity of a segment or a limited acceptance capacity of its downstream segment, whichever is binding. In other words, the effective capacity of segment j at time t , denoted as C_{eff}^{jt} , is computed as in Equation 3.8

$$C_{eff}^{jt} = \min(C_{out}^{jt}, C_{acc}^{j't}) \quad (3.8)$$

where C_{out}^{jt} is the output capacity of the current segment at time t , and $C_{acc}^{j't}$ the acceptance capacity of segment j 's downstream segment j' at time t . Please note that if lane-groups are used, then all capacities should be defined at the lane-group level and Equation 3.8 holds by lane group. For simplicity and without loss of generality, we assume the capacity is defined at the segment level here. Note also that for time-based simulation models, "time t " actually means "time-step t ".

$C_{acc}^{j't}$ is determined by the available space that the downstream segment j' has. The more vehicles are on the downstream segment, the less acceptance capacity it has. For time-based simulation models, the acceptance capacity of a segment at time t is often computed from the segment's available space at time $t - 1$, as shown in Equation 3.9:

$$C_{acc}^{j't} = \frac{L^{j'} \bullet m^{j'} / \bar{L} - n^{j'(t-1)} + \Delta n^{j't}}{\Delta T} \quad (3.9)$$

where $L^{j'}$ is the effective length of segment j' , $m^{j'}$ the number of lanes in segment j' , \bar{L} the average effective vehicle length, $n^{j'(t-1)}$ the number of vehicles on segment j' at time $t - 1$, $\Delta n^{j't}$ the (expected) number of vehicles to move out of segment j' between time $t - 1$ and t , and ΔT the time-step size. Note that the number of vehicles moving out of segment j' depends on the speed and capacity of the current segment and its downstream segments. Therefore, at the beginning of the time-step t , $\Delta n^{j't}$ is generally unknown, unless the current and downstream segments have been processed in the simulation. Typically, the simulator may need to “guess” the value of $\Delta n^{j't}$. For example, one could use the number from the previous time-step ($\Delta n^{j't-1}$) or even simply assume zero.

When the assumed $\Delta n^{j't}$ is smaller than the actual value, the acceptance capacity is effectively underestimated. However, one may argue that the impact of $\Delta n^{j't}$ is not as significant as it appears. Roughly speaking, if the network is not too congested and segment j' has sufficient space, then the binding constraint in Equation 3.8 is likely to be $C_{out}^{j't}$ (as long as $C_{out}^{j't} < C_{acc}^{j't}$). On the other hand, if the network is congested and segment j' does not have much available space, then $\Delta n^{j't}$ is likely to be small, as vehicles tend to move slowly and downstream segments of segment j' may be queued to prevent a fast discharge from segment j' . Therefore, under most circumstances, the underestimation caused by $\Delta n^{j't}$ is insignificant.

While the above argument may hold when the segment is long enough, it is not the case for short segments. Suppose the downstream segment j' can only hold one vehicle: $L^{j'} = \bar{L}$, and $m^{j'} = 1$. If $\Delta n^{j't}$ is ignored, then whenever there is a vehicle

on it (i.e., $n^{j'(t-1)} = 1$), its acceptance capacity computed from Equation 3.9 is zero - as in the case “no more space is available on the segment”. In reality, however, if the vehicle is moving, as soon as it moves out of the segment, the segment can accept another vehicle.

Failing to recognize this inaccuracy in calculating acceptance capacities may over-predicate congestion. Because of computational efficiency, in models designed for large-scale networks or real-time applications, for computational efficiency considerations, typically the acceptance capacity is not updated every time a vehicle is moved, but rather assumed constant for a short period of time (such as a minute). In such cases, the acceptance capacity stays zero at each time-steps of the whole update period, and practically blocks the upstream traffic unnecessarily. One solution is to ignore the acceptance capacity constraint when there is no queuing in the downstream segment, i.e., use Equation 3.10 instead of Equation 3.8

$$C_{eff}^{jt} = \min(C_{out}^{jt}, \delta_q^{j't} M + C_{acc}^{j't}) \quad (3.10)$$

where $\delta_q^{j't}$ is a binary variable equal to 1 if there is no queue in segment j' at time-step t and 0 otherwise, and M is a sufficiently large positive number (practically positive infinity). Equation 3.10 is equivalent to Equation 3.8 when there is a queue in segment j' ; however, when there is no queue, $\delta_q^{j't} M + C_{acc}^{j't}$ is always greater than C_{out}^{jt} , thus the acceptance capacity is ignored.

3.3.3 Result

After accounting for vehicles' moving state, a revised capacity model that treats specifically for short links was implemented in the DTA model, and the abnormal queuing phenomenon was eliminated. Examples exhibiting the change of state for a typical freeway short link, in terms of density, speed and flow, are given in the following.

A typical short link on freeway (link #372) was picked out from the Beijing case study for the demonstration shown in Figure 3-12 (the red line). This link contains



Figure 3-9: A Typical Freeway Short Link #372

one 3-lane segment that is 15.3 meters in length(roughly equals to 3 vehicles length). in total, this link can hold 9 vehicles at a time. Figure 3-9,3-10,3-11,3-12 compare the simulation results of this segment before/after the enhancement. Before the treatment for short links was implemented, the segment density fluctuated dramatically between 0 and jam density(k_{jam}). Most of the time, the speed can either be free flow speed or minimum speed (13 mph in this case). Note that, in Figure 3-11, when the speed is shown as "free-flow" speed, actually there are no vehicles on the segment. As a result, the flow is constrained within a range with a mean of 1000 veh/hour.

The red dotted lines demonstrate the simulation results after the treatments for short links were employed. The abnormal phenomena disappeared. The density of this segment is much less than before. Vehicles traverse on this segment at a higher speed, and the flow has increased to more than 3000 veh/hour, which is almost 3 times than before. These changes indicate that there are fewer vehicles congested on this short segment after the enhancement, and the effective capacity of this segment has increased significantly.

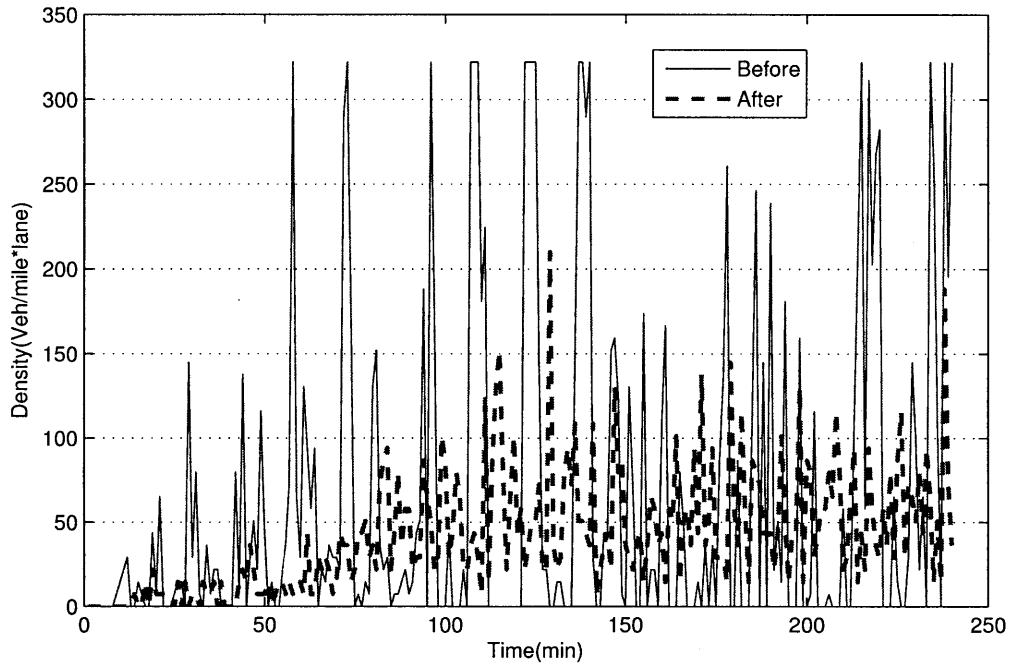


Figure 3-10: Density Change on Short Link #372

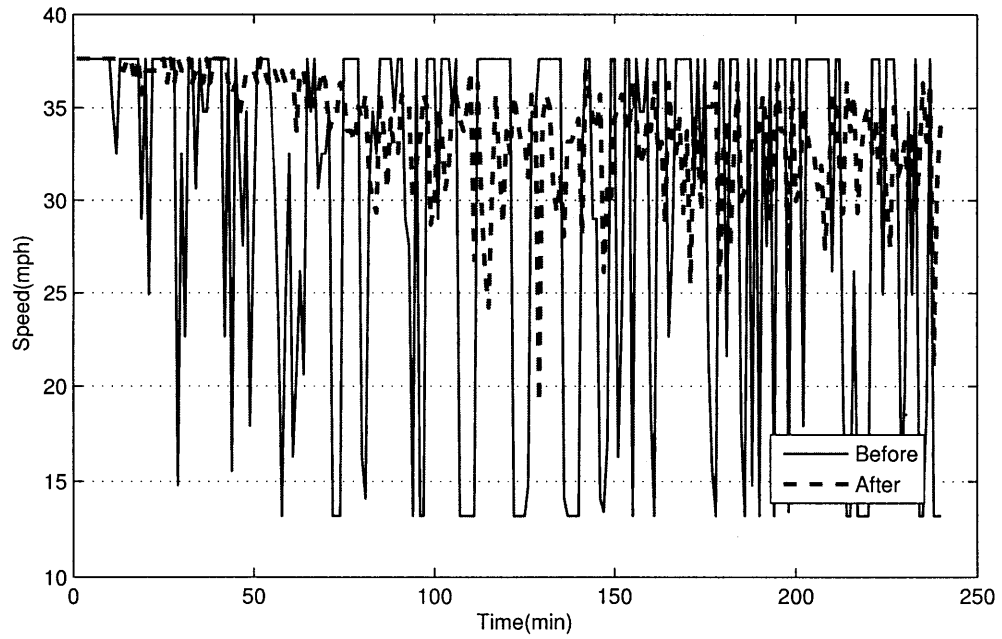


Figure 3-11: Speed Change on Short Link #372

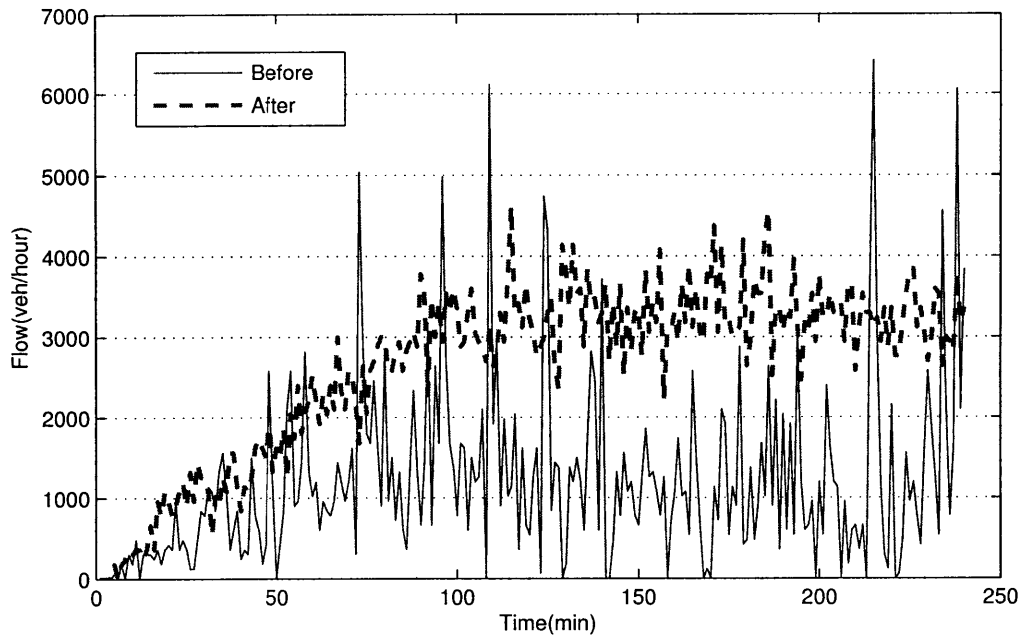


Figure 3-12: Flow Change on Short Link #372

3.4 Centroid (Loader) Access Capacity

Traffic simulators load vehicles onto the network or remove them from the network at specific nodes called "*Centroid*" or "*Loaders*". A centroid or loader is usually assigned to the node that serves as an OD point in the network, and is used as the network representation for the Traffic Analysis Zone (TAZ). The placement of the centroid is designed to replicate the center of activity within the TAZ. Both the loader and centroid are where the trips are generated from and are traversing to. The rest of this section uses these two terms interchangeably.

Every loader has two important attributes "loader input capacity" and "loader output capacity". The rate at which vehicles can be loaded onto the network is referred to as "loader input capacity" and the rate at which vehicles can be removed from the network is known as "loader output capacity". Usually these two values are specified as model parameters in the simulator, and can be determined and set to appropriate values according to the type of loader. For example, higher rates (closer to 2200 vehicles/lane/hour) correspond to freeway loaders, while lower values

correspond to arterial loaders.

3.4.1 Adding Centroid Connector

Another important concept that is easily overlooked in the modeling process is called centroid (loader) access capacity. The centroid (loader) access capacity is defined here as the total capacities for link access to a centroid (loader), or the total capacities for link access from a centroid (loader). One problem that needs to be addressed during the network coding process is how to reserve enough access capacity for a centroid. During the network coding process, not all roadways are included in the network. Local or neighborhood roads are typically excluded from the model. These roadways are represented in general by links called *centroid connectors*. By definition, the access capacity for a centroid is defined as the total capacity of centroid connectors.

The location and number of the centroid connectors can have a significant impact on how traffic is assigned to the network (Ismart, 1990). Maintaining enough centroid connectors is a way to provide sufficient centroid access capacity. Lacking sufficient centroid connectors may result in artificial bottleneck in the network. An example is given in Figure 3-13, 3-14, 3-15, which shows an area of the network in the Beijing case study that exhibited severe bottlenecks because of the lack in centroid access capacity.

Figure 3-13 shows a map of the real network for this study area (From Google Map). There are many local roads (the white lines) in this area with access to the freeway (light and dark yellow lines). Figure 3-14 is a simplified computer network representation of this area. During the network simplification process, a centroid (1741) was designated to represent the aggregate trip demand and the local roads are excluded from the abstracted network. Two roads were left connecting the centroid to the network. However, this process is problematic because it removed the capacities of these local roads, which exist in reality. The total remaining capacity of the roads connected to the centroid is now less than 2400 vehicles/hour, while the trip demand attracted by this centroid is more than 4000 vehicles/hour. The consequence is shown in Figure 3-15. Red denotes the high density and queuing on the road. Queues

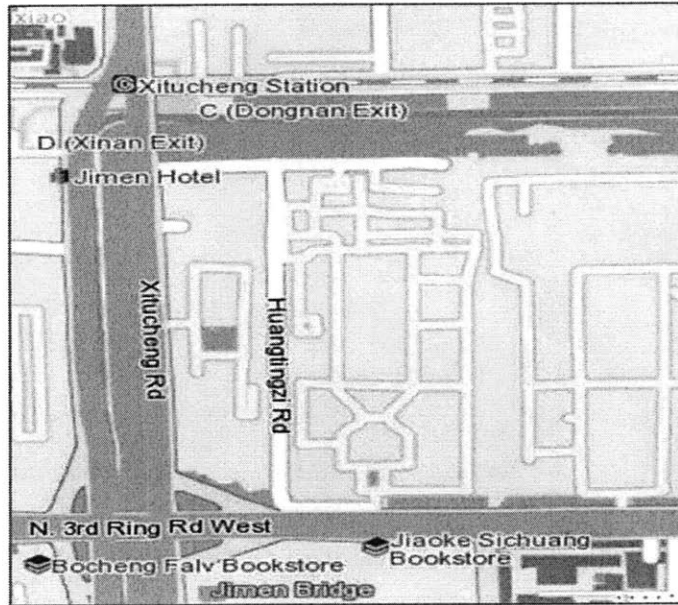


Figure 3-13: Real world network of the area of interest

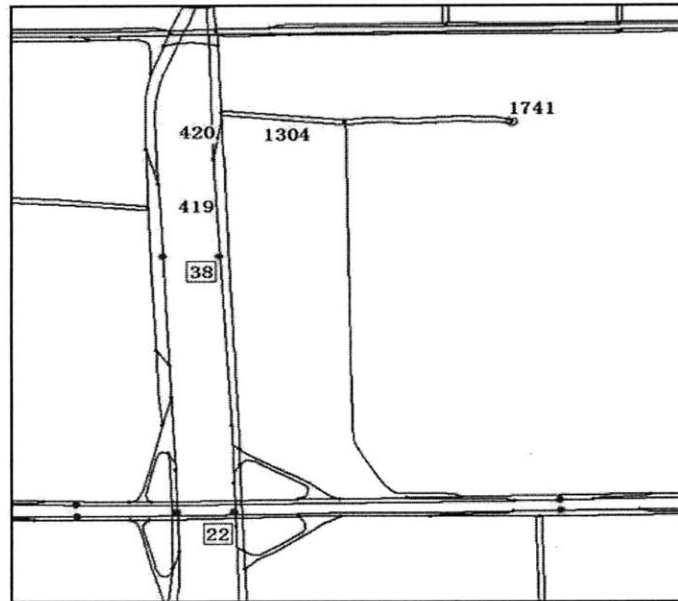


Figure 3-14: Computer representation of the area of interest

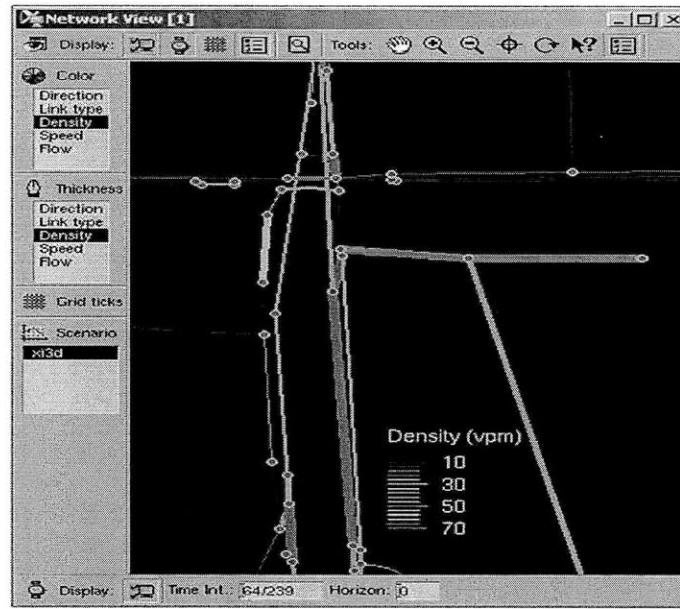


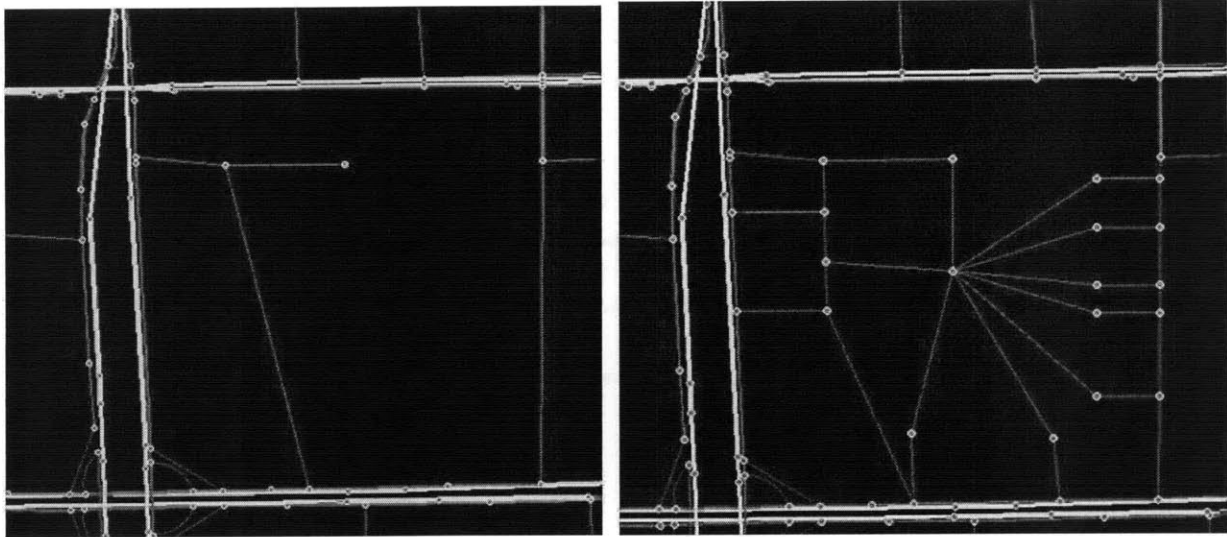
Figure 3-15: Simulation result of the artificial bottleneck

formed on the connector link and spilled back to the ramp and freeway. The network simplification process artificially created a bottleneck in the network, resulting in unrealistic congestion in the simulation.

While this problem is critical to the traffic modeling process, there has been little research on how to retain enough access capacity in the centroid. To address this problem, there are two common solutions:

The first is to preserve a high fidelity to the network and include as many local roads as possible to ensure that we are not cutting capacity and causing artificial bottlenecks. This method is not recommended because the scale of the modeling problem increases along with the increase in number of links. The redundant links in the network may pose an extra challenge to the simulation and calibration, especially some have very low demand.

The second is a more systematic solution used by most cutting-edge traffic simulators. It is to employ centroid (loader) connectors. A centroid (loader) connector is the representation of the realistic access to and from the centroid within a TAZ. These connectors are usually given a high capacity value to maintain enough access capacity to the centroid. Placement of centroid connectors is dependent upon



(a) The network with only one connector

(b) The network with sufficient connectors

Figure 3-16: Compare of the network with-and-without connector

the size and shape of the TAZs and how traffic should be loaded onto the network. A summary of guidelines of how to code centroid connectors are discussed in (Cambridge Systematics, Inc., 2007).

The second method was employed in the Beijing case study. Our first step is to move the centroid to the center of this zone. Second, we added centroid connectors, and gave them sufficient capacity. Because maintaining enough access capacity in the centroids is the only concern in this case, the process of adding centroid connectors was quite arbitrary in terms of how many and where to add them. If the accessibility of the centroid or TAZ to the surrounding freeway roads, bus stops, or local business has to be taken into consideration, then the addition of centroid connectors should follow the guideline (Cambridge Systematics, Inc., 2007). Figure 3-16 shows the comparison between the original problematic network with the revised network.

3.4.2 Result

After the network was revised by adding enough access capacity to the centroid, the congestion originating in this zone was relieved. The density change on the jammed freeway link and off-ramp are illustrated in Figure 3-17 and Figure 3-18,

which shows the congestion caused by the artificial bottleneck relieved after the model enhancement.

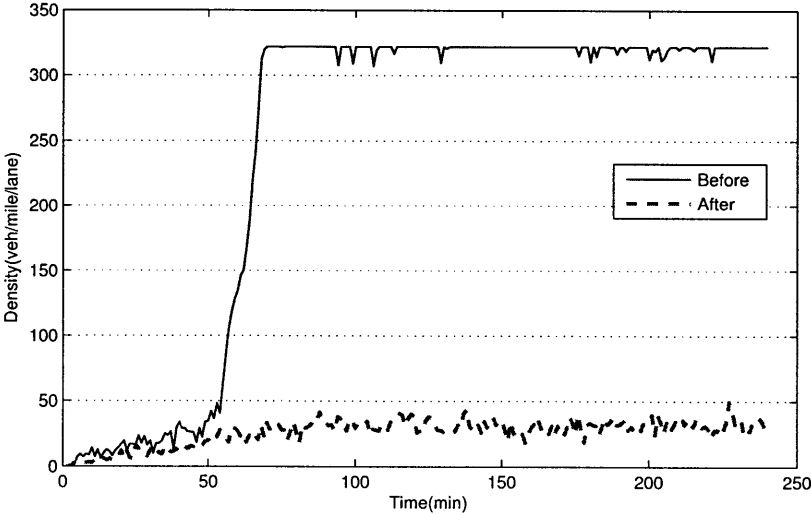


Figure 3-17: Density change on link 419 (ring road freeway)

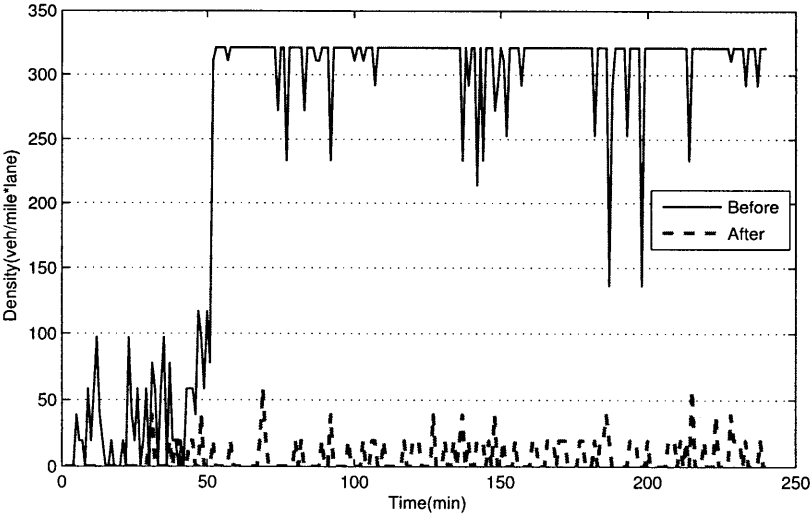


Figure 3-18: Density change on link 420 (off ramp)

3.5 Variable Output Capacity

In urban networks (as often seen in developing countries), the mixed traffic condition is serious. The impact of non-motorized transport modes such as bicycles and pedestrians on motorized vehicle travelling in the network, especially at the intersections, cannot be ignored (See Figure 3-19 a typical scene in urban cities in China). To model this impact, we introduced the "variable output capacity" in the DTA model. As briefly described in section 3.2, the output capacity is a parameter for each lane group (or segment) in the supply model. It is typically calibrated and validated off-line in advance. In most existing studies, the output capacity is fixed during the simulation period, i.e., C_{out}^{jt} in Equation 3.8 or Equation 3.10 is constant across all possible time-step t . We refer to it as "static capacity".

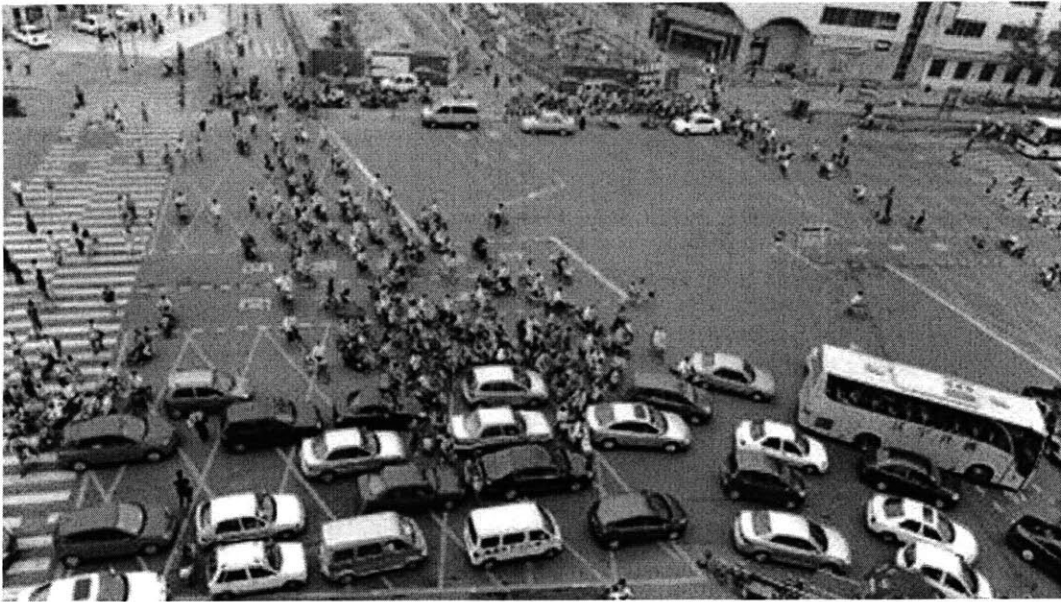


Figure 3-19: Severe interferences from non-motorized

Static capacity does not fully reflect traffic situations with significant interferences from bicycles and pedestrians, which may cause capacity reductions for motorized traffic at intersections, especially during rush hours. Since most bicycles and pedestrian trips are for commuting purposes, their flows are also time-dependent. Therefore, the conflicts between bicycles/pedestrians and vehicles are different during different times of the day. For example, during peak hours, there are many more

bicyclists and pedestrians. Thus, the capacities of the lanes at the intersections drop significantly. To capture such time-dependent capacity reductions, we drop the static assumption on C_{out}^{jt} and make it a time-dependent variable. The initial idea and development of a dynamic output capacity model was done in Xu (2009b). In a model with dynamic output capacity, for each segment, the output capacity may assume different values at different time-of-day. Those values are treated as parameters to the supply model, and can be calibrated during the off-line calibration process.

The variable output capacity feature is important for urban traffic networks that are distinguished by mixed transport mode and on which the motor vehicles are badly influenced by the non-motorized travelers. This feature has been implemented in our DTA model. However, the efficient calibration of these capacities is still a problem to be solved. Thus the usefulness of this feature has not been validated and is a direction for future research.

Chapter 4

Off-line Model Calibration

4.1 Need for DTA Calibration

In order to make effective traffic management decisions, traffic managers need to continuously know the current state of the network and the predicted future states. The better the knowledge, the higher the likelihood of effective decisions. However, in spite of the vast improvements in traffic sensors, it is impossible to measure each and every variable related to the state of the system at every point in time. While sensor and historical data of a network can provide useful information, managers need to be able to estimate and predict useful traffic performance indices which cannot be measured directly.

DTA models provide a useful way to model the transportation system. The effectiveness of DTA models depends on the ability to replicate network conditions. Various inputs and model parameters within the DTA system need to be set to appropriate values so as to replicate the real world accurately. These parameters are usually unknown and difficult to observe directly, and thus should be estimated. The vector containing these parameters to be calibrated is usually called the “Decision Vector” during the calibration.

Calibration is the process of estimating and assigning values to model parameters so as to replicate the traffic measurements closely. Off-line calibration obtains estimates of the parameters of interests using archived data that contain available

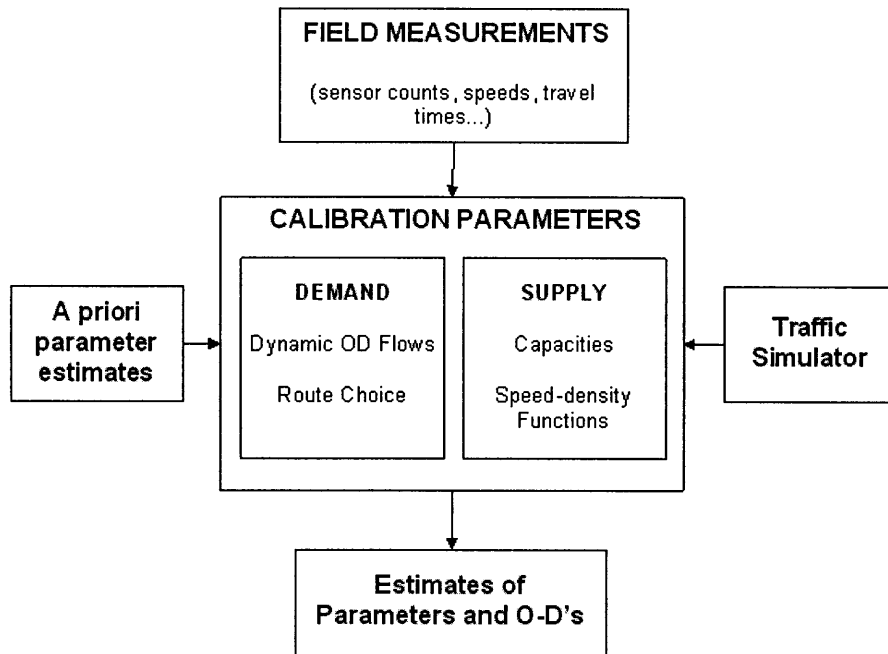


Figure 4-1: Off-line calibration framework (Source: Balakrishna R. , 2006)

system measurements, provides state estimation and updates simulator parameters and inputs for on-line application and prediction. This thesis focuses only on the state estimation of DTA systems and only off-line calibration was employed.

The off-line calibration of a DTA model is summarized in Figure 4-1. A general off-line calibration problem involves the estimation of OD flows as well as various model parameters using a variety of data from sensor measurements and a priori values of those to be estimated. We group the set of unknown DTA model parameters into demand- and supply-side variables. Demand variables are typically common to all DTA models and include time-dependent OD flows as well as travel behavior model parameters. For the supply side, mesoscopic models capture network performance through aggregate traffic flow models involving parameters such as segment/lane group capacities and those for speed-density relationships in Equation 3.5. The calibration is generally carried out in an iterative manner. At each iteration, the traffic simulator will be called upon by the calibration process one or more times depending on the different algorithms used. The optimization algorithm plays a central

role in this process as a way compare the simulation output with field observations in order to evaluate the performance of the updated “decision vector” in the current iteration and to find the search direction for the next iteration.

4.2 Calibration Methodology

The calibration problem can be described as an optimization problem with the objective of minimizing the goodness-of-fit measure between the observed and fitted measurement values. In the following section, an overview of an aggregate calibration methodology (Balakrishna, 2006; Balakrishna et al., 2006) is presented. Aggregate calibration of simulation models has received significant attention in recent years because it possesses several advantages over other existing methods. The first advantage of an aggregate calibration algorithm is that it is capable of handling different types of model parameters at the same time and jointly adjust them simultaneously. This feature is attractive especially when there is a large-scale calibration problem and the decision vector contains parameters in different orders of magnitude. Compared with conventional calibration methodologies, which need to iterate between various parameter subsets to calibrate them separately, the benefit of this feature is huge saving in computational resources. Second, aggregate calibration also introduces the flexibility to incorporate any general traffic measurement, such as sensor counts, speed, point-to-point travel time, etc. More sophisticated data such as point-to-point counts and travel times from automatic vehicle identification (AVI) systems and probe vehicles may also be accommodated.

4.3 Problem Formulation and Characteristics

Let the time period of interest be divided into intervals $h = 1, 2 \dots H$. Let x_h denote the vector of OD flows departing from their respective origins during time interval h . Let β_h be the vector for simulation model parameters that will be calibrated together with the OD flows. The decision vector is $\theta = \langle \mathbf{x}_1, \dots, \mathbf{x}_H, \beta_1, \dots, \beta_H \rangle$. The calibra-

tion problem may then be formulated mathematically in the following optimization framework:

$$\text{Minimize } z(\theta) = \sum_{h=1}^H [z_1(\mathbf{M}_h, M_h) + z_2(\mathbf{x}_h, \mathbf{x}_h^a) + z_3(\beta_h, \beta_h^a)] \quad (4.1)$$

Subject to the following constraints:

$$\left. \begin{aligned} M_h &= f(\theta, G_1, \dots, G_h) \\ l_h^x &\leq \mathbf{x}_h \leq u_h^x \\ l_h^\beta &\leq \beta_h \leq u_h^\beta \end{aligned} \right\} \forall h \in \{1, 2, \dots, H\} \quad (4.2)$$

where \mathbf{M}_h and M_h are the observed and fitted measurements (from both sensors and floating cars) for interval h , respectively. \mathbf{x}_h^a and β_h^a are a priori values corresponding to x_h and β_h ; z_1 , z_2 and z_3 are functions measuring the distances between observed and fitted values or calibrated and a priori values. $f()$, the simulation model, describes the simulated measurements as a function of the OD flows, the network G_h and model parameters β_h up to interval h .

Equations 4.1 and 4.2 together represent a complex non-linear non-analytical optimization problem, owing to the use of a sophisticated simulator to obtain the fitted measurements. The high degree of non-linearity introduces an objective function with potentially many local optima. The local minimum closest to the starting solution may thus be far from a global optimum. The non-analytical nature is attributed to the lack of an explicit form for $f()$ as a function of the calibration variables. Consequently, classical algorithms that rely on the knowledge of exact analytical gradients are not suitable for calibration. Methods that work directly with function values are more appropriate. Since the simulator is often stochastic, the optimization problem must also account for the inherent noises in model outputs. Sophisticated DTA systems contain several sources of stochasticity, such as simulated drivers' behavioral characteristics and the order for processing the vehicles and network elements. Furthermore, the problem is large in scale. The number of OD pairs and time intervals increase rapidly with the size of the network and the desired temporal modeling resolution.

Consequently, appropriate solution algorithms must be able to:

- Perform a global search by overcoming local minima and maxima;
- Work without analytical derivatives (which would generally be unavailable);
- Converge in a reasonable time frame that does not grow rapidly with problem size.

4.4 Solution Algorithm

Previous studies (e.g. Balakrishna, 2006) indicate that the *Simultaneous Perturbation Stochastic Approximation (SPSA)* algorithm developed by Spall (1999, 1998) may provide a possible solution to the problem with the above mentioned characteristics.

4.4.1 Algorithm Feature

SPSA has several features that make it attractive for many practical applications:

- 1) SPSA uses model outputs to directly capture complex relationships between data and model parameters. The approximation of the gradient directly relies on the measurement of the objective function.
- 2) SPSA is designed for stochastic problems and allows for input corrupted by noise.
- 3) The SPSA algorithm is appropriate for large-scale problems because of its efficient gradient approximation by perturbing all variables at once. Compared with traditional gradient approaches, SPSA is more efficient in terms of computational running time.

All optimization problems can be reduced to the basic problem of finding a root θ^* to the gradient of the objective function: $g(\theta) = dz(\theta)/d\theta = 0$, where $z(\theta)$ is a differentiable objective function and θ is the vector of the decision variables. In a

standard optimization algorithm for a multivariate problem of a non-differentiable, noisy objective function, at iteration $k + 1$, the estimate for the decision vector θ_{k+1} would be calculated as follows:

$$\theta_{k+1} = \theta_k - a_k g(\theta_k) \quad (4.3)$$

where θ_k is the k^{th} iteration for the decision vector θ , $g(\theta_k)$ the vector of gradient estimates computed at θ_k , and a_k is the gain sequence vector for the k^{th} iteration in the direction of a reducing gradient (satisfying some conditions for the stable convergence of the algorithm). The SPSA algorithm requires only two computations of the objective function at a given iteration to obtain the estimate of the gradient vector, irrespective of the dimensional size of the decision vector. This approximate estimate is obtained by simultaneous perturbations of all the parameters in the decision vector. A random perturbation vector $\Delta_k = \{\Delta_{k1}, \Delta_{k2}, \Delta_{k3}, \dots, \Delta_{kn}\}$ is generated based on an appropriate random variable distribution (i.e. Bernoulli with 1 and -1 in this case study), and using an appropriate step size vector $c_k = \{c_{k1}, c_{k2}, c_{k3}, \dots, c_{kn}\}$. The partial gradient estimate with respect to the i^{th} decision variable $g_i(\theta_k)$ is then computed for iteration k as follows:

$$g_i(\theta_k) = \frac{z(\theta_k + c_k \otimes \Delta_k) - z(\theta_k - c_k \otimes \Delta_k)}{2c_{ki}\Delta_{ki}} \quad (4.4)$$

where $c_k \otimes \Delta_k$ is the component-by-component multiplication of vectors c_k and Δ_k .

The gradient vector estimate at iteration k is then

$$g(\theta_k) = \{g_1(\theta_k), g_2(\theta_k), g_3(\theta_k), \dots, g_n(\theta_k)\}.$$

4.4.2 Choice of the Gain Sequences

For a given iteration k , the gain sequences are computed as follows:

$$\begin{aligned} a_k &= \frac{a}{(A + k + 1)^\alpha} \\ c_{ki} &= \frac{c_i}{(k + 1)^\gamma} \end{aligned} \quad (4.5)$$

where the values of constants a , c_i , A , α and γ are fixed based on standard guidelines.

The choice of the gain sequences is critical to the performance of the SPSA algorithm. There are some general guidelines available for the choice of values for these parameters in Subsection IIIB of Spall (1998). See also Subsection 5.3.5 of Balakrishna (2006). Practically effective and theoretically valid values from Spall (1998) are 0.602 and 0.101 respectively. The value of the "stability constant" $A = 50$ was also found to work well (Subsections 3.7.2 of Balakrishna 2006).

For a noisy setting where the error in the objective function measurement is high, it is recommended that one picks a larger c and a smaller a value, since the confidence in the gradient estimates is low. On the contrary, when near perfect measurements of the objective function are available, c can be chosen as a relatively smaller positive number. All these values could be adjusted to suit the application at hand, e.g., the value of the constant c_i could depend on the scale of the starting value of the i^{th} decision variable.

4.4.3 Algorithm Process Description

The workflow of SPSA algorithm is illustrated in Figure 4-2. The optimization is carried out as follows:

- 1) Initialize parameters a , A , c , α and β ;
- 2) Set initial decision vector $\theta = \theta_0$ at step $i = 0$;
- 3) Calculate the initial objective function value z_0 based on θ_0 by DynaMIT-P, and set initial objective function value $z = z_0$;
- 4) Update step $i = i + 1$;
- 5) Evaluate $z(\theta_k + c_k \otimes \Delta_k)$ and $z(\theta_k - c_k \otimes \Delta_k)$ in Equation 4.4 by using DynaMIT-P;
- 6) Calculate the current gradient vector by Equation 4.4.

- 7) Update decision vector θ by Equation 4.3, constrained by the upper and lower bound;
- 8) Update z by DynaMIT-P based on current decision vector θ ;
- 9) If the termination criteria is satisfied, stop; otherwise, go back to step 4. The termination requirements in this case are either convergence, meaning the objective function value is flat after several iterations, or has reached maximum running iterations.
- 10) The decision vector θ are set by the current vector when the termination criteria are satisfied.

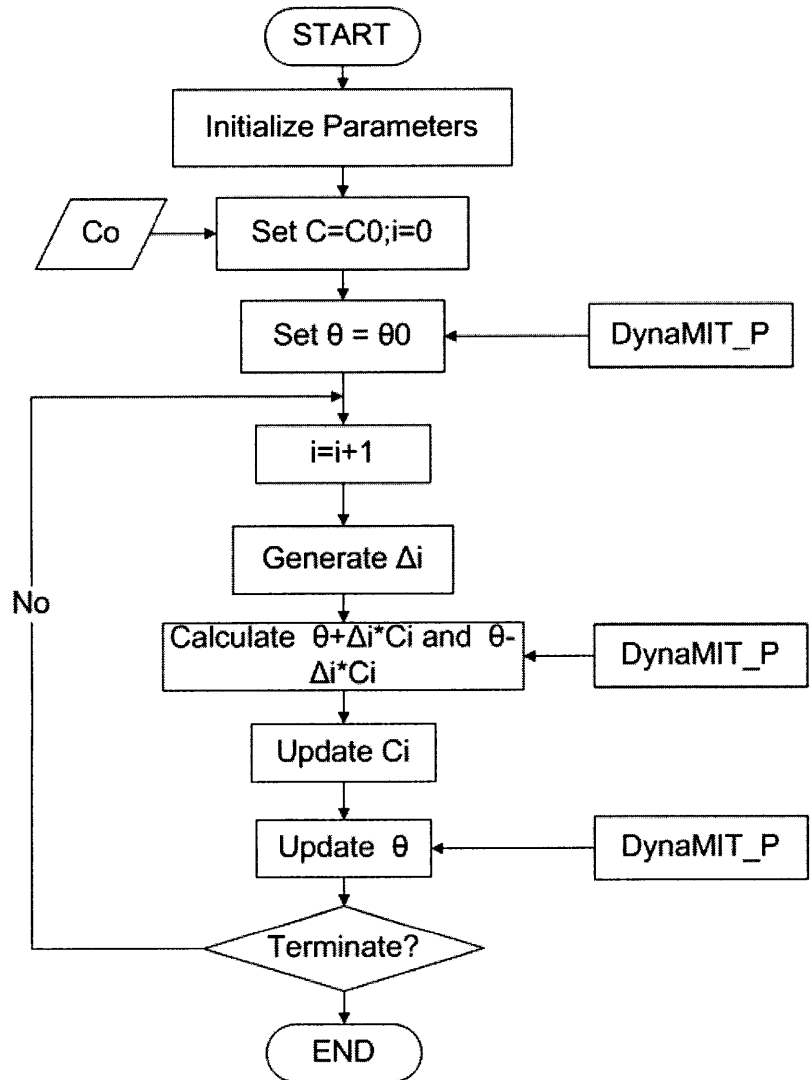


Figure 4-2: SPSA Algorithm Workflow

Chapter 5

Case Study : City of Beijing

5.1 Introduction to DynaMIT-P

In this case study, we use DynaMIT-P (Dynamic network assignment for the Management of Information to Travelers) DTA system to model a large-scale urban network in Beijing. DynaMIT-P is a simulation-based DTA system (Ben-Akiva et al., 1997, 2001a, 2002a) for planning applications. It uses a built-in microscopic demand simulator, a mesoscopic supply simulator and a learning model to capture complex demand-supply interactions. It can predict the day-to-day evolution of travel demand and network conditions, and within-day traffic patterns. It models travelers' short-term and within-day decisions, such as choices related to trip frequency, destination, departure time, mode, and route, assuming that long-term travel decisions (such as residential locations and auto-ownership) are known.

Details about the features and framework of DynaMIT-P can be found in Appendix A of Balakrishna (2006). The travel decisions are modeled in the discrete choice framework (Ben-Akiva and Lerman, 1985), where the aggregate origin-destination (OD) flows are converted into individual vehicles through DynaMIT-P's demand simulator. The packets are then simulated in the mesoscopic supply simulator to obtain the performance of the network by measuring time-dependent flows, travel times, and queue lengths. By adopting the mesoscopic simulation approach in DynaMIT-P and leveraging the efficient algorithmic design and implementation, we were able to sig-

nificantly shorten the running time for the simulation of the network in comparison to typical microscopic traffic simulators.

In previous studies, DynaMIT-P and its corresponding real-time version have been applied successfully in major cities in the United States. In Los Angeles, California, a real-time version was calibrated and deployed as a route guidance system in the South Park area for traffic state estimation and prediction (Wen et al., 2008). In Lower Westchester County, New York, DynaMIT-P was combined with NYSDOT's ITS infrastructure for traffic condition improvements (Rathi et al., 2008). In Boston, Massachusetts, DynaMIT-P was used for the evaluation of emergency evacuation plans (Balakrishna et al., 2008). However, these networks can hardly be labeled as highly congested urban networks as illustrated in the introduction.

5.2 Network and Data

Beijing, China is one of the 10 most populated mega cities in the world. In recent years, the vehicle volume has increased at an annual rate of 20 percent. In 2009, there were reportedly 4 million registered motor vehicles, of which 2.3 million were private passenger cars. Urban trips within the Sixth Ring Road, the outermost ring road, reached 35 million trips per day (including 8.8 million walking trips)(Sun, 2009). The significant pressure on the transportation system results in severe traffic congestion and air pollution. Figure 5-1 shows the link volume-over-capacity (V/C) ratios during morning peak hours on weekdays in 2007 from a static transportation planning package, where red roughly indicates a level of service D (Highway Capacity Manual, 2000) or worse.

The skeleton of the Beijing urban transportation network comprises of a series of ring roads connected by arterial roads. The study area is the West 2nd Ring Road network and its northern and southern extensions (Figure 5-2). Several major ring roads and arterial roads intersect the 2nd Ring Road within this area resulting in several interchanges. The ring roads are elevated roadways supplemented by parallel side roads with frequent on- and off-ramps. These ramps are generally spaced between



Figure 5-1: Network V/C ratios of morning peak on a weekday, 2007

200 to 600 feet to ensure access to and exit from the ring roads (Figure 5-3). The study network passes through the center of the city. It is not uncommon for the northern part of the West 2nd Ring Road to be in a complete jam condition extending several miles. The situation is further complicated by the presence of unusually short links - some as short as 20 feet, which could cause unexpected problems for the simulation model. Modeling such conditions is difficult because the congestion is so severe that a small over-estimation in demand or a small under-estimation in supply would result in a gridlock.

The computer representation of this network consists of 1,698 nodes connected by 3,180 directed links (Figure 5-6) in an area of approximately 35 square miles. The historical dataset includes static demand during the AM peak hours for 2,927 non-zero origin-destination (OD) pairs, derived from the most recent household surveys and calibrated against surveillance data. The static demand was processed to derive an initial time-dependent demand for 15-minute intervals. The simulation ran from 6:00am to 10:00am. Approximately 630,000 vehicles were simulated.

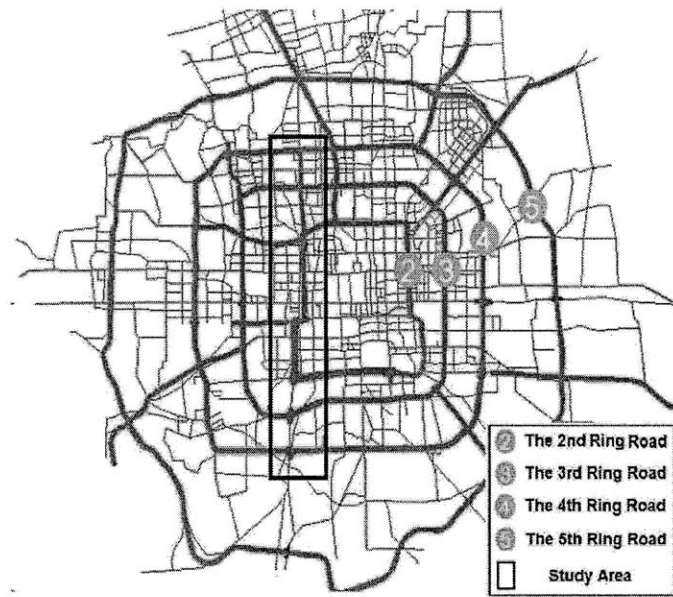


Figure 5-2: Study area (within the black rectangle)

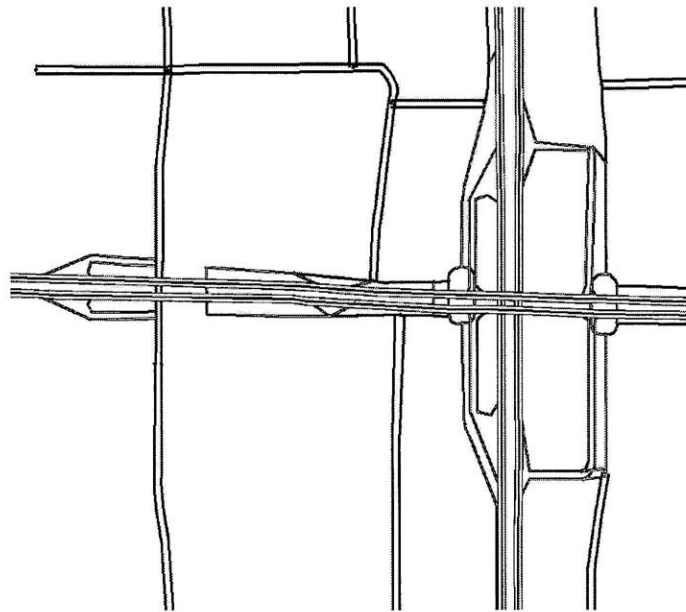


Figure 5-3: Frequent on- and off-ramps

5.2.1 Surveillance Data

Surveillance information used in the Beijing case study includes traffic counts and link travel times from 6 weekdays during December 2007 between the hours of 6:00am and 10:00am.

The traffic counts was obtained from Remote Traffic Microwave Sensors (RTMS) (triangles shown in Figure 5-6). The RTMS is a side-looking microwave radar which emits low-power modulation microwaves within the detector area (usually coverings all lanes in a road segment). The radar counts passing vehicles by receiving the reflected microwave. Each RTMS detector reports accumulated traffic counts, speed and occupancy every two minutes.

Within the study area, there are a total of 154 RTMS detectors deployed (Figure 5-4). 140 of them are functioning normally, providing 24-hours traffic flow information continuously. Most of the detectors are located only along freeways. There are no traffic counts on the arterial roads. The link travel times are extracted from *Floating Car Data* (FCD). The FCD was obtained from GPS (Global Positioning System) equipped taxi fleets. Figure 5-5 shows the FCD coverage in the whole city. Nearly 90% of all the major roads in Beijing are covered by the FCD, which can makes up for the lack of count observations on arterials, and thus improves sufficient system observability.

We have no access to the raw data. The observed counts used in our case study are given by 15-minute interval aggregated by the data provider, *Beijing Transportation Research Center* (BTRC). The link travel time data are aggregated to 5-minute, available on freeways, ramps, arterials, secondary roads and local roads. There is data quality limitation regarding the RTMS counts, which will be discussed in Section 5.2.2.

Since DynaMIT-P's data reporting interval is 1 minute, the output counts from DynaMIT-P were aggregated to 15-minute counts in order to be comparable. The 5-minute average travel time observations were repeated 5 times for the corresponding 5 1-minute intervals.

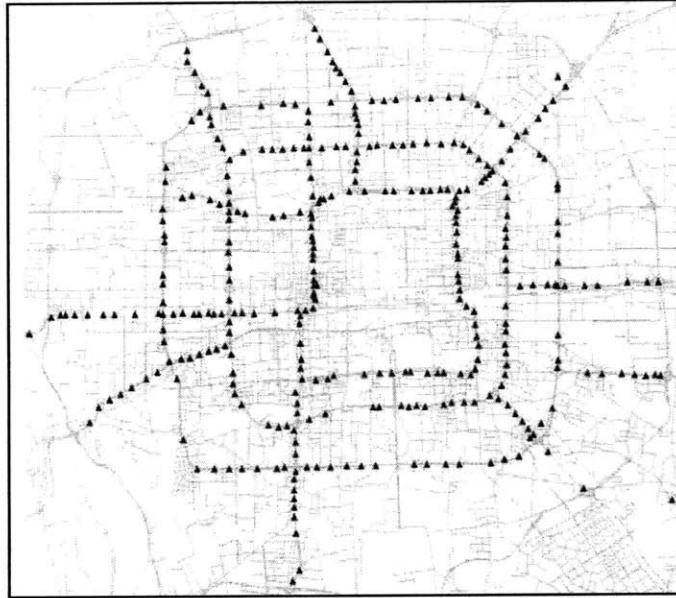


Figure 5-4: Deployment of RTMS detectors



Figure 5-5: Coverage of FCD system

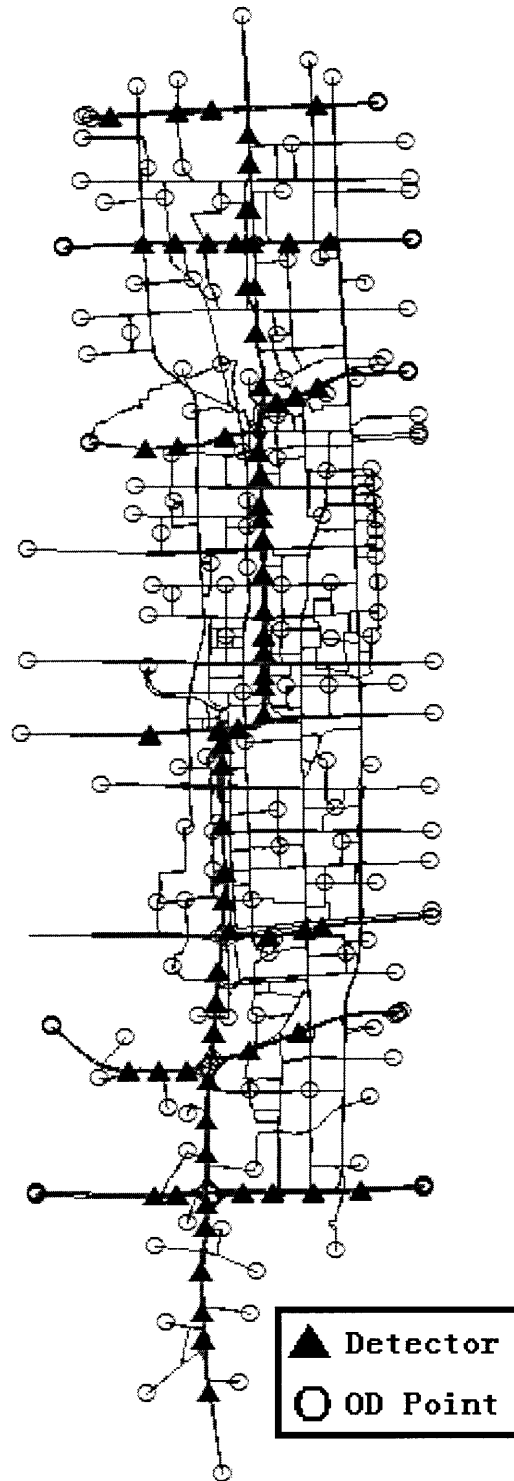


Figure 5-6: Network of the study area

5.2.2 Sensor Data Analysis

Data quality will affect calibration and model efficiency. According to the previous experiences of BTRC, the RTMS detector will count some vehicles more than once during periods of heavy congestion, which is also verified by Klein (1997). While we can do little to address this because of the inherent hardware limitations of RTMS detectors, the problem should be recognized when the RTMS data from congested networks are employed.

We conducted some preliminary data cleaning to remove spatial and temporal inconsistencies in the sensor data. For example, Figure 5-7 shows the locations of the RTMS detectors in a specific area of the network. Note that, detector No.117 and detector No.114 are distributed adjacently on a 4-lane freeway, distanced 480 meters from each other. The direction of traffic runs from detector No.117 to No.114. There is an on-ramp merging onto the freeway after detector No.117, before detector No.114, and no exit or off-ramps between these two detectors.

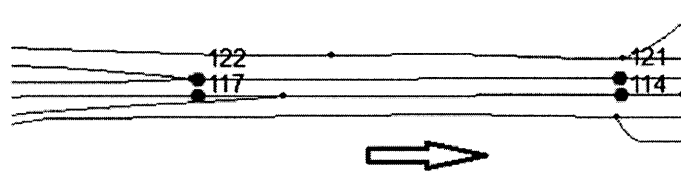


Figure 5-7: Locations of inconsistent RTMS detectors (1)

It is reasonable to believe that the count reported by detector No.117 should be no more than that of detector No.114 in a normal situation. The opposite could be true only when there is an incident somewhere between these two sensors. In this case, traffic flow enters but few vehicles exit. Due to the imbalance of in-and-out flow, this period will last till the inflow fills up the road space from detector No.117 to the incident point. During this period, the counts reported by detector No.117 is likely to be higher than the counts reported by detector No.114. The duration of this period and the maximum total discrepancy of the counts reported by two sensors are

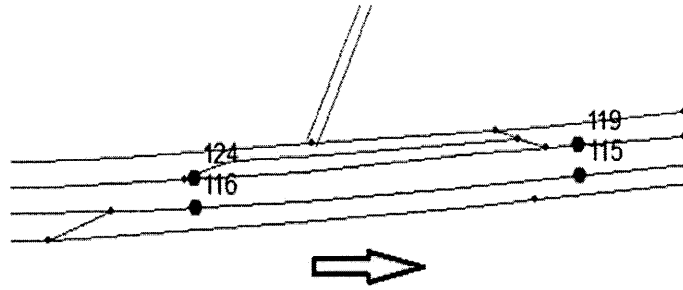


Figure 5-8: Locations of inconsistent RTMS detectors (2)

decided by the distance between these two sensors.

In our case, the distance between the two sensors is 480 meters, with space that can contain a maximum of around 380 vehicles in 4-lanes, assuming the average length of a vehicle is 5 meters. Therefore, even if the above case occurred, the maximum counts discrepancy between these two sensors during the whole incident period is 380.

However, the observed counts of these two detectors (Table 5.1) show huge discrepancies. For instance, during the interval from 8:00AM to 8:15AM, detector No.117 reported 1367 counts, but detector No.114 only reported 174 counts. During the interval from 8:45AM to 9:00AM, detector No.117 reported 1236 while No.114 reported 175.

The difference is obviously unrealistic. This problem is typical in multiple locations within the study area. (See Figure 5-8 for another case). Therefore, data cleaning and validation is required before a good calibration can be obtained.

After manually checking and screening the available sensor data to ensure spatial and temporal consistency across the network, a subset of 133 sensors was chosen for use in the remainder of this case study.

Table 5.1: Inconsistent Observed Counts from RTMS detectors

Sensor ID	117	114	116	115
6:00-6:15	247.7667	91.11041	170.7815	357.5216
6:15-6:30	289.1792	93.4703	236.28527	475.0912
6:30-6:45	433.4181	110.7799	303.4776	634.674
6:45-7:00	602.3742	115.4303	393.4808	798.6409
7:00-7:15	866.7772	113.1364	482.961	1004.683
7:15-7:30	1191.263	173.4423	666.7613	1321.522
7:30-7:45	1335.821	165.4067	683.8692	1452.826
7:45-8:00	1328.751	192.2045	777.8875	1528.467
8:00-8:15	1367.312	174.278	721.2708	1525.521
8:15-8:30	1217.863	171.3316	766.5458	1473.429
8:30-8:45	1303.629	196.1971	755.45	1520.602
8:45-9:00	1236.958	175.46	828.7547	1543.076
9:00-9:15	1296.626	216.899	777.2801	1513.777
9:15-9:30	1237.589	239.077	842.8722	1585.565
9:30-9:45	1369.111	262.9876	806.5917	1597.372
9:45-10:00	1250.729	238.7888	809.5639	1647.568

5.3 Calibration

5.3.1 Calibration Variables

Calibration variables include those from both the demand and supply modules of DynaMIT-P. Specifically, these include 46,832 time-dependent OD trips from 2,927 OD pairs and 16 time periods; one coefficient of travel time β_{TT} in the route choice model $V_i = \beta_{TT}TT_i$, where V_i is the systematic utility of path i and TT_i is the travel time on path i ; 19,080 segment speed-density parameters (6 parameters, v_{max} , k_{min} , k_{jam} , α , β and v_{min} for each segment); and 3,180 segment output capacities on ring roads and arterials. The total number of calibration variables is 69,093.

5.3.2 Methodology

All the demand and supply parameters of DynaMIT-P were calibrated simultaneously using SPSA the methodology discussed in Chapter 4. For this specific problem, different weights were given to the functions that measure the distances between observed and fitted values or calibrated and a priori values. The minimization problem

in Equation 4.1 was rewritten here as Equation 5.1.

$$\underset{x_1, \dots, x_H, \beta_1, \dots, \beta_H}{\text{minimize}} \sum_{h=1}^H \left[w_1 \|\mathbf{M}_h^c - M_h^c\|^2 + w_2 \|\mathbf{M}_h^t - M_h^t\|^2 + w_3 \|x_h - x_h^a\|^2 + w_4 \|\beta_h - \beta_h^a\|^2 \right] \quad (5.1)$$

where \mathbf{M}_h^c and M_h^c are the observed and fitted counts for interval h respectively, and \mathbf{M}_h^t and M_h^t are the observed and fitted link travel times for interval h respectively.

The objective function is a weighted sum of distances between time-dependent location specific simulated measurements and field measurements (both counts and link travel times) and distances between calibrated variable values and their respective a priori values.

The weights w_1, \dots, w_4 depend on the relative confidence one can put on the corresponding measurements and a priori values. For example, if sensors are not reliable, a lower weight might be put on counts. The weights also depend on the order of magnitude of the measurement in order to avoid a situation where a parameter with a bigger magnitude or more observations dominates the others in the fitting function.

Initially, we assigned a weight of 1 (w_1) to the sensor count measurements, 0.05 (w_2) to the floating car travel times due to the high volume of observations, and 1 (w_3, w_4) to the a priori values. These weights were changed during the calibration in response to the performance of the SPSA iterations to accelerate the optimization process. Because the a priori values are just a guess for which we do not have much confidence, w_3, w_4 the weights for the a priori values were adjusted to 0.

The optimization problem is solved using the *Simultaneous Perturbation Stochastic Approximation* (SPSA) algorithm.

5.3.3 Results

The quantification of error in model performance is important for the evaluation of the calibration. After the model is calibrated, validation tests are carried out. The fit

to counts and travel times are computed across all reliable sensors and all available floating car observations. The following two error statistics have been adopted to measure the discrepancies between observed (y_i) and simulated (\hat{y}_i) quantities, where S is the dimension of the unknown vector:

- Root mean square error (RMSE) (Pindyck and Rubinfeld, 1997)

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^S (y_i - \hat{y}_i)^2}{S}} \quad (5.2)$$

- Normalized root mean square error (RMSN) (Ashok and Ben-Akiva, 2002) Toledo and Koutsopoulos, 2004)

$$\text{RMSN} = \frac{\text{RMSE}}{(\sum_{i=1}^S y_i)/S} \quad (5.3)$$

The first 30 minutes of the study period (6:00am-10:00am) is used to warm up and load the network. Thus the focus of the calibration and evaluation is limited to 6:30am to 10:00am.

A lower value of RMSE or RMSN indicates a lower discrepancy between the simulation results and the observations. The calibration starting point in terms of error statistics is given in Table 5.2:

Table 5.2: Starting values of model fit

	No. of Observations	Observed Average	Simulated Average	RMSE	RMSN
Counts (Veh/15min)	1722	1236.07	424.38	968.23	0.8073
Link Travel Time (s)	52545	39.67	57.431	165.64	4.1567

The gain sequences and objective function weights were adjusted during the calibration to accelerate the process. The objective function value also usually changes significantly right after an adjustment. Figure 5-9 shows the last 530 iterations of SPSA, where the weights and gain sequences are kept constant. An improvement was found, which indicates that the SPSA algorithm is appropriate and efficient for this particular large scale problem.

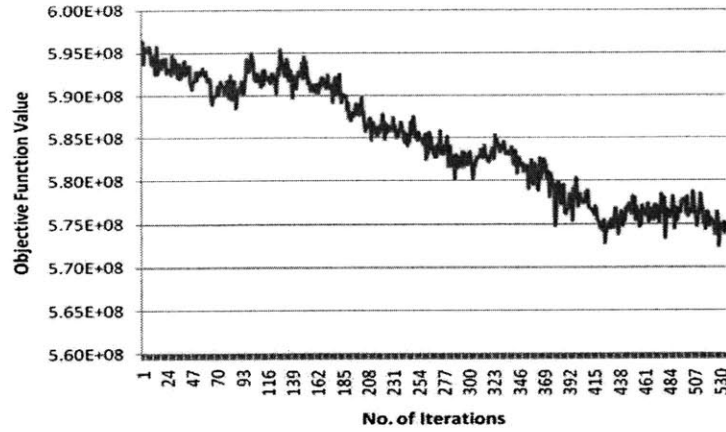


Figure 5-9: Trend of objective value for 530 iterations

Table 5.3 contains the error statistics on the fit to counts and fit to link travel time across all time horizons.

Table 5.3: Overall calibration results

	No. of Observations	Observed Average	Simulated Average	RMSE	RMSN
Counts (Veh/15min)	1722	1236.07	1153.20	427.00	0.345
Link Travel Time (s)	52545	39.67	37.42	18.40	0.464

The comparisons between simulated and observed counts by time horizon are given in Table 5.4. An intuitive illustration is given in Figure 5-10 for the fit-to-counts during the peak periods between 8:30 to 9:00AM. The x -axis shows the observed sensor counts and the y -axis is the calibrated sensor counts. The 45-degree line indicates a perfect match between the simulated counts and the observed counts. The sensors with counts deviating more than 50% from the observed values were marked with sensor numbers. Most of the observed and simulated sensor counts fall around the 45 degree line, indicating that most of the deviations between the simulated and observed counts are within an acceptable range. The complete calibration results for all time periods are presented in the Appendix B.

Figure 5-11 shows the RMSN and RMSE of counts at different flow rates. The first group gives the overall calibration result, and the remaining ones are by flow rate levels: high (>1400 veh/15min), medium (1000-1400 veh/15min), and low (0-

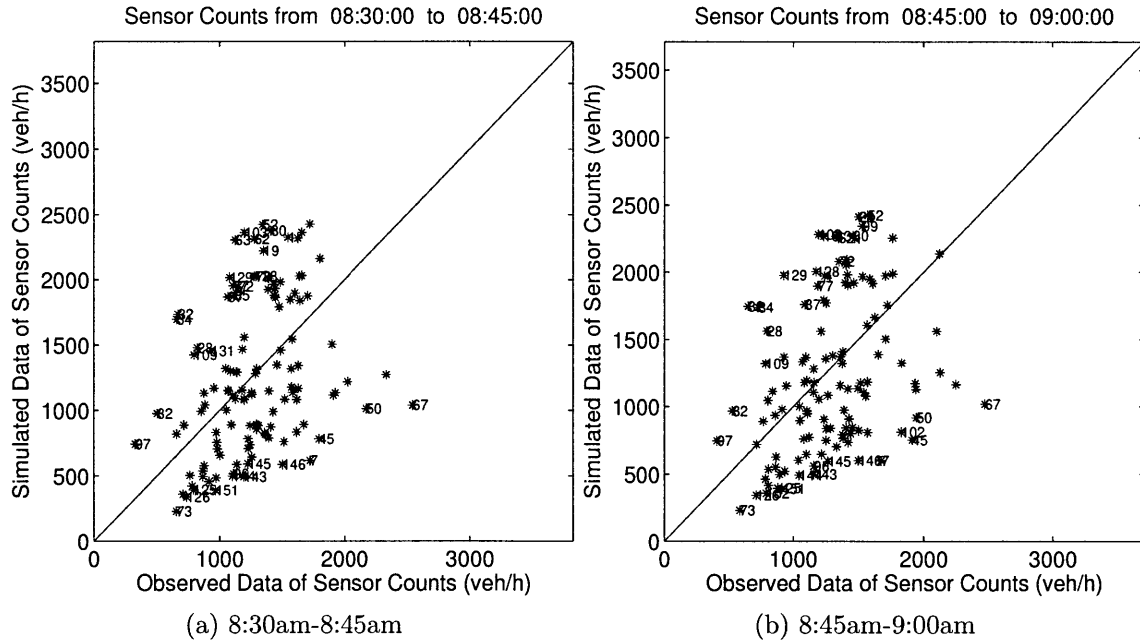


Figure 5-10: Calibration results for the peak periods

1000 veh/15min). The best fit to count is in the flow rate between 1000-1400 vehicles/15min, in which the RMSN is around 28%.

A similar analysis on floating car travel time was conducted and is shown in Figure 5-12. The travel times are grouped into 4 categories: 0-20s, 20-40s, 40-60s and longer than 60s. The best fit for travel time is reached in the category with link travel times between 40-60s, with a 33.8% RMSN.

An in-depth analysis was also performed on the path travel time in order to better evaluate the model capabilities for replicating realistic traffic situations. The floating car data are given in the form of link by link travel time, and not every link in an OD path has floating car data available. In order to carry out path travel time analysis, we have to check the data availability for any given set of paths for an OD pairs. In our case, the path set for each OD pairs was generated by DynaMIT-P. Among them, 313 paths, with floating car data available were chosen for analysis. The comparison between simulated and observed path travel time is given in Figure 5-13. Most of the red asterisks are around the 45 degree line, indicating a good match between the simulated path travel time and the observed data.

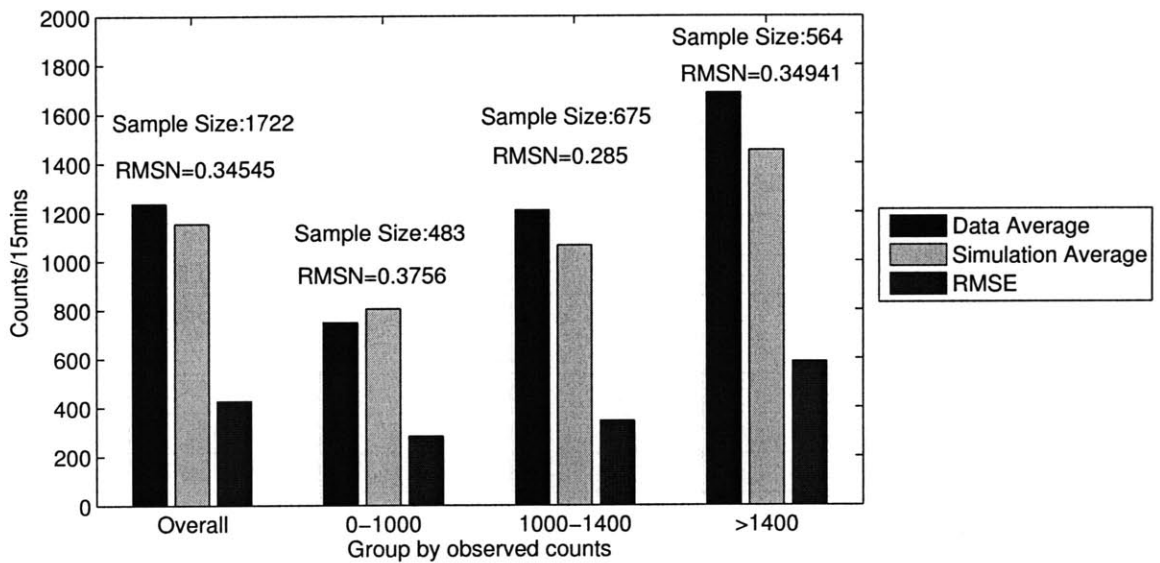


Figure 5-11: Fit to counts analysis

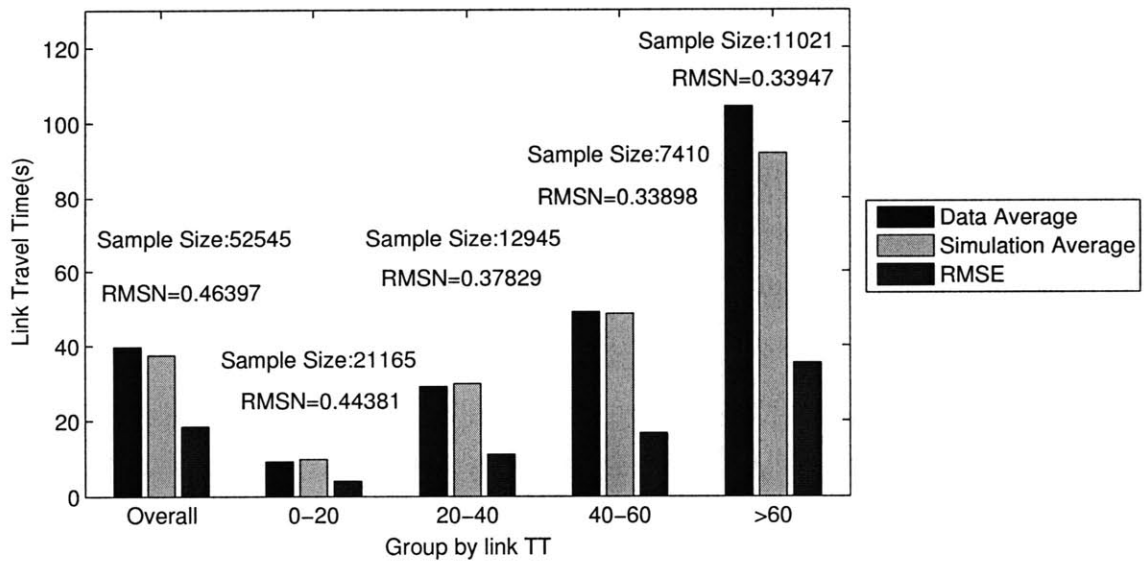


Figure 5-12: Fit to link travel time analysis

Table 5.4: Fit-to-counts by time interval

Time Interval	Observed Average	Simulated Average	RMSE	RMSN
6:30-6:45	889.94	756.85	413.67	46.48%
6:45-7:00	1064.16	965.29	398.24	37.42%
7:00-7:15	1165.74	1111.72	399.69	34.29%
7:15-7:30	1231.92	1186.87	423.23	34.36%
7:30-7:45	1274.23	1203.71	448.35	35.19%
7:45-8:00	1281.68	1199.35	456.27	35.60%
8:00-8:15	1263.04	1208.94	434.56	34.41%
8:15-8:3	1276.67	1203.15	444.77	34.84%
8:30-8:45	1273.45	1215.68	415.31	32.61%
8:45-9:00	1292.26	1216.91	430.70	33.33%
9:00-9:15	1304.49	1223.17	421.75	32.33%
9:15-9:30	1324.79	1215.67	428.63	32.35%
9:30-9:45	1331.69	1215.35	427.83	32.13%
9:45-10:00	1331.00	1211.09	430.72	32.36%

5.4 Application Analysis

After calibrating our model, we conducted an analysis to evaluate the Rotating No-Driving Day restriction scenario in Beijing. This restriction is a strategy proposed by the Beijing municipal government to reduce pollution and relieve traffic. Implemented as part of a six-month trial that took effect after the 2008 Olympics, it prohibit private cars from being on the roads one weekday per week according to a rotation schedule based on license plate numbers. For example, cars with a license plate number ending with 1 or 6 were not allowed to be on the road on Mondays. Those with plate numbers ending in 2 or 7 were banned from the roads on Tuesdays, and so on. This restriction reduced Beijing's 3.5 millions car network demand significantly.

To model this scenario, we decreased the demand by 20%, assuming that the last digit of the license plate number is randomly and evenly distributed. We then compared the simulation results of different scenarios with a base case with no restrictions. The results are presented as follows.

We focused on the two most congested areas of interest for transportation management: the TianNingSi area (Figure 5-14) and XiZhiMen area (Figure 5-15). The

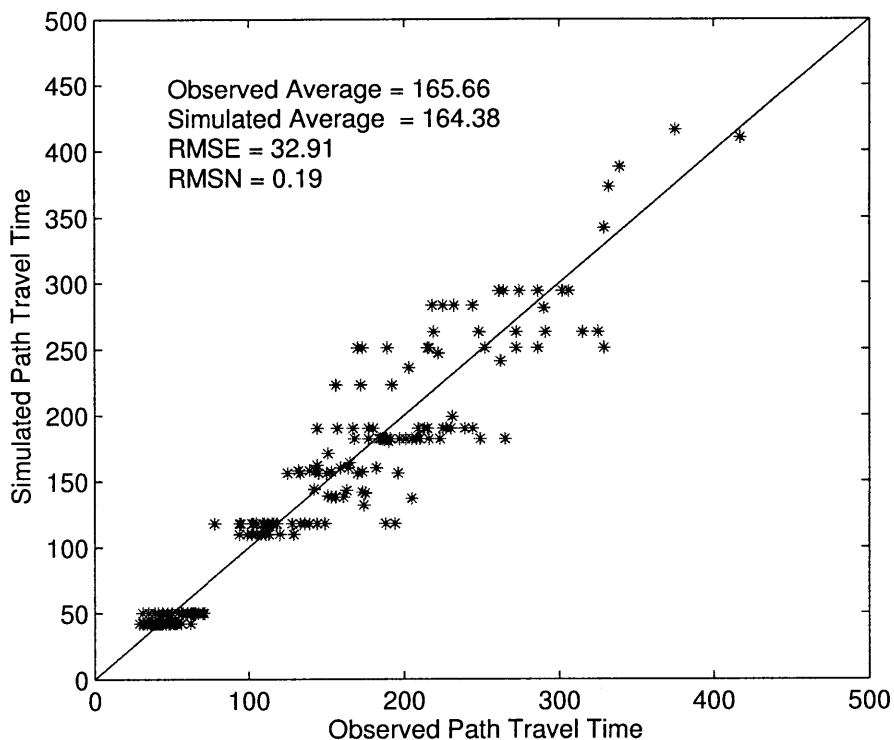


Figure 5-13: Fit to path travel time analysis

pictures are screenshots of the simulation results during the same time period, for different scenarios, in each location. The pictures on the left present the results of the Rotating No-Driving Day restriction. The pictures on the right present the control case without any restriction. As shown in the legend, the color of the segment denotes its density, with red indicating high density and thus severe congestion. It is shown that the reduction of demand resulted in significant drops in traffic densities.

Quantitative analyses were also performed. To analyze the impacts of imposing the Rotating No-Driving Day restriction in terms of alleviating traffic congestion, two criteria are considered: the average travel time and the number of links with long queuing times. The queuing time of a link is defined as the amount of time during which the link has one or more lanes containing queues. In the scenario with restrictions, the traffic demand was reduced by 20%. However, the number of vehicles that reached their destination during the simulation time period (i.e., 550,588, as shown in Table 5.5) was only 8.4% less than the control case. This implies that the

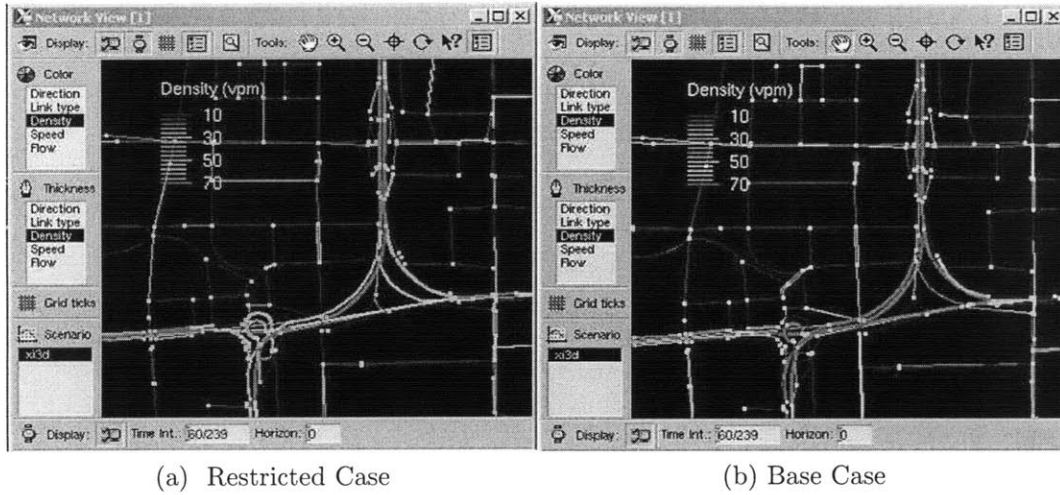
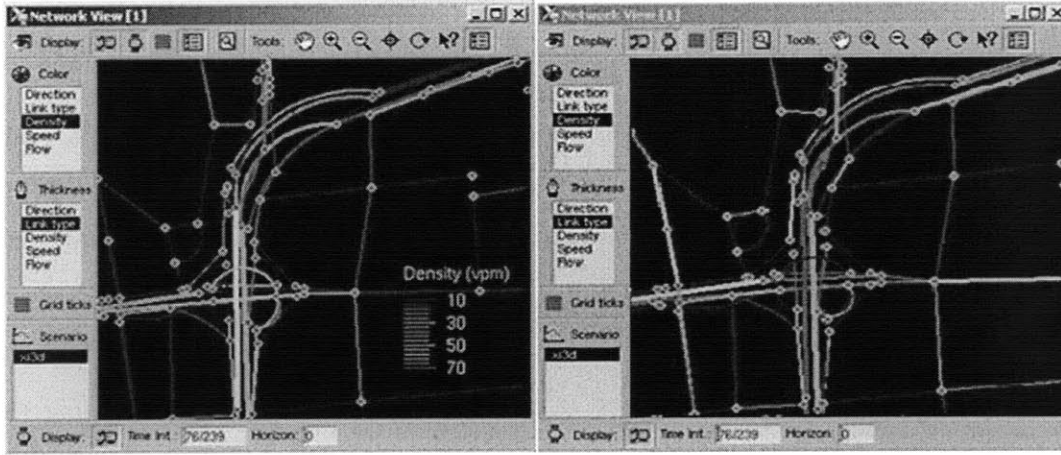


Figure 5-14: Application analysis in TianNingSi Area, 7:00AM

Rotating No-driving Day restriction reduced network demand and increased network efficiency. The average travel time was aggregated from the travel time of every vehicle that reaches its destination within the simulation time period. The average travel time under the restriction scenario is about 2 minutes less (17.9%) than the control scenario. We also calculated the average travel times for some major OD pairs. For example, the average travel time from the south end (Origin Point: 1662) to the north end (Destination Point: 1624) was 2701.1 sec, which is a 20% reduction compared to the base case. In addition, the number of links with queuing times longer than 30 minutes was reduced by 50% under the restriction scenario. Overall, based on the simulation results, imposing such a restriction could significantly increase road efficiency and reduce congestion. This case study demonstrates the enhanced DTA model's ability to evaluate traffic management strategies.



(a) Restricted Case

(b) Base Case

Figure 5-15: Application analysis in XiZhiMen Area, 7:15AM

Table 5.5: Analysis of Rotating No-Driving Day Restriction

	With Restriction	No Restriction
Number of vehicles reaching destinations (veh)	550,588	601,622
Average Travel Time for All OD pairs (s)	497.73	606.89
Average Travel Time for OD pairs from South end to North end	2701.1	3393.5
Number of links with queuing time >30 min	58	112

Chapter 6

Conclusion

6.1 Summary

Despite the increasing demand for traffic management and planning tools as traffic congestion increases in urban cities, there has been little to no research in dealing with highly congested, large-scale urban traffic networks.

This thesis discusses the enhancements of DTA model, in order to have the capabilities of simulating such a network, contributing to its effective planning and management. The enhancements were validated through a case study on a traffic network in the downtown area of Beijing, China. At first, attempts to simulate the studied network resulted in heavy gridlocks that did not exist in the real-world. We were able to identify several model features contributing to the problem, which include 1) a route choice model which can account for overlapping routes, 2) explicit representations of lane groups to properly model traffic queues and spillbacks, 3) the ability to handle short links, 4) maintaining enough loader access capacity to avoid artificial bottleneck and 5) the impacts of bicycles and pedestrians on auto traffic modeled by calibrated dynamic road segment capacities.

Unrealistic congestion on the freeway was resolved. We identified the route choice bias on overlapping segments generated by MNL model. Usually, a large-scale complex network does not abide by the i.i.d assumption required by MNL model because there are a large number of overlapping paths that share the same freeway links.

We revised the route choice model by incorporating PSL model to accounts for the overlapping paths.

We also implemented a lane-group-based queuing feature that captures the specific lane turning restrictions. This stopped spillbacks from one lane-group from affecting traffic in another lane group, and decreased the congestion in the simulation.

Short links, some as short as one car length, caused a major problem in the simulation, as they showed that vehicles moved abnormally slow and non-existent queuing. We increased the minimum speed of vehicles traversing through short links by enhancing the speed-density function to realistically represent the dynamics within a short link. In addition, we treated the queuing problem using a revised capacity model so that the output capacity and acceptance capacity would be able to reflect real-world traffic behavior. The speed-density and capacity functions normally used were not appropriate for short links because of the inherent different between long segments (which are used for transport) and short links (which are connectors between long road segments).

The high user demand problem was associated with the network simplification process. In the simulation, it would be too time and computationally demanding to reflect every single roadway in a network. Therefore, some minor or neighborhood roadways are eliminated. However, this also eliminates network capacity, which combined with the high user demand, caused non-existent traffic jams. By adding centroid connectors, which signified capacity, while still allowing for network simplification, the congestion was eliminated.

A distinguishing factor about Beijing, is the large volume of non-motorized traffic resulting for pedestrians and bicyclists. To simulate this phenomenon, we introduced dynamic output capacity into the DTA model, which was able to capture the time-of-day dependent dynamics of the mixed traffic conditions.

After addressing the main problems that we encountered, the enhanced DTA model was applied to a real world network. The network we used to validate the enhancements was the West 2nd Ring Road in Beijing, China, and its northern and southern extensions. The BTRC provided surveillance data including traffic counts

and FCD from six weekdays in December, 2007, from 6:00-10:00AM. Model was calibrated offline using SPSA algorithm. The total number of calibration variables for this specific case study is 69,093. The effectiveness of the SPSA algorithm was proved to be acceptable for large-scale problem.

Finally, in order to test the newly enhanced model capability of evaluating management strategies for transportation planning, the model was applied with a new traffic reduction policy implemented in Beijing after the 2008 Olympics, which restricted private vehicle usage based according to a weekly rotation plan based on license plate numbers. The simulation results were reflective of the real-world traffic results, indicating that the new DTA enhancements were effective and that DynaMIT-P is now applicable for large-scale, highly congested urban networks.

6.2 Future Research Directions

The realism of the enhanced model and the efficiency of the calibration algorithm have been demonstrated in this thesis. Further research directions are suggested:

- Develop and calibrate a new route choice model using individual trajectory data. As a key component of a DTA model, the route choice model is critical for an accurate prediction of traffic congestion and for the estimation of origin destination flows. In this thesis, the route choice model parameter was calibrated using aggregate data such as sensor counts and floating car travel times. The route choice model can also be estimated based on disaggregate data at the individual level, which provides detailed behavioral information generally unavailable at an aggregate level. The estimated route choice model parameters can serve as a good starting point for the overall aggregate calibration and is expected to improve the calibration quality.
- In DynaMIT-P, traffic congestion is caused by limited capacity, either the explicit exit capacity at the end of a segment or the implicit capacity determined by the speed-density relationship. In the first case, queues will form on the

segment because of insufficient output capacity. In the second case, vehicles traverse on the congested segment at a very low speed but still in the moving part of the segment. The current calibration results show a possible underestimation of the implicit capacity. Future calibration should start with different sets of speed-density relationship parameters that imply higher capacities.

- Although SPSA proved to be suitable for large scale calibration problem in this thesis, the effectiveness of the algorithm in converging the objective function value with the global minima still needs to be improved. Some experiences indicate that the SPSA can effectively converge the objective function value with the local minima rather than the global minima(e.g. a much better solution can be obtained by changing the initial free-flow travel time and k_{min}). It is worthwhile to investigate improvements in SPSA, such as automatic gain selection, "second-order" SPSA, as well as alternative calibration algorithms for high-dimensional stochastic problems.
- Higher quality and additional input data. Input, including enhanced network coding and more accurate surveillance data, are expected to improve model accuracy. Currently, there are only 6 days of surveillance data available in this study, including traffic counts and link travel times. It will be a benefit if more data can be utilized in the model development and calibration process.
- Speeding up the simulation using scalable DTA solutions (Wen, 2009). The running time is a major limitation on the calibration performance. Within a certain time period, the more calibration iterations that could be done, the faster the objective function value converges. Then it will be easier to test new enhancements for the model.

Appendix A

MATLAB Code for SPSA

Algorithm

```
%Matlab code for SPSA calibration. Based on J.C. Spall, Jan. 2000
%Zheng Wei, Dec.2009 Revise Ratio Version

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This code implements SPSA with constraints for theta to lie in
%a specified hypercube (i.e., component-wise constraints). Allows for multiple cases
%for purposes of statistical evaluation based on knowledge of true (noise-free) loss value
%(set cases=1 if user only wants one run).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

clear all

global no_ODs          % # OD pairs
global no_intervals   % # Time intervals
global no_segments    % # Segments (spddsy)
global no_groups       % # Segment groups
global no_other_params % # other parameters
global demand_factor  % scaling factor for demand
global paramsPerDay   % # parameters to be calibrated per day of data
global day
global counter_glob;

counter_glob = 0;
day = 2;

% Set running parameter
```

```

RunType =2;          %0:start from seed file
                    %1:Continue from last calibration iteration
                    %2:Continue from DynaMIT input file

n=30;               %total no. of loss measurements
cases=1;
grad_reps =1;      % no. of reps for averaging the gradient approximation
alpha =.602;
gamma =.101;       %chosen by standard guidelines
a=10^(-10)*(50+1)^0.602;
A=50;

% Pre-set parameters for calibration
no_ODs = 2927;
no_intervals = 16;
no_segments = 3180;
no_groups = 3180;
demand_factor = 4;
no_other_params = 1;
paramsPerDay = no_ODs*no_intervals + no_segments + 6*no_groups + no_other_params;

%Load Weights (Optional)
weights_theta = ones(paramsPerDay,1);

%Generate C
c = 0.01;

% Start at the SPSA directory
if (RunType == 0 || RunType == 1)

    % load input information
    load seed/od_seed.dat;
    load seed/cap_seed.dat;
    load seed/spddsy_seed.dat;
    load seed/param_seed.dat;
    rtchoice_param = 1000*(0-param_seed(1:1));
    theta_0_dynamit = [od_seed; cap_seed; spddsy_seed; rtchoice_param];

    if (RunType ==0)

        theta_0_spsa = ones(paramsPerDay,1);
        p = length(theta_0_spsa); % Set p, the dimension of the parameter vector

        cd DynaMIT
        od = [no_ODs;no_intervals;demand_factor;od_seed];

```

```

cap = [no_segments;cap_seed];

system('rm OD_vector.dat');
system('rm cap_vector.dat');
system('rm spddsy_vector.dat');
system('rm rt_choice.dat');

save OD_vector.dat od -ascii;
save cap_vector.dat cap -ascii;
save spddsy_vector.dat spddsy_seed -ascii;
save rt_choice.dat param_seed -ascii;

cd ..

%Initial Objective Value
initial_fn_value = FUNC(theta_0_dynamit);
tmp_x = theta_0_spsa';
tmp_fn = initial_fn_value;

save Results/fn_theta_0.dat tmp_fn -ascii;

%Initialize
x_values = [];
fn_values = [];
fn_path = [];
fn_path = [fn_path; initial_fn_value];
grad_log = [];
deltas = [];
nstart = 0;

else
%load last time information
load Results/x_final.dat;
load Results/fn_path.dat;
theta_0_spsa = x_final; %
p = length(theta_0_spsa); % Set p, the dimension of the parameter vector
nstart = length(fn_path); % continue from iteration n
counter_glob = (nstart-1)*3+1; %set the global counter

initial_fn_value = fn_path(nstart,1);
tmp_x = theta_0_spsa';
tmp_fn = initial_fn_value;
save ./Results/fn_theta_0.dat tmp_fn -ascii;

x_values = [];

```

```

        fn_values = [];
        grad_log = [];
        deltas = [];
    end

elseif(RunType == 2)

    % load input information from DynaMIT directory
    cd DynaMIT

    load rt_choice.dat;
    load spddsy_vector.dat;
    load cap_vector.dat;
    load OD_vector.dat;

    od_seed = OD_vector(4:length(OD_vector));
    spddsy_seed = spddsy_vector;
    cap_seed = cap_vector(2:length(cap_vector));
    rtchoice_param = rt_choice;

    % Pre-set parameters for calibration
    no_ODs = OD_vector(1);
    no_intervals = OD_vector(2);
    no_segments = cap_vector(1);
    no_groups = length(spddsy_seed)/6;
    demand_factor = OD_vector(3);
    no_other_params = length(rtchoice_param);

    paramsPerDay = no_ODs*no_intervals + no_segments + 6*no_groups + no_other_params;

    %Load Weights (Optional)
    weights_theta = ones(paramsPerDay,1);

    theta_0_dynamit = [od_seed; cap_seed; spddsy_seed; rtchoice_param];
    theta_0_spsa = ones(paramsPerDay,1);
    p = length(theta_0_spsa); % Set p, the dimension of the parameter vector

    cd ..

    %Initial Objective Value
    initial_fn_value = FUNC(theta_0_dynamit);
    tmp_x = theta_0_spsa';
    tmp_fn = initial_fn_value;
    save ./Results/fn_theta_0.dat tmp_fn -ascii;

```



```

%Initialize
x_values = [];
fn_values = [];
fn_path = [];
fn_path = [fn_path; initial_fn_value];
grad_log = [];
deltas = [];
nstart = 0;

end

mag_scale = 1; % we scale delta by this percent of parameter magnitude
lossfinalsq=0;      %variable for cum.(over 'cases')squared loss values
lossfinal=0;       %variable for cum. loss values

%Lower bound
lb = [];
lb = [lb; theta_0_spsa(1:no_ODs*no_intervals)*0.5]; %OD
lb = [lb; theta_0_spsa(no_ODs*no_intervals+1:no_ODs*no_intervals+no_segments)*0.5]; %capacity
lb = [lb; theta_0_spsa(no_ODs*no_intervals+no_segments+1:paramsPerDay-1)*0.1]; %spddsy
lb = [lb; theta_0_spsa(paramsPerDay,1)*0.5]; %behavior

%Upper bound
ub = [];
ub = [ub; theta_0_spsa(1:no_ODs*no_intervals)*2]; %OD
ub = [ub; theta_0_spsa(no_ODs*no_intervals+1:no_ODs*no_intervals+no_segments)*2]; %capacity
ub = [ub; theta_0_spsa(no_ODs*no_intervals+no_segments+1:paramsPerDay-1)*10]; %spddsy
ub = [ub; theta_0_spsa(paramsPerDay,1)*1.5]; %behavior

theta_lo_spsa = lb;
theta_hi_spsa = ub;

%Main Loop
for i=1:cases

    theta = theta_0_spsa; % Start at seed parameter values

    for k= nstart:nstart+n-1

        ak = a/(k+1+A)^alpha;
        ck = c/(k+1)^gamma;
        ck = ck.*weights_theta;
        ghat = zeros(paramsPerDay,1);

        tmp_fn_grad_reps = []; % Store the fn values for averaging

```

```

for xx = 1:grad_reps,

    delta = 2*round(rand(p,1))-1;
    deltas = [deltas; (ck.*delta)'];

    save ./Results/deltas.dat deltas -ascii;

    thetaplus = theta + ck.*delta;
    thetaminus = theta - ck.*delta;

    % These four lines below invoke component-wise constraints
    thetaplus=min(thetaplus,theta_hi_spsa);
    thetaplus=max(thetaplus,theta_lo_spsa);
    thetaminus=min(thetaminus,theta_hi_spsa);
    thetaminus=max(thetaminus,theta_lo_spsa);

    % Store the three x values
    tmp_x = [thetaminus' tmp_x thetaplus'];
    x_values = [x_values; tmp_x];
    tmp_x = theta'; % reset to theta for next grad rep

    yplus=FUNC(thetaplus.*theta_0_dynamit);
    yminus=FUNC(thetaminus.*theta_0_dynamit);

    % Store the two fn values
    tmp_fn = [yminus yplus];
    tmp_fn_grad_reps = [tmp_fn_grad_reps; tmp_fn];

    thetadif = thetaplus - thetaminus;

    ghat= ghat + ((yplus - yminus)./ thetadif);

end % end of reps for grad averaging

aa = mean(tmp_fn_grad_reps(:,1:1));
bb = mean(tmp_fn_grad_reps(:,2:2));

tmp_fn = [aa bb];
fn_values = [fn_values; tmp_fn];

ghat = ghat/grad_reps;

grad_log = [grad_log; ak*ghat'];
save ./Results/grad_log.dat grad_log -ascii;

```

```

theta=theta-ak*ghat;

% Two lines below invoke component-wise constraints
theta=min(theta,theta_hi_spsa);
theta=max(theta,theta_lo_spsa);

path_y = FUNC(theta.*theta_0_dynamit);
fn_path = [fn_path; path_y];
%cd(main_dir);
save ./Results/fn_path.dat fn_path -ascii;

% Reset tmp_x and save x_values, fn_values to disk
tmp_x = theta';
save ./Results/x_values.dat x_values -ascii;
save ./Results/fn_values.dat fn_values -ascii;

end % iterations (k = 0:n-1)

% save current fn values (at theta)

save ./Results/x_final.dat theta -ascii;

lossvalue=path_y ;
lossfinalsq=lossfinalsq+lossvalue^2;
lossfinal=lossfinal+lossvalue;

end % replications (i = 1:cases)

% Display results: Mean loss value and standard deviation
%
disp('mean loss value over "cases" runs')

final_fn = lossfinal/cases

save ./Results/final_fn.dat final_fn -ascii;
save ./Results/final_c.dat c -ascii;
%
if cases > 1
    disp('sample standard deviation of mean loss value')
    sd=((cases/(cases-1))^0.5)*(lossfinalsq/cases-(lossfinal/cases)^2)^0.5;
    sd=sd/(cases^0.5)
else
end

```

```

%Objective Function Evaluation For BTRC Project
%By Zheng, Dec 19th, 2009
%We don't use the first 2 intervals

function y = FUNC(X)

global no_ODs          % # OD pairs
global no_intervals   % # Time intervals
global no_segments    % # Segments
global no_groups       % # Segment groups
global no_other_params % # other parameters
global paramsPerDay    % # parameters to be calibrated per day of data
global day
global counter_glob;
global main_dir

counter_glob = counter_glob + 1
n_iterations = 1; % no. of runs over which averging is done

format long;
bigsum = 0;
z = 0;

%Weights
Weight_FCD = 0.01;
Weight_Counts = 1;
Weight_Speed = 1;
Weight_Priori = 1;

day

CHANGEPARAMETERS(X); % Change Directory in it

for counter = 1:n_iterations,

    str = ['./backupToDir'];
    % temp_dir' int2str(counter_glob)];
    system(str);

    % Run DynaMIT-P
    system('./DynaMIT_P dtaparam.dat');

% Compare the FCD data with eq_tt

```

```

%Read Equilibrium Travel time
eq_tt = dlmread('eq_tt.out', ' ', [4,0,no_segments + 3,272]);
load ../seed/FCD.dat;

%Orgnize data
EQTT=[];
for i=1:no_segments
    EQTT(eq_tt(i,1),:) = eq_tt(i,:);
end

diff_FCD =0;
FCD_tt=[];
Sim_tt=[];
no_reliable_FCD = 0;
[no_FCD_data,temp] = size(FCD);

for i=1:no_FCD_data
    link = FCD(i,1);
    time = FCD(i,2) - 360+1;%starting from 6am = 360
    if (time>0 & FCD(i,3)>0 & EQTT(link,time+3:time+7)>0)
        FCD_tt = [FCD_tt;FCD(i,3)];
        Sim_tt=[Sim_tt;sum(EQTT(link,time+3:time+7))/5];
    end
end

weights_FCD = ones(size(FCD_tt),1)*Weight_FCD;
dev = FCD_tt - Sim_tt;
no_reliable_FCD = length(dev);
diff_FCD = dev'*(weights_FCD.*dev);
RMSE_FCD = RMSE(FCD_tt,Sim_tt);
RMSN_FCD = RMSN(FCD_tt,Sim_tt);

% Load and Compute Counts
% Load counts
load ../seed/counts_seed.dat;

% Load Simulated Counts

for i = 1:nIntervals
    str = ['load Sim' int2str(i) '.dat'];
    eval(str)
end

sim_counts = [];

```

```

nSensors = length(sim1);
for i = 1:nIntervals
    str = ['sim_counts = [sim_counts; Sim' int2str(i) ']''];
    eval(str);
end

counts_seed = counts_seed(2*nSensors+1:nIntervals*nSensors);
sim_counts = sim_counts(2*nSensors+1:nIntervals*nSensors);
dev = counts_seed - sim_counts;

reliable_data_index = find(counts_seed > 0);
diff_counts = dev(reliable_data_index)'*dev(reliable_data_index);

no_reliable_data = length(reliable_data_index);

RMSE_counts = RMSE(counts_seed(reliable_data_index),sim_counts(reliable_data_index)) ;
RMSN_counts = RMSN(counts_seed(reliable_data_index),sim_counts(reliable_data_index)) ;

% Load and Compute speeds
% Load observed speed
load ../seed/speeds_seed.dat;

% Load Simulated speeds
cd output/;
load sen_spd_Est_060000-100000.out;
spd = sen_spd_Est_060000_100000;
spd_sim = [];
cd ..;

[n nSensors] = size(spd)
nSensors = nSensors - 1
nIntervals;

for i = 1:nIntervals
    for j = 1:nSensors+1
        min1 = (i-1)*min_interval + 1 ;
        min2 = min1 + min_interval - 1;
        avg = 0;
        for k = min1+1:min2+1
            avg = avg + spd(k,j);
        end
        avg = avg / min_interval;
        spd_sim = [spd_sim; avg];
    end
end
end

```

```

spd_sim(isnan(spd_sim)) = 0;

weights_Speed = ones(size(speeds_seed),1)*Weight_Speed;
%Do not calculate the first 2 intervals
speeds_seed = speeds_seed(2*nSensors+1:nIntervals*nSensors);
spd_sim = spd_sim(2*nSensors+1:nIntervals*nSensors);
dev = speeds_seed - spd_sim;

diff_Speed = dev(reliable_data_index)*dev(reliable_data_index);

reliable_data_index = find(speeds_seed > 0);
no_reliable_data = length(reliable_data_index)
RMSE_Speed = RMSE(speeds_seed,spd_sim)
RMSN_Speed = RMSN(speeds_seed,spd_sim)

% Add to total
bigsum = bigsum + diff_FCD + diff_counts;%+ RMSE_Speed;

end

cd ../Results

if(counter_glob == 1)
    fn_FCD = [];
    fn_Counts = [];
    fn_Speed = [];
    fn_FCD = [1 diff_FCD];
    fn_Counts = [1 diff_counts];
    fn_Speed = [1 diff_Speed];
    fn_RMSE = [];
    fn_RMSN = [];
    fn_RMSE = [1 RMSE_FCD RMSE_counts RMSE_Speed];
    fn_RMSN = [1 RMSN_FCD RMSN_counts RMSN_Speed];

    save fn_FCD.dat fn_FCD -ascii;
    save fn_Counts.dat fn_Counts -ascii;
    save fn_Speed.dat fn_Speed -ascii;
    save fn_RMSE.dat fn_RMSE -ascii;
    save fn_RMSN.dat fn_RMSN -ascii;

else if( mod(counter_glob, 3) == 1 && counter_glob ~=1 )

    load fn_FCD.dat;
    fn_FCD = [fn_FCD ;counter_glob diff_FCD];

```

```

save fn_FCD.dat fn_FCD -ascii;

load fn_Counts.dat;
fn_Counts = [fn_Counts ;counter_glob diff_counts];
save fn_Counts.dat fn_Counts -ascii;

load fn_Speed.dat;
fn_Speed = [fn_Speed;counter_glob diff_Speed];
save fn_Speed.dat fn_Speed -ascii;

load fn_RMSE.dat;
fn_RMSE = [fn_RMSE ; counter_glob RMSE_FCD RMSE_counts RMSE_Speed];
save fn_RMSE.dat fn_RMSE -ascii;

load fn_RMSN.dat;
fn_RMSN = [fn_RMSN ; counter_glob RMSN_FCD RMSN_counts RMSN_Speed];
save fn_RMSN.dat fn_RMSN -ascii;

end
end

cd ..

bigsum = bigsum/n_iterations;

y = bigsum; % Return total objective function for day

```


Appendix B

Fit-To-Counts for Every Simulation Interval

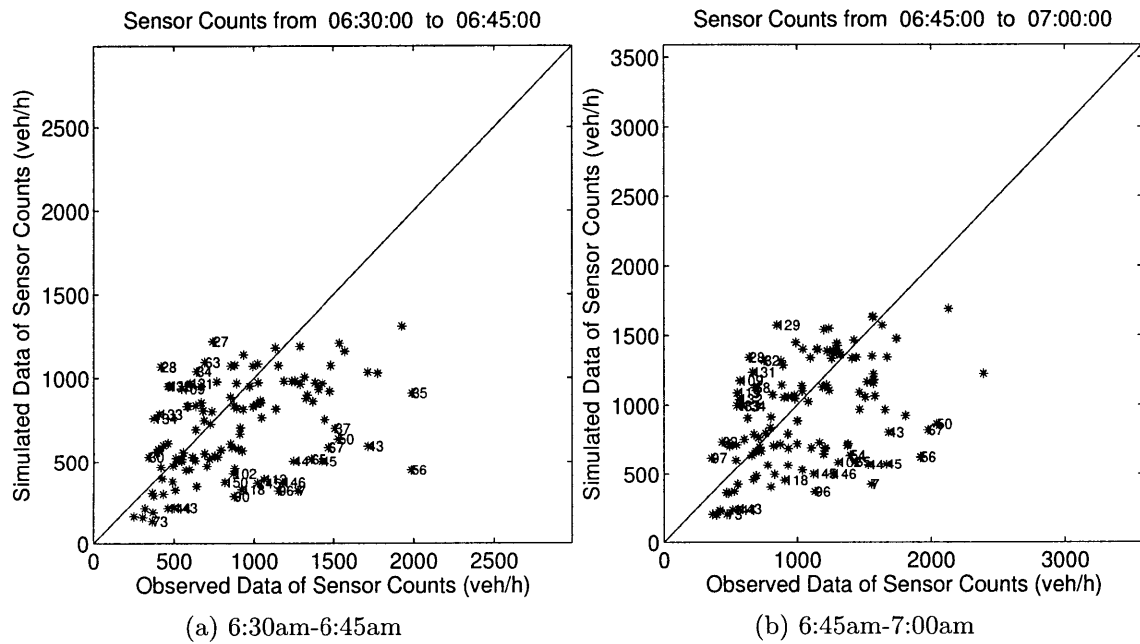


Figure B-1: Fit-To-Counts for 6:30 to 7:00

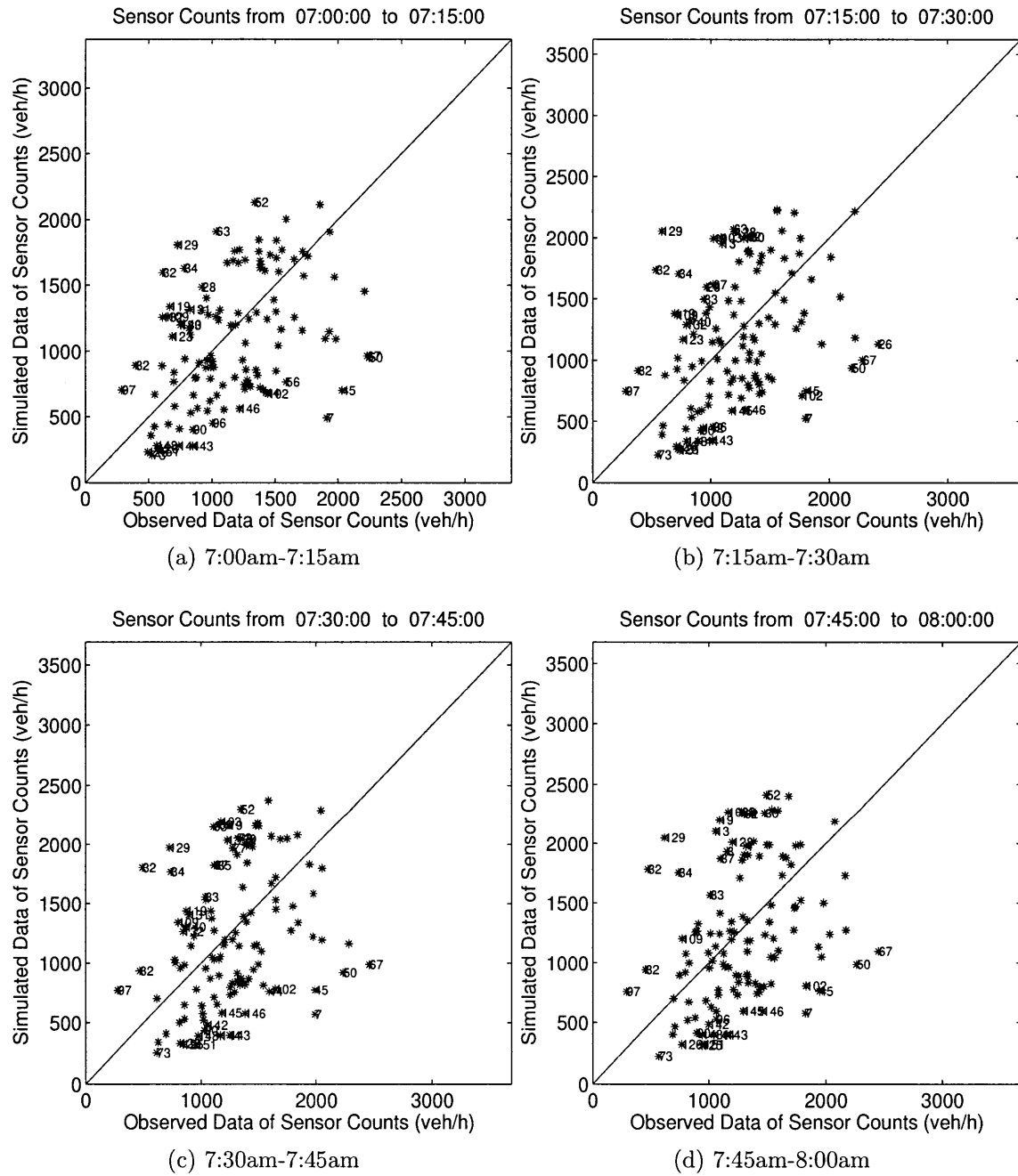


Figure B-2: Fit-To-Counts for 7:00 to 8:00

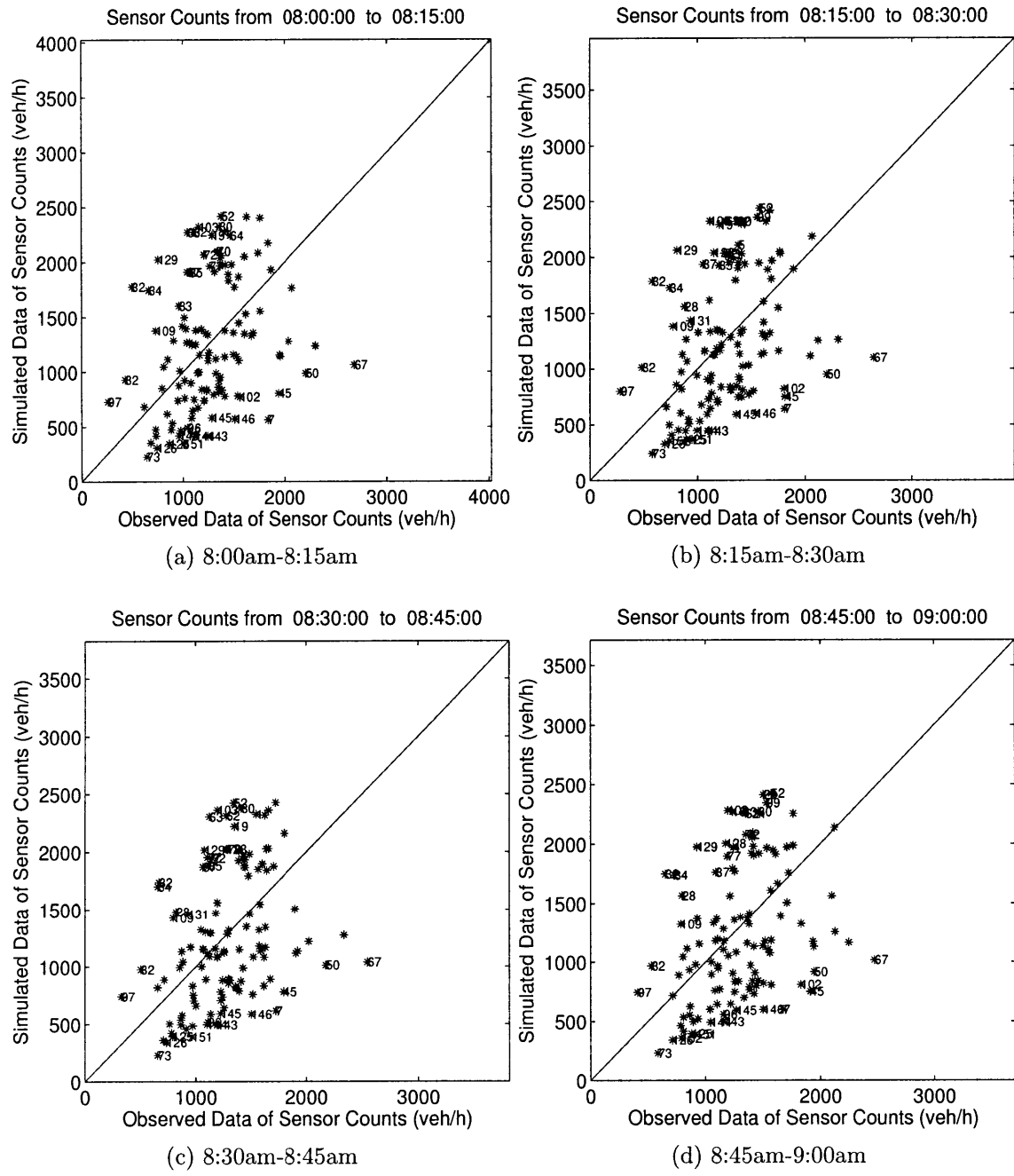


Figure B-3: Fit-To-Counts for 8:00 to 9:00

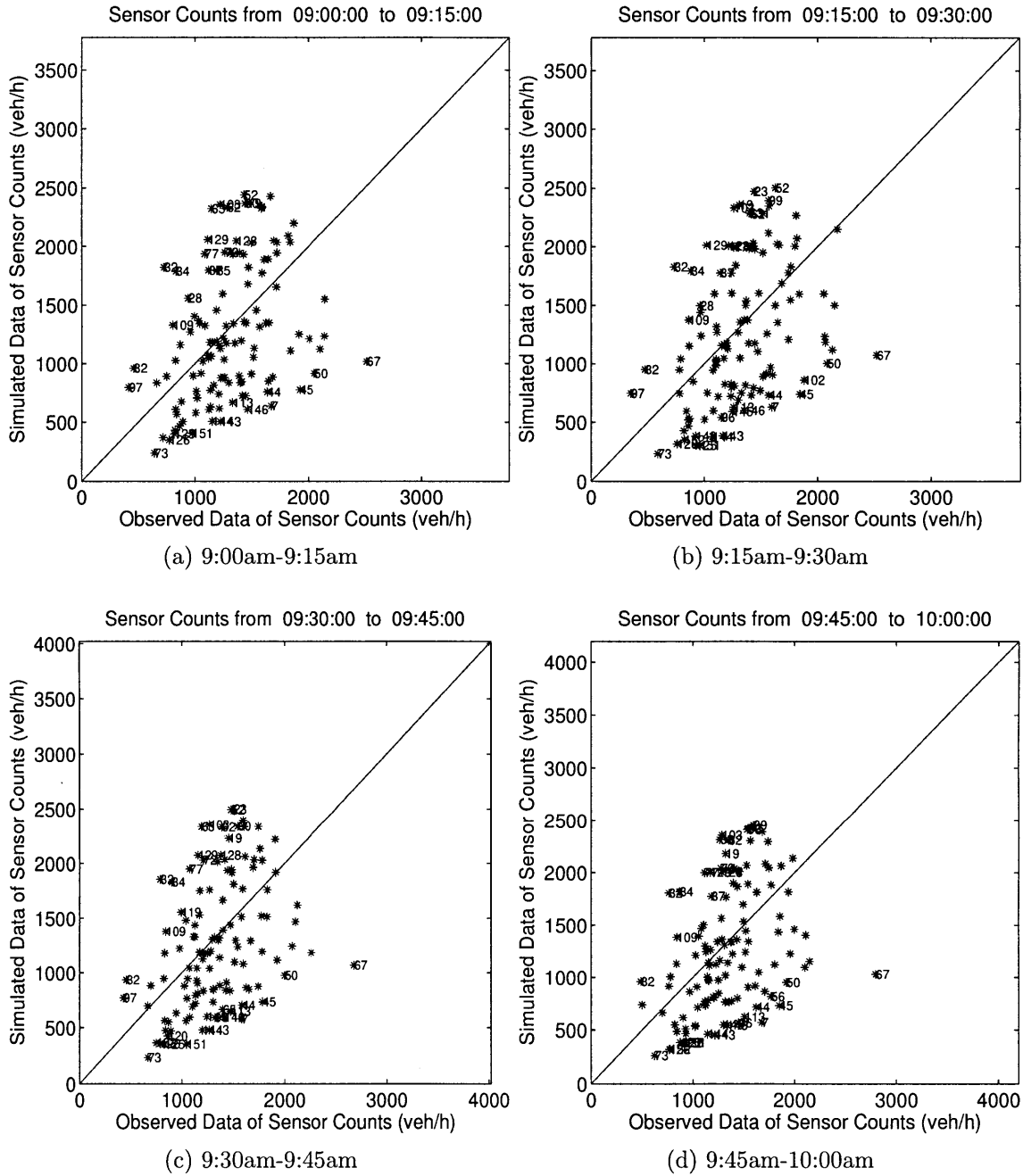


Figure B-4: Fit-To-Counts for 9:00 to 10:00

Bibliography

- M. Van Aerde, B. Hellings, M. Baker, and H. Rakha. INTEGRATION: An overview of traffic simulation features. *Transportation Research Records*, 1996.
- K Ashok and M Ben-Akiva. Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Transportation Science*, Vol.36(2):184–198, 2002.
- R Balakrishna. *Off-line Calibration of Dynamic Traffic Assignment Models*. PhD thesis, MIT, 2006.
- R. Balakrishna, H.N Koutsopoulos, and M. Ben-Akiva. Simultaneous Off-line Demand and Supply Calibration of Dynamic Traffic Assignment Systems. In *Presented at the 85th Annual Meeting of the Transportation Research Board*, 2006.
- R Balakrishna, Y Wen, M. Ben-Akiva, and C Antoniou. Simulation-based framework for transportation network management for emergencies. *Journal of the Transportation Research Board*, 2041:80–88, 2008.
- J. Barcelo and J. Casas. Heuristic dynamic assignment based on microscopic simulation. In *Proceedings of the 9th Meeting of the Euro Working Group on Transportation*, Bari, Italy, 2002.
- J. Barcelo and J Casas. Dynamic network simulation with AIMSUN. In *Proceedings of the Internat. Symposium on Transport Simulation*, Kluwer, Yokohama, 2003.
- J. Barcelo and J. Casas. Stochastic heuristic dynamic assignment based on AIMSUN microscopic traffic simulator. *Transportation Research Record: Journal of the Transportation Research Board*, 1964:70–80, 2006.
- M. Ben-Akiva and M. Bierlaire. *Discrete Choice Models with Applications to Departure Time and Route Choice*, chapter 7, page 37. Kluwer Academic Publishers, Boston, 2003.
- M. Ben-Akiva and S. R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, Cambridge, Massachusetts, 1985.
- M. Ben-Akiva and S. Ramming. Lecture notes: discrete choice models of traveler behavior in networks. Prepared for Advanced Methods for Planning and Management of Transportation Networks. Capri, Italy., 1998.

- M. Ben-Akiva, M. Bierlaire, J. Bottom, H. N. Koutsopoulos, and R. G. Mishalani. Development of a route guidance generation system for real-time application. In *Proceedings of the 8th International Federation of Automatic Control Symposium on Transportation Systems*, Chania, Greece,, 1997. IFAC.
- M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani. Network State Estimation and Prediction for Real-Time Transportation Management Applications. *Networks and Spatial Economics*, Vol.1(3/4):291–318, 2001a.
- M. Ben-Akiva, Margaret Cortes, Angus Davol, Haris Koutsopoulos, and Tomer Toledo. MITSIMLab: Enhancements and Applications for Urban Networks. In *9th World Conference on Transportation Research (WCTR)*, Seoul, Korea, 2001b.
- M. Ben-Akiva, M. Bierlaire, H. N. Koutsopoulos, and R. Mishalani. *Real-Time Simulation of Traffic Demand-Supply Interactions within DynaMIT*, pages 19–36. Kluwer, 2002a.
- M. Ben-Akiva, A. Davol, T. Toledo, and H.N. Koustopoulos. Calibration and evaluation of MITSIMLab in Stockolm. In *81st Transportation Research Board Meeting*. MIT, Intelligent Transportation Systems Program, 2002b.
- M. Ben-Akiva, S. Gao, Z. Wei, and Y. Wen. A dynamic traffic assignment model for a highly congested network. In Preparation, 2010.
- M. J. Box. A new method of constrained optimization and a comparison with other methods. *Computer Journal*, vol. 8, no. 1:42–52., 1965.
- Sharon Adams Boxill and Lei Yu. An Evaluation of Traffic Simulation Models for Supporting ITS Development. Technical report, Texas Southern University, 2000.
- Caliper Corporation. A dynamic traffic simulation model on planning networks. In *TRB Planning Application Conference*, Houston,, 2009.
- Cambridge Systematics, Inc. A Recommended Approach to Delineating Traffic Analysis Zones in Florida. Technical report, Cambridge Systematics, Inc., 2007.
- E Cascetta. *Transportation Systems Engineering: Theory and Methods*. Kluwer Academic Publishers, Dordrecht, Boston, MA,, 2001.
- E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta. A modified logit route choice model overcoming path overlapping problems. specification and some calibration results for interurban networks. In *Proceedings of the 13th International Symposium on the Theory of Road Traffic Flow*, Lyon, France,, 1996.
- Y. C. Chiu, H. Zheng, J. A. Villalobos, W. Peacock, and R. Henk. Evaluating Regional Contra-Flow and Phased Evacuation Strategies for Texas Using a Large-Scale Dynamic Traffic Simulation and Assignment Approach. *Journal of Homeland Security and Emergency Management*, 5:(1), 2008.

- A. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm. *ACM Transactions on Mathematical Software*, 13:262–280, 1987.
- C. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research B*, 28B(4), 1994.
- C. Daganzo. The cell-transmission model, part II: Network traffic. *Transportation Research B*, 29B(2):79–93, 1995.
- A. de Palma and F. Marchal. Real cases applications of the fully dynamic METROPOLIS tool-box: an advocacy for large-scale mesoscopic transportation systems. *Networks and Spatial Economics*, 2:347–369, 2002.
- R. A. Dittberner and R. Kerns. Traffic Simulation in Congested Urban Networks: Pennsylvania Avenue Case. *ITE Journal*, Vol 72:No.10, 2002.
- H. and Haj-Salem Elloumi, N. and et al. METACOR: A macroscopic Modelling Tool for Urban Corridor. In *TRIennial Symposium on Transport ANalysis*, Capri, 1994.
- FHWA. *Highway Capacity Manual*. Federal Highway Administration, Washington, DC, USA, 2000.
- FHWA. Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation. Technical report, Federal Highway Administration, 2005.
- M. Florian, M. Mahut, and N. Tremblay. A hybrid optimization-mesoscopic simulation dynamic traffic assignment model. In *Proceeding In IEEE Intelligent Transportation Systems Conference*, pages 118–121, Oakland, California, USA,, 2001.
- M. Florian, M. Mahut, and N. Tremblay. Application of a simulation based dynamic traffic assignment model. *Simulation Approaches in Transportation Analysis*, Recent Advances and Challenges:1–22, 2005.
- M. Florian, M. Mahut, and N Tremblay. A simulation based dynamic traffic assignment: the model, solution algorithm and applications. In *International Symposium of Transport Simulation 2006, ISTS06 Compendium of Papers CDROM*. EPFL, Lausanne, Switzerland., 2006.
- C. Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393–407, 1998.
- G. Gentile, L. Meschini, and N. Papola. Spillback congestion in dynamic traffic assignment: A macroscopic flow model with time-varying bottlenecks. *Transportation Research Part B: Methodological*, 41 (10):1114–1138, 2007.

- D.E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, New York, 1989.
- J.H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Michigan, 1975.
- R. Hooke and T.A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM*, vol. 8:212–229, 1961.
- John T Hughes. AIMSUN2 simulation of a congested Auckland freeway. *Transportation Planning*, state of the art:141–161, 2002.
- W. Huyer and A Neumaier. SNOBFIT stable noisy optimization by branch and fit. *ACM Transactions on Mathematical Software*, Vol. 35,:No. 2, 2006.
- Juran Ido and Joseph N. Prashker. A dynamic traffic assignment model for the assessment of moving bottlenecks. *Transportation Research Part C*, 17(2009):240–258, 2009.
- D. Ismart. Calibration and Adjustment of System Planning Models. Technical report, Federal Highway Administration, Washington, DC,, 1990.
- Isam A. Kaysi, Mohamed S. Moghrabi, and Hani S. Mahmassani. Hot Spot Management Benefits: Robustness Analysis for a Congested Developing City. *Journal of Transportation Engineering*, Vol. 129, No. 2:203–211, 2003.
- J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, vol. 23, no. 3:462–466, 1952.
- Lawrence A. Klein. Vehicle Detector Technologies for Traffic Management Applications, Part 1. ITS Online, 1997.
- K. K. Kunde. Calibration of mesoscopic traffic simulation models for dynamic traffic assignment. Master’s thesis, Massachusetts Institute of Technology, 2002.
- D. R. Leonard, P. Power, and N. B. Taylor. CONTRAM: structure of the model. Technical report, Transportation Research Laboratory, 1989.
- M. H. Lighthill and G. B. Whitham. On kinematic waves II: a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London*, 1955.
- Yue Liu, Jie Yu, Gang-Len Chang, and S. Rahwanji. A Lane-group Based Macroscopic Model for Signalized Intersections Account for Shared Lanes and Blockages. In *Intelligent Transportation Systems, 11th International IEEE Conference*, pages 639–644, 2008.
- H. Mahmassani, X. Qin, and X. Zhou. DYNASMART-X evaluation for realtime tmc application: Irvine test bed: TrEPS phase 1.5b final report. Technical report. Technical report, Maryland Transportation Initiative, University of Maryland, College Park, Maryland., 2004.

- H. S. Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1(3/4):267–292, 2001.
- H. S. Mahmassani and Y. Hawas. Data requirements for development, calibration of dynamic traffic models, and algorithms for ATMS/ATIS. In *Presented at the 76th Annual Meeting of the Transportation Research Board*, 1997.
- M. Mahut. A multi-lane link model of traffic dynamics based on the spacetime queue. In *Proceeding in IEEE Intelligent Transportation Systems Conference*, pages 122–126, Oakland, California, USA., 2001.
- M Mahut. DTA in practice: Modeling dynamic networks in the real world. In *TRB Planning Application Conference*, Houston, 2009.
- M. Mahut, M. Florian, N. Tremblay, M. Campbell, D. Patman, and Z. K. McDaniel. Calibration and application of a simulation-based dynamic traffic assignment model. *Transportation Research Record: Journal of the Transportation Research Board*, 1876:101–111, 2004.
- N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, vol.21:1087–1092, 1953.
- J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, vol. 7, no. 4:308–313., 1965.
- M.and J.-M. Blosseville Papageorgiou and et al. Macroscopic modelling of traffic flow on the Boulevard Peripherique in Paris. *Transportation Research B*, 23B:29–47, 1989.
- H. J. Payne. Models of freeway traffic and control. *Simulation Council*, 1:51–61, 1971.
- S Peeta and A. K. Ziliaskopoulos. Foundations of Dynamic Traffic Assignment: The Past, the Present and the Future. *Networks and Spatial Economics*, 1:233–265, 2001.
- R.S. Pindyck and D. L. Rubinfeld. *Econometric models and economic forecasts*. Irwin McGraw-Hill,, Boston MA, 4th edition, 1997.
- The Smartest Project. Review of Micro-Simulation Models Appendix D: Analysis of Tools. Technical report, Leeds University, 1997.
- PTV. *VISSIM User’s Manual 4.10*. PTV Planung Transport Verkehr AG, 2005.
- M. S. Ramming. *Network Knowledge and Route Choice*. PhD thesis, MIT, 2002.

- V. Rathi, C. Antoniou, Y. Wen, M. Ben-Akiva, and M. Cusack. Assessment of the Impact of Dynamic Prediction-Based Route Guidance using a simulation-based, closed-loop framework. In *Presented at the 87th annual meeting of the Transportation Research Board. January 13-17*, Washington, D.C, 2008.
- P. I. Richards. Shockwaves on the Highway. *Operations Research*, 4:42–51, 1956.
- David Schrank and Tim Lomax. The 2009 URBAN MOBILITY REPORT. Technical report, Texas Transportation Institute, The Texas A&M University System, 2009.
- M. Smith and et al. S. Druitt. Paramics Final Report. Technical report, University of Edinburgh, 1994.
- J. C Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, Vol,37:332–341, 1992.
- J. C. Spall. *Stochastic Optimization, Stochastic Approximation and Simulated Annealing*, pages 529–542. Wiley-Interscience, 1999.
- J.C. Spall. Implementation of the Simultaneous Perturbation Algorithm for Stochastic Approximation. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34: 817–823, 1998.
- Jianping Sun. Introduction of beijing transportation system. Beijing Transportation Research Center, 2009.
- N. B. Taylor. The CONTRAM dynamic traffic assignment model. *Networks and Spatial Economics*, 3(3):297–322, 2003.
- Zhili Tian. Capacity Analysis of Traffic-Actuated Intersections. Master’s thesis, Massachusetts Institute of Technology, 2002.
- C. O. Tong and S. C. Wong. A predictive dynamic traffic assignment model in congested capacity-constrained road networks. *Transportation Research Part B*, vol 34:625–644, 2000.
- H. R. Varia and S. L. Dhingra. Dynamic User Equilibrium Traffic Assignment on Congested Multidestination Network. *Journal of Transportation Engineering*, Vol.130, No. 2:211, 2004.
- Y Wen. *Scalability of Dynamic Traffic Assignment*. PhD thesis, MIT, Massachusetts Institute of Technology, Cambridge, MA., 2009.
- Y. Wen, R. Balakrishna, M. Ben-Akiva, and S. Smith. Online deployment of Dynamic Traffic Assignment: architecture and run-time management. In *IEE Proceedings Intelligent Transport Systems*, pages 153(1):76–84, 2006a.

- Y. Wen, R. Balakrishna, M. Ben-Akiva, and S. Smith. Online deployment of Dynamic Traffic Assignment: Evaluation and lessons. In *the 87th annual meeting of the Transportation Research Board, DVD-ROM*, Washington, D.C., USA, 2008.
- Yang Wen, Ramachandran Balakrishna, Ashish Gupta, Moshe Ben-Akiva, and Scott Smith. Deployment of DynaMIT in the City of Los Angeles, Final Report. Technical report, MIT, 2006b.
- Shunan Xu. Development and Test of Dynamic Congestion Pricing Model. Master's thesis, Massachusetts Institute of Technology, 2009a.
- Shunan Xu. Technical Memorandum and User Guide on the Enhancement of DynaMIT for Variable Capacity. Technical report, Massachusetts Institute of Technology, Intelligent Transportation Systems Program, 2009b.
- A.K. Ziliaskopoulos, S.T. Waller, Y. Li, and M. Byram. Large Scale Dynamic Traffic Assignment: Implementation Issues and Computational. *Journal of Transportation Engineering*, Volume 130, No. 5:585–593, 2004.