W-SPSA: An Efficient Stochastic Approximation Algorithm for the Off-line Calibration of Dynamic Traffic Assignment Models

by

Lu Lu

Bachelor of Science in Automation
Tsinghua University, Beijing, China, 2011

Submitted to the Department of Civil and Environmental Engineering and the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Master of Science in Transportation
and

Master of Science in Electrical Engineering and Computer Science at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

© Massachusetts Institute of Technology 2013. All rights reserved.

Author

Department of Civil and Environmental Engineering
Department of Electrical Engineering and Computer Science

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

Francisco C. Pereira
Research Affiliate, Department of Civil and Environmental Engineering
Thesis Supervisor

Leslie A. Kolodziejski
Chair of the Committee on Graduate Students

Heidi Nepf
Chair, Departmental Committee for Graduate Students
W-SPSA: An Efficient Stochastic Approximation Algorithm for the Off-line Calibration of Dynamic Traffic Assignment Models

by

Lu Lu

Submitted to the Department of Civil and Environmental Engineering and the Department of Electrical Engineering and Computer Science on Dec 16, 2013, in partial fulfillment of the requirements for the degrees of Master of Science in Transportation and Master of Science in Electrical Engineering and Computer Science

Abstract

The off-line calibration is a crucial step for the successful application of Dynamic Traffic Assignment (DTA) models in transportation planning and real time traffic management. While traditional approaches focus on the separate or sequential estimation of demand and supply in a DTA system, a recently proposed framework calibrates the demand and supply models simultaneously by formulating the off-line calibration as a constrained optimization problem. Simultaneous Perturbation Stochastic Approximation (SPSA) has been reported in the literature to be the most suitable solution algorithm for this problem due to its highly efficient gradient estimation approach.

However, it turns out that the performance of SPSA in terms of convergence rate and long run accuracy can deteriorate significantly when the physical network size and the number of considered time intervals increase. To overcome this problem, this thesis proposes a new algorithm, called Weighted SPSA, or W-SPSA. W-SPSA improves SPSA’s gradient estimation process by effectively reducing the noise generated by irrelevant measurements. Synthetic tests are performed to systematically compare the performance of SPSA and W-SPSA. W-SPSA shows scalability and robustness in the tests and outperforms SPSA under different problem scales and characteristics. The application of W-SPSA in real world large-scale DTA systems is demonstrated with a case study of the entire Singapore expressway network. Results show that W-SPSA is a more suitable algorithm than SPSA for the off-line calibration of large-scale DTA models.

The contributions of the thesis include: 1) identifying limitations of a state-of-the-art solution algorithm for the DTA off-line calibration problem, 2) presenting rigorous definitions of an enhanced algorithm and proposing approaches to estimate the required algorithm parameters, 3) systematically comparing the performance of
the new algorithm against the state-of-the-art, 4) demonstrating the characteristics of the new algorithm through experiments, and 5) discussing the general steps and empirical technical considerations when tackling real world DTA off-line calibration problems.

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

Thesis Supervisor: Francisco C. Pereira  
Title: Research Affiliate, Department of Civil and Environmental Engineering
Acknowledgments

First of all, I would like to express my heartfelt gratitude to my research supervisor Professor Moshe Ben-Akiva. It has been more than three years since I first met him at Beijing international airport on July 6th, 2010. His invaluable guidance and constant help during my internship at MIT in the summer of 2010 and during my entire MIT graduate study changed my life.

I express my sincere thanks to my co-supervisor Dr. Francisco Pereira. Besides his insightful guidance on my research, his support and understanding during my hardest times helped me go through this tough but rewarding journey. Thanks to Professor Constantinos Antoniou for being a constant source of knowledge and inspiration in this work. Thanks to Professor Patrick Jaillet for the generous help with my dual Masters degree in EECS and evaluating my thesis.

I would also like to thank my academic advisor and administrators Professor Marta Gonzalez, Kris Kipp and Janet Fischer for the helps during my MIT study.

During my internship in the lab in the summer of 2010, I worked on a project funded by Beijing Transportation Research Center together with Professor Song Gao, Dr. Yang Wen, Zheng Wei and Issac Buenrostro. Thanks to them for providing me invaluable help and guidance. Thanks to my labmates Arnau Dria Torra, Joo Barbosa, Peter Swartz, Li Qu, Martin Milkovits for making my summer extremely happy and memorable. Thanks to Eric Huang for helping me with both research and life during the summer.

Thanks to Singapore-MIT Alliance for Research and Technology (SMART) for providing funding for my study and research at MIT, and my trips to Singapore. My three visits to Singapore was made fruitful and enjoyable by the constant help and companion from Yan Xu, Melani Jayasuriya, Stephen Robinson, Park Giho Edward, Vinh An Vu(Jenny). Thank you.

Thanks to my friends in 1-249 for being so friendly, helpful and making my everyday MIT life so pleasant. They are: Varun Pattabhiraman, Yin Wang, Yichen Sun, Xiang Song, Runmin Xu, He Sun, Rinal Chheda, Annesha Enam, Haizheng
Zhang, Tianli Zhou, Peiguang Hu, Nathanael Cox … and Professor Ruimin Li. Special thanks to Tina Xue, who was the first person ever who replied my Email with an mit.edu email address and helped me a lot with my summer internship, Katherine Rosa, who offered me so many generous helps before and after I join ITS lab, and Eunice Kim, who, together with Katie, organized everything for the lab so well and it was always enjoyable to talk with them.

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

1 Introduction ........................................ 13
  1.1 Dynamic traffic assignment systems ................. 14
   1.1.1 Generic structure .................................. 14
   1.1.2 Simulation-based DTA models ...................... 16
  1.2 Thesis motivation and problem statement ............ 18
  1.3 Thesis outline ....................................... 21

2 Literature Review .................................... 23
  2.1 Estimation of demand models ......................... 24
   2.1.1 Travel behavior models .......................... 24
   2.1.2 Estimation of OD flows ............................ 26
  2.2 Estimation of supply models ......................... 29
  2.3 Demand-supply joint calibration ...................... 31
   2.3.1 Iterative demand-supply calibration ............... 32
   2.3.2 Simultaneous demand-supply calibration .......... 35
  2.4 Summary ............................................. 38

3 Methodology ......................................... 41
  3.1 General problem formulation .......................... 41
  3.2 The SPSA algorithm ................................... 44
  3.3 Shortcomings of the application of SPSA to DTA calibration .... 48
  3.4 The Weighted SPSA algorithm ......................... 52
  3.5 Estimation of weight matrices ....................... 54
3.5.1 Analytical approximation ........................................ 55
3.5.2 Numerical approximation ........................................ 58
3.5.3 SPSA ............................................................. 59
3.5.4 Hybrid method ................................................... 60

4 Synthetic Tests ......................................................... 63
4.1 Experimental design .................................................. 63
4.2 Results ............................................................... 65
   4.2.1 Scalability ...................................................... 65
   4.2.2 Network correlation ........................................... 66
   4.2.3 Weight matrix .................................................. 68
   4.2.4 Sensitivity to noise ............................................ 73
4.3 Summary ............................................................. 74

5 Case Study .............................................................. 77
5.1 Model and parameters ............................................... 77
5.2 Network and data .................................................... 79
   5.2.1 Network overview .............................................. 79
   5.2.2 Data overview and cleaning .................................. 81
5.3 Calibration process and empirical considerations .................. 85
   5.3.1 Weight matrix calculation ..................................... 85
   5.3.2 Empirical considerations ..................................... 87
5.4 Results ............................................................... 90
5.5 Summary ............................................................. 96

6 Conclusion ............................................................. 99
6.1 Summary ............................................................. 99
6.2 Future research directions ......................................... 101
List of Figures

1-1  Generic DTA Systems Structure ........................................ 15
1-2  Off-line and On-line DTA Model Calibration .......................... 20
2-1  Iterative demand-supply calibration framework ....................... 33
3-1  Relative paths of SPSA and FDSA in a 2 parameter problem (Source: Spall (1998)) .................................................. 47
3-2  An example network for the illustration of estimation error in SPSA . 50
3-3  Work flow of W-SPSA ..................................................... 55
3-4  An example network for the illustration of analytical weight matrix estimation ....................................................... 56
3-5  The flowchart of one experiment in the numerical approximation approach ....................................................... 58
4-1  Scalability performance of SPSA and W-SPSA in terms of RMSN with number of algorithm iterations ............................. 66
4-2  Performance of SPSA and W-SPSA in terms of RMSN with different degrees of network correlation ............................. 67
4-3  Performance of SPSA and W-SPSA with different weight matrices . 68
4-4  The influence of unknown weights ......................................... 70
4-5  The influence of number of iterations in the numerical weight matrix approximation approach to final RMSN ..................... 71
4-6  The influence of number of iterations in the numerical weight matrix approximation approach to sensitivity and precision ............ 72
4-7 Sensitivity and precision under different threshold 72
4-8 Influence of different levels noise from the simulator 74
4-9 Influence of different levels noise from measurements 75
5-1 Singapore road network (OpenStreetMap) 80
5-2 Computer representation of the entire Singapore expressway system in DynaMIT 80
5-3 Example of cameras that measures traffic flow on Singapore expressways (photo taken by Stephen Robinson) 82
5-4 Three scenarios in the flow data consistency check 83
5-5 An example of two inconsistent sensors under case 1 scenario 84
5-6 An example of two inconsistent sensors under case 2 scenario 84
5-7 Fit-to-counts at different intervals 95
5-8 Cumulative counts at different sensors 96
5-9 Some sensor errors that may caused by systematic network/data issues 97
List of Tables

3.1 Assignment fractions of different OD flows to different measurements 57

5.1 Overall calibration results ........................................... 92
5.2 Fit-to-counts by time interval ..................................... 93
5.3 Fit-to-speeds by time interval ..................................... 94
Chapter 1

Introduction

With the rapid urbanization and economy growth, people's demands for travel have never been greater. Particularly in road (highway) systems, demand continues to grow in a rate much higher than the growth rate of transportation infrastructures. According to the *Highway Statistics 2011* (FHWA, 2011), from 1982 to 2011, the total length of US highways increased from 3,865,894 miles to 4,094,447 miles, or 5.9%. Meanwhile, the total miles of vehicle trips increased from 1,595,010 million miles to 2,964,720 million miles, or 85.8%. As a consequence, the degree and frequency of traffic congestion increased dramatically. According to *2012 Urban Mobility Report* (Schrank et al., 2012), congestion affected each individual driver by an average delay of 38 hours and a cost of 818 US dollars in 2011, while the numbers were 16 hours and 342 US dollars in 1982 (in 2011 US dollar value). This report also points out that transportation is now the second largest emitting sector of carbon dioxide greenhouse gases, which are argued to be a major cause of global warming and other environmental problems.

The government and academia have paid more and more attention on traffic congestion problems due to their significant social, economic and environmental impact. As the expansion of the road networks to increase capacity is limited by physical and financial constraints, more efficient traffic management and better utilization of the existing infrastructures become the key solutions to congestion problems. Intelligent Transportation Systems (ITS) are believed to be a most promising means that take
advantage of the innovations and breakthroughs in mobile, computing, and surveillance technology to improve traveler information and traffic management systems. The wide deployment of ITS in the past years provides real time traffic data from a variety of advanced sensors (e.g., loop detectors, Global Positioning System (GPS) equipped floating cars, high definition video cameras, etc.) with higher quality and coverage. Extensive efforts have been made by researchers and practitioners to better utilize the abundant traffic data to accurately model the traffic dynamics, capture drivers' behaviors and their response to information, evaluate network performance, and provide consistent traffic predictions. These works are essential for the mitigation of congestions: better estimation of traffic states and their evolution enables traffic planners to design and evaluate more effective network configurations and traffic policies; more accurate route guidance information saves individual driver's time and at the same time reduce the level of congestion in the network by more balanced road usage. Dynamic traffic assignment (DTA) models are designed to achieve these goals and they have shown great potentials through their successful applications across the world.

1.1 Dynamic traffic assignment systems

1.1.1 Generic structure

Dynamic traffic assignment (DTA) systems are capable of capturing the complex dynamics in a traffic network and providing consistent estimation and prediction of the transition of network states. Peeta and Ziliaskopoulos (2001) present an early comprehensive review of DTA researches. The generic DTA model structure is shown in Figure 1-1. DTA models typically consist of two components (Cascetta, 2001; Florian et al., 2001; Ben-Akiva et al., 2001): 1) a set of demand models that focus on the way travelers make trip decisions (origin-destination, departure time, mode, route, etc.), and 2) a set of supply models that decide the movement patterns of traffic in the network (signal timing, segment capacity, speed-density functions, etc.).
Sophisticated algorithms captures the interaction between demand models and supply models, create a consistent framework that output the estimated traffic conditions given model input and parameter values.

![Diagram of DTA System](image)

Figure 1-1: Generic DTA Systems Structure

Under this general framework, different types of DTA systems may have different means to model the demand, supply, and their interactions. DTA systems can be categorized into two very different mainstreams: analytical models, which rely on mathematical formulations and optimizations, and simulation-based models, which rely on computer simulation techniques.

Analytical DTA models approximate the dynamic traffic assignment problem through mathematical formulations and explicit constraints. Optimization algorithms are usually applied to solve for unknowns in the formulation. Three major types of analytical DTA models have been developed: mathematical programming formulations, optimal control formulations, and variational inequality formulations. More detailed review of the three formulations can be found in Balakrishna (2006) and Peeta and Ziliaskopoulos (2001).

Despite the rigorous mathematical formulation, the usually simplified assumptions in analytical DTA models make them unable to capture the complex patterns of traffic congestion and the travel behaviors of drivers. On the other hand, attempts
to more closely represent such patterns require much more complex formulations and constraints, which lead to enormous computational overhead. Advanced traffic management systems require accurate network state estimation and real time prediction in large-scale congested networks, which highly depends on the ability to model drivers' behavior and their response to information, and provide consistent estimation with reasonable computational burden. Simulation-based DTA systems possess such capabilities and are reviewed in detail in next sub-section.

1.1.2 Simulation-based DTA models

Simulation-based DTA systems can be divided into three categories based on the level of detail with which they model traffic dynamics: microscopic models, mesoscopic models and macroscopic models.

Macroscopic DTA models have the least detailed representation of studied network and traffic dynamics, yet they have the advantage of computational efficiency, especially on very large networks. Such systems model traffic as uniformed fluid flows without distinguishing individual cars. Physical concepts and theories are applied to simulate the propagation of the traffic flows in the network. An example of macroscopic simulation-based DTA models is the Cell Transmission Model (CTM), (Daganzo, 1994), in which the road network is modeled with cells. Flow-density relationships are applied to calculate the flow at the boundary of each cell at each time step. Other macroscopic systems reported in the literature include METANET (Wang et al., 2001), EMME/2 (INRO, 2013), and Visum (PTV, 2013b).

Microscopic DTA models are at the other end of the spectrum with the highest level of fidelity yet the most computational overhead. These models have detailed network representation and they model the travel and driving behaviors of individual cars and their interactions in the road network, including but not limited to route choice, car-following, lane-changing, yielding and merging, etc. A much larger number of model parameters in these models need to be calibrated to better reflect the location specific driving behaviors, which add more complexity to the application of such models. Nevertheless, researchers and commercial companies have put extensive
Effort on the developing of microscopic models in recent decades and they have been successfully applied in a good number of networks all around the world. Examples of popular microscopic DTA systems include PARAMICS (Smith et al., 1995), MITSIM-Lab (Yang and Koutsopoulos, 1996), AIMSUN2 (Barceló and Casas, 2005), CORSIM (FHWA, 2013a), VISSIM (PTV, 2013a), and TransModeler (FHWA, 2013b).

Mesoscopic DTA models combine the concept of microscopic models and macroscopic models in order to find a balance between accuracy and performance. Such systems have a detailed network representation and model individual vehicles and their route choice behaviors in the network as microscopic models. However, the individual driving behaviors and the interaction between cars are not considered in such models. Instead, more macroscopic approaches are applied. There are various ways to implement the mesoscopic model. One way to simulate the traffic is to group vehicles into packets. Vehicles in a same packet share the same speed in a link, which is computed using link speed-density functions. An example of such modeling approach is CONTRAM (Leonard et al., 1989). Another popular approach is called queue-server. In queue-server based mesoscopic DTA systems, each road segment is divided into a moving part and a queuing part. Vehicles in the moving part move with a speed computed using segment speed-density function. Based on the capacity of the current segment and the number of vehicles that have passed within the current interval, the queue-server may start letting vehicles form a queue at the end of the current segment instead of moving to the next one. The movement of vehicles in the queuing part follows queuing theory. Examples of queue-server based mesoscopic DTA systems include DynaMIT (Ben-Akiva et al., 1997), DYNASMART (Mahmassani, 2001). Dynameq (Florian et al., 2006) is a recently developed traffic planning software package that applies run-time saving algorithms to include as much microscopic behaviors as possible while not significantly increasing the computational overhead.

Due to their capability of replicating complex congestion patterns and providing accurate estimation of network states, simulation-based DTA systems have been widely adopted in transportation planning (see, e.g., Florian et al., 2001; Ziliaskopoulos et al., 2004; Balakrishna, Wen, Ben-Akiva and Antoniou, 2008; Rathi et al., 2008;
Ben-Akiva et al., 2012). Microscopic and mesoscopic models with sophisticated models to capture the travel behaviors of drivers and their response to real time traffic information have also been deployed on-line in the context advanced traffic management systems (ATMS) and advanced traveler information systems (ATIS) to provide consistent traffic prediction and real time route guidance (see, e.g., Mahmassani, 2001; Antoniou, 2004; Wen et al., 2006; Wen, 2008).

1.2 Thesis motivation and problem statement

In network planning applications, the value of DTA models lies in their capability of assisting transportation planners to evaluate the possible impact of a physical change of network configuration (e.g., adding/closing a link), or new traffic management strategies (e.g., signal timing plan, VMS, etc.) to the existing traffic system. To achieve this goal, the first and extremely important step is to accurately replicate the traffic patterns and dynamics under current conditions for the network, without which traffic planners cannot be confident about the estimated results for hypothetical scenarios. Moreover, in real time traffic management systems, in order to provide consistent prediction of future traffic conditions and useful route guidance to drivers, DTA models have to first be able to replicate the traffic conditions accurately at and shortly before current time point. Besides careful design of model specifications, these goals are achieved by accurately estimating the values of all the parameters in the DTA model using observed traffic data. This process is usually called model calibration. The values of DTA model parameters are usually case-specific, which means that applying a same DTA system in different regions may result in very different values of model parameters. For example, the driving behavior models are very different in Beijing, where drivers change lanes aggressively, than in Boston, where lane-changing happens much less frequently and more moderately.

DTA model calibration focuses on model parameters in all the model components. In demand side, the most important parameters to be calibrated are time-dependent OD flows, travel behavior model parameters, for example, departure time, mode,
route choice, etc. In supply side, the candidate parameters include driving behavior models and interaction between individual vehicles (e.g., car-following, lane-changing, yielding and merging), segment capacities and speed-density functions, parameters related to the influence of incidents and weather, etc. The specific set of supply parameters to calibrate varies depending on the type of DTA model (microscopic, mesoscopic or macroscopic).

There are two types of DTA model calibration: off-line calibration and on-line calibration.

Off-line calibration utilizes archived historical traffic data (sensor counts, segment speeds, travel times, etc.) and a priori values of model parameters if available (obtained from other sources, e.g., planning OD matrices, behavior model parameters estimated individually from disaggregate data) to create a historical database of parameter values for the DTA model to be able to replicate the average conditions across a wide range of short-term scenarios, such as different weather conditions, incidents, special events etc. The historical database can be used directly in most planning applications that are not sensitive to short-term abnormal traffic patterns. However, longer-term factors including the type of day (weekday or weekend), the month in a year, etc., may affect the estimated results significantly even for planning purposes. Under such situations, different historical databases should be built using corresponding traffic data archive.

In real-time traffic prediction systems, more accurate short-term traffic dynamics need to be captured and the results are more sensitive to short-term factors like weather, events and incidents. Therefore, the historical database obtained from off-line calibration can only be used as the starting point, or the a priori estimation for further on-line adjustments (Figure 1-2). On-line calibration is based on a different set of methodologies using real-time traffic surveillance data to adjust the values of model parameters in real-time to better fit the current traffic condition. This thesis focuses on the framework and methodologies of off-line DTA calibration. Thorough discussions of DTA on-line calibration can be found in Antoniou (2004) and Huang (2010). Details about the relationship between off-line and on-line calibration are
introduced in Antoniou et al. (2011).

![Figure 1-2: Off-line and On-line DTA Model Calibration](image)

Traditional off-line calibration methodologies focus on separately estimating demand and supply parameters in a iterative or sequential manner. One component is estimated while the other component is considered to be fixed. This framework is able to take advantage of the extensive research efforts on demand estimation approaches and supply estimation methodologies. However, it fails to capture the important interactions between demand and supply and potentially have heavy computational overhead. To solve this problem, Balakrishna (2006) proposes a simultaneous demand-supply calibration framework. In this framework, the off-line calibration problem is formulated as an optimization problem over a vector of DTA model parameters. The objective function is the weighted sum of two types of deviations: 1) deviations between observed traffic measurements and simulated traffic measurements, and 2) deviations between a priori parameter values and estimated parameter values. A more rigorous problem formulation can be found in Chapter 3. There have been a number of candidate solution algorithms for the simultaneous demand-supply off-line calibration problem, including Box-Complex (Box, 1965), stable noisy optimization by branch and fit (SNOBFIT) (Huyer and Neumaier, 2008), genetic algorithms (GA) (Goldberg and Holland, 1988), and simultaneous perturbation stochastic approxima-
tion (SPSA) (Spall, 1998). Through experiments and case studies, SPSA has been shown to be the most suitable solution algorithm due to its efficiency.

Although SPSA have been applied successfully in a great number of studies (e.g., Balakrishna (2006); Ma et al. (2007); Vaze et al. (2009); Paz et al. (2012)), it is found that the good performance of SPSA deteriorates significantly when the size of the target network and the total number of simulation intervals increase. This has motivated the attempts to design and implement a better scalable algorithm that can achieve satisfactory accuracy in large-scale multi-interval applications. In this thesis, the reasons for the performance deterioration of SPSA are explored and the shortcomings of applying the SPSA algorithm in the off-line calibration framework are identified. An enhanced algorithm - Weighted SPSA, or W-SPSA - that combines the idea of SPSA with a more accurate gradient approximation method is proposed and tested through synthetic tests and a real world case study. This new algorithm is proved to outperform SPSA significantly in large-scale problems.

1.3 Thesis outline

The remainder of this thesis is organized as follows. Chapter 2 presents an in-depth review of different DTA calibration frameworks. In Chapter 3, rigorous formulations of the DTA off-line calibration problem using the state-of-the-art simultaneous framework are first presented. Then, the SPSA algorithm is reviewed in detail, followed by analyses of its limitations. Finally, an enhanced algorithm: W-SPSA is proposed and approaches to estimate the key element in the algorithm, the weight matrix, is discussed. Chapter 4 demonstrates the performance comparison between SPSA and W-SPSA, and explores the characteristics of W-SPSA through synthetic tests. In Chapter 5, a case study of the entire Singapore expressway system is conducted to demonstrate the performance advantage and technical details of W-SPSA in the context of real world large-scale networks. Chapter 6 concludes the thesis.
Chapter 2

Literature Review

This chapter reviews the systematic off-line calibration of simulation-based DTA systems. Typically, DTA systems consist of two components: 1) the demand models that mainly focus on how travelers make trip decisions (origin-destination, departure time, mode, route, etc.), and 2) the supply models that load traffic into the network, model the movement patterns of vehicles and the formation, dissipation and spill-back of queues. The DTA off-line calibration literature are categorized into three groups based on the model components they calibrate: 1) estimation of demand models, 2) estimation of supply models, and 3) demand-supply joint calibration.

Extensive works have been done on the separate estimation of OD flows, behavior model parameters, and supply model parameters using both disaggregate and aggregate traffic field data. This thesis, however, focuses on the joint demand-supply calibration framework, which has received more and more attention recently due to its capability of capturing the complex interaction between demand and supply. The first two sections in this chapter provide an overview of literature related to the estimation of demand models, followed by works related to the estimation of supply models. A more in-depth review of the studies related to the demand-supply joint calibration is then presented in the third section.
2.1 Estimation of demand models

There are two major parts in DTA demand models: travel behavior models and time-dependent OD flows. The remainder of this section first illustrates the techniques in the estimation of travel behavior models with an example of route choice. Then the methodologies for aggregate OD estimation are reviewed.

2.1.1 Travel behavior models

At the disaggregate level, individual travelers' behaviors are modeled by mode, departure time and route choice models for both pre-trip and en-route. Pre-trip decisions include travelers' choice of mode, departure time, as well as route under the expected or perceived traffic conditions before taking the trip, while en-route decisions are made by travelers based on real time information and traffic condition during the trip.

With the abundance of disaggregate data with higher accuracy and coverage, the estimation of high quality travel behavior models becomes possible. The route choice model, which is among the core components of DTA demand models, is a potential beneficiary of such data and a good example to illustrate the techniques used in the estimation of most travel behavior models. Route choice captures drivers' preference in choosing a route from an origin to a destination in a road network and it is modeled using discrete choice analysis (Ben-Akiva and Lerman, 1985; Train, 2003), where the driver selects one single route from a set of alternatives (i.e., the "choice set"). In a real network, there is a huge number of possible routes between an OD pair. For computational tractability, different choice set generation algorithms are developed to build a smaller subset of routes that includes as much as possible the routes actually chosen by drivers in reality while eliminating routes that are never selected. Such algorithms can be classified into two groups: 1) deterministic algorithms such as link elimination (Azevedo et al., 1993), link penalty (de la Barra et al., 1993), labeling (Ben-Akiva et al., 1984), and 2) stochastic algorithms such as simulation (Ramming, 2002), doubly stochastic choice set generation (Bovy and Fiorenzo-Catalano, 2006).
Once the choice set is generated, a route choice model can be developed. In the model, each driver is described by a vector of characteristics and each alternative route in the choice set is described by a vector of attributes. Each driver perceives a utility associated with each route, which is a real number mapped from the characteristics of the driver and the attributes of the route. Discrete choice theory assumes that each individual driver will select the route that has the maximum perceived utility. In reality, however, the utilities are not directly measured. Random utility theory captures the discrepancy between the systematic model utilities and the "true" utilities using an error term, where the systematic utility is calculated as a linear function of the characteristics of a driver and the attributes of an alternative route. Different models assume different distributions of the error term. The Multinomial Logit (MNL) model is one of the most attractive models in real applications due to its simple assumption on identically and independently distributed Gumbel errors and its closed-form formula to compute the probability of selecting an alternative in the choice set. However, in route choice applications where different paths have a relatively high degree of overlapping, MNL fails to reproduce the route choice behavior of drivers accurately and the C-Logit model (Cascetta et al., 1996) and Path Size Logit model (Ben-Akiva and Bierlaire, 1999) are proposed. Path Size Logit model has been successfully implemented in a simulation based DTA system to model a congested urban area in the city of Beijing (Ben-Akiva et al., 2012).

Disaggregate data that reveal individuals' route choice behaviors are used to estimate the parameters in route choice models. The sources of data include traditional surveys, such as mail, telephone, and the Internet (Ben-Akiva et al., 1984; Prato, 2004), or emerging technologies, such as GPS trajectories (Frejinger, 2007). The parameters to estimate are the coefficients in the systematic utility function and they are estimated by maximizing the joint probability of the actual selected route in the choice set. More rigorous mathematical derivations in discrete choice analysis can be found in Ben-Akiva and Lerman (1985).

Disaggregate estimation of route choice models has the advantage of reflecting actual behaviors of individual drivers. However, it also has limitations due to sam-
pling issues and high cost of data gathering (with advance in information technology the cost is reducing significantly, though. e.g., the Future Mobility Survey project (Cottrill et al., 2013)). Ashok (1996) and Toledo, Ben-Akiva, Darda, Jha and Koutsopoulos (2004) made the first attempts to improve the route choice estimation using aggregate data, which has begun to receive attention.

2.1.2 Estimation of OD flows

At the aggregate level, the OD estimation problem focuses on estimating a matrix of point-to-point trip demand using observed aggregate traffic flow data collected at specific links in the network. The number of rows in the matrix equals the number of different trip origins and the number of columns equals the number of different trip destinations. Each element in the matrix specifies the number of trips from an origin to a destination. Accurate OD matrices are fundamental inputs for DTA models in both real-time and planning applications and therefore the estimation methodologies have received significant attention in the literature.

The OD estimation studies can be divided into two groups. First there is static OD estimation, where the OD flows are assumed to be static across a relatively long time period (a day, a morning/evening peak, for example). The OD flows are estimated using a single set of link counts, which is usually the average flows across the entire study period. Example studies on static OD estimation include Cascetta (1984), Hazelton (2000), and Li (2005). However, DTA systems require dynamic demand input to model the time-dependent traffic conditions that may change significantly across different time periods. Therefore, the focus of this section is on the approaches to estimate dynamic OD flows.

The most widely applied dynamic OD flow estimation approach is the generalized least square (GLS) framework proposed in Cascetta et al. (1993). The OD estimation problem is expressed through a fixed-point model and a bi-level programming optimization method is proposed to solve it. At the upper level, the OD matrix is estimated by minimizing the sum of two parts: 1) the distance between the estimated OD demands and their a priori values, and 2) the distance between the observed
sensor counts from field traffic data and the fitted counts that are obtained by assigning the estimated OD demand to the network. Considering the distance between estimated values and a priori values is important, because in real applications the number of link sensor counts is usually much smaller than the number of OD flows. Failure to consider a priori information will result in an indeterminate problem. At the lower level, an assignment matrix is estimated assuming that the OD matrix is fixed. The assignment matrix is a linear approximation of the relationship between OD flows and sensor counts that maps OD flows into flows at different sensor locations (links). It is used at the upper level to generate fitted flows. The authors propose two estimators under the GLS framework. The sequential estimator solves for the OD flows one interval at a time and the simultaneous estimator optimizes the OD flows in multiple intervals simultaneously. Although the simultaneous estimator is able to capture the influence of the OD flow in one interval on all subsequent intervals, it requires the calculation and storage of a large assignment matrix and has been found to have significant computational overhead on large networks in Toledo et al. (2003) and Bierlaire and Crittin (2004).

Ashok (1996) formulates the OD flow estimation problem with a state-space representation, where the interval-to-interval evolution of network states is captured by transition equations (autoregressive process) and the sensor counts are incorporated using measurement equations that links states with measurements. OD flows themselves do not have symmetrical distributions and therefore do not possess satisfactory estimation properties in the state-space framework. To solve this problem, the author innovatively defines the state as the difference between OD flows and their historical values, i.e., deviations. Then, the author further proposes a Kalman Filter solution approach for the estimation and discusses the methods to obtain the initial inputs.

In both GLS and state-space framework, the assignment matrix is a key element for OD flow estimation (at the lower level of GLS and in the measurement equations of state-space). There are two approaches to obtain assignment matrices outlined in Ashok (1996): 1) load the current best OD flows into a traffic simulator, keep tracking the vehicles from different OD pairs, record and count their appearances at
sensor locations, and 2) analytically calculate the fractions using knowledge of network topology, time-dependent travel times as well as route choice models. Balakrishna (2006) argues that the latter approach can provide more accurate results because the simulation based method requires small starting flows to prevent artificial bottlenecks and therefore results in highly stochastic and small fractions due to the limited number of vehicles being loaded into the network.

It has been widely recognized that, in congested conditions, the linear assumption on the assignment matrix fails to represent the complex relationship between OD flows and sensor counts, when increasing the input demand results in more congestion and less link flow. At the same time, practitioners start to apply DTA models in large-scale, highly congested urban networks, which requires more computational efficient OD flow estimation methodologies. Great efforts have been made to address these problems in the past decade. As a result, researchers have proposed a good number of novel frameworks. Nie and Zhang (2008) formulate the OD estimation problem as a single-level framework based on variational inequality. Travel time is incorporated into this framework by Qian and Zhang (2011) to further improve its performance. Djukic et al. (2012) apply principal component analysis (PCA) to preprocess OD data. This methodology can significantly reduce the size of time series of OD demand without sacrificing much of the accuracy, which lead to a impressive reduction of computational costs. Frederix et al. (2011) introduce the use of marginal computation (MaC) and the use of kinematic wave theory principles to estimate the often non-linear relationships between OD flows and link flows under congested conditions. MaC efficiently estimates the relationships by perturbing the OD flows and calculate the sensitivity of link flows to the perturbation. The derived relationships were used in the dynamic OD estimation problem and showed its advantage over traditional OD estimation approaches that rely on linear assignment matrices. There are also attempts to formulate the OD estimation problem as a stochastic optimization problem and use meta-heuristic approaches as the solution algorithms. For example, Stathopoulos and Tsekeris (2004) experiment with three different meta-heuristic optimization algorithms: a genetic algorithm (GA), a simulated annealing algorithm
(SA), and a hybrid algorithm (GASA) based on GA and SA. The authors argue that GASA outperforms the other two algorithms in terms of convergence rate and final accuracy; Kattan and Abdulhai (2006) apply evolutionary algorithms (EA) in a parallel computing framework to improve the computation efficiency as well as the solution quality. Balakrishna, Ben-Akiva and Koutsopoulos (2008); Cipriani et al. (2011) proposes to apply an efficient gradient approximation method, simultaneous perturbation stochastic approximation (SPSA) that significantly reduces the number of runs to estimate numerical derivatives when conducting path search.

The evolution of OD demand from day to day is also studied using multiple day sensor counts. Zhou et al. (2003) propose an approach extended from the traditional single day OD flow estimation framework, which is based on bi-level optimization. Moreover, adaptive weights on the two types of derivations in the GLS framework are determined by an interactive approach to achieve the best results. Hazelton (2008) views the problem in a statistical perspective with a multivariate model for the link counts. The author describes the day-to-day evolution of OD demands with a modest number of parameters and proposes a Bayesian approach to inference.

### 2.2 Estimation of supply models

As discussed in Chapter 1, simulation-based DTA systems can be categorized into three groups based on their level of fidelity with which they represent the traffic system: Macroscopic, Mesoscopic and Microscopic. Macroscopic/microscopic DTA systems have macroscopic/microscopic demand and supply models, while mesoscopic systems have microscopic demand models and macroscopic supply models. Therefore, the DTA supply estimation can be classified into two categories: 1) the estimation of macroscopic supply models, and 2) the estimation of microscopic supply models.

The macroscopic supply estimation mainly focuses on capacities and link-based speed-density models.

Capacities are mostly computed using observed flows and the number of lanes. For example, in the approach proposed by Mufioz et al. (2004) to estimate the pa-
parameters in a Cell Transmission Model (CTM), capacities for non-bottleneck cells are chosen to be slightly larger than the maximum observed flows while capacities for bottleneck cells are computed to match the observed mainline and ramp flows. The *Highway Capacity Manual* has also been widely used as a reference for determining the capacities for different types of links in DTA models.

Link based functions that describe the relationships between speeds, densities and flows are crucial components for macroscopic and mesoscopic supply models. These functions are estimated by fitting appropriate curves to observed traffic data. Due to the usually low level of sensor coverage rate in reality, Van Aerde and Rakha (1995) propose an approach to group links in the study network based on their characteristics. Different supply functions for each group are estimated using observed data from group-specific sensors. This approach has been applied widely in DTA applications.

Fitting speed-density functions only to data from sensors that are local to the target links potentially leads to over-fitting and inconsistency across contiguous links. Moreover, the localized fits do not necessarily reflect the overall traffic dynamics in the entire network. As a result, the DTA model will have poor prediction power under scenarios that are different from the one under which the data are collected. An attempt to solve this problem is made by Kunde (2002). The author outlines a three-stage approach to supply calibration. The first stage is to estimate the supply parameters using traditional methods with local data. Then, sub-networks are chosen to refine the estimations by incorporating interactions between segments within each subnetwork. Finally the demand-supply interactions are considered within the entire study network. A case study using data from Irvine, California, is used to demonstrate the methodology.

While most of the flow-density relationship estimation literature focus on methodologies and applications for expressway corridors (see, e.g., Wang et al. (2001); Ngo-duy et al. (2006)), Leclercq (2005) develops methods to address the specificity of traffic behavior on urban streets (Toulouse, France). A two-step approach is proposed. Firstly the data are processed and aggregated to make them suitable for equilibrium traffic state representation. Then, the flow-density relationship is esti-
mated by fitting to the processed data using an interior point, conjugate gradient optimization method.

Chiu et al. (2009) propose a new mesoscopic model: the vehicle-based anisotropic mesoscopic simulation (AMS) model. In this model, individual vehicles have their own prevailing speed, rather than sharing a same speed with vehicles in the same segment. Vehicle trajectory data from the Next Generation Simulation (NGSIM) program was used in the calibration process.

The microscopic supply calibration mainly focuses on parameters in driving behavior models and microscopic network supply parameters (e.g., signal timing). Although driving behavior models are usually categorized as demand models, they are sometimes calibrated in the supply calibration due to their close relationship with the movement of vehicles in the network. Beyond the early attempts on manual adjustment and heuristics, a wide range of system optimization methods are developed. Darda (2002) applies the Box-Complex (Box, 1965) algorithm to calibrate car-following and lane-changing models in MITSIMLab given fixed demand model parameters. However, convergence is not able to be ascertained in this study. Brockfeld et al. (2005) calibrate a small number of supply parameters in a variety of microscopic and macroscopic DTA systems using a gradient-free downhill simplex algorithm. Kim and Rilett (2004) applies genetic algorithms (GA) to calibrate driving behavior parameters in CORSIM and TRANSIMS. However, given the small scale of the problem in their study, there are still computational constraints. More studies on applying these methods on real-world scale networks should be done to test their efficiency and ability to handle large-scale problems.

2.3 Demand-supply joint calibration

As reviewed in previous sections, a great number of approaches have been developed and successfully applied for the systematic calibration of demand and supply components in DTA models. However, researchers have recently started to realize that the independent calibration of demand/supply while fixing the other one may not be
efficient nor optimal due to its inability to capture the interaction between the two components. To solve this problem, demand-supply joint calibration frameworks have been proposed, which can be grouped into two categories: 1) iterative demand-supply joint calibration, and 2) simultaneous demand-supply joint calibration. The second framework outperforms the first one and becomes the state-of-the-art in DTA off-line calibration. This section firstly reviews the iterative demand-supply calibration framework, followed by the simultaneous efforts.

2.3.1 Iterative demand-supply calibration

The iterative demand-supply calibration framework combines the independent demand and supply estimation into a consistent framework. It attempts to consider the interaction between the two DTA components by iteratively updating them during the calibration process.

The general framework of iterative demand-supply calibration (shown in Figure 2-1) usually consists of two parts that are executed sequentially: disaggregate estimation and aggregate calibration. In the disaggregate estimation part, initial parameter values in behavior models are estimated using disaggregate data. For example, using GPS trajectories of individual cars to estimate parameters in the route choice model, etc. Disaggregate validation and model refinement are done iteratively within this part. After getting the initial parameter values for behavior models, in the second part, demand calibration and supply calibration are done iteratively using aggregate traffic data until reaching convergence. In demand calibration, the parameter values in the supply model are fixed when OD flows and parameters in behavior models are calibrated using aggregate traffic data, for example, sensor counts, travel times, etc. Then, using the calibrated demand parameters, supply models are calibrated and simulated traffic conditions that are used in the calibration of demand models are updated, for example, the habitual link travel times that are used in the route choice models. If convergence is not reached, the demand calibration is done again using the updated supply model parameters and simulated traffic conditions, followed by the re-calibration of the supply components.
Figure 2-1: Iterative demand-supply calibration framework
Mahut et al. (2004) present an iterative demand-supply calibration approach of a simulation-based DTA model, DTASQ. The model was applied on a portion of the City of Calgary road network in Alberta, Canada. In demand side, the OD flows were estimated from turning movement counts. In supply side, the model has more in common with microscopic models. It simulates the driving behavior of individual cars with a small number of parameters (e.g., car-following, gap acceptance, lane-changing model parameters, etc.). To establish dynamic user equilibrium travel times, the demand model iteratively re-assign path flows to the network based on travel times obtained from the supply model. The parameters in supply model, together with other global parameters, were adjusted manually to fit a set of one-hour turning counts. The calibrated model was validated using an independent set of 15-minute turning counts and the results showed high accuracy. The author claims that this work is the first calibration exercise of an iterative simulation-based DTA. However, the success for the Calgary network does not ascertain the transferability of the approach as the parameters were adjusted manually.

Gupta (2005) present an iterative off-line calibration method for calibration of demand and supply models in DynaMIT. The demand calibration consists of the estimation of behavior models (e.g., route choice model) and OD flows. Behavior model parameters are estimated through manual line or grid searches. OD flows are estimated using a Generalized Least Square (GLS) approach that relies on assignment matrices. These two parts in the demand model are estimated iteratively by fixing the other part. In supply side, speed density parameters are estimated by fitting curves to sensor data. Segments are grouped based on their characteristics and speed-density functions are estimated for each group. Segment capacities are decided based on Highway Capacity Manual. This framework enables the iterative updating of behavior models and OD flows. However, it fails to capture the interaction between demand and supply, because the estimation of supply models are independent from the demand model estimation. On the other hand, a significant contribution of this work is that the author develops an observability test, which verifies if unique OD flows can be estimated from a given sensor configuration.
Similar frameworks have been applied by other researchers. Toledo, Koutsopoulos, Davol, M. E., Burghout, Andreasson, Johansson and Lundin. (2004) integrated this bi-level iterative methodology in the microscopic simulator MITSIMLab and applied to a network in Stockholm, Sweden. Kim (2006) applied genetic algorithm and disaggregate data to estimate behavior model parameters at the upper level and estimated OD flows using the extended Kalman filter (EKF) at the lower level. The methodology was evaluated using VISSIM software package.

Chi et al. (2010) present a bi-level framework for the estimation of OD flows in the mesoscopic simulation-based DTA system DYNASMART-P. Different from the independent estimation of OD flows, this framework iterates between the dynamic OD estimation process and the traffic flow model (supply) fine tuning process. The authors emphasize the importance of the fine tuning of traffic flow parameters after the initial estimation. The fine tuning problem is formulated to minimize the deviation between observed and simulated measurements (flow and speed data are used in this study). Stochastic algorithm SPSA is applied to solve this problem. A case study in a section of northbound highway I-95 in Miami, Dade County, Florida was conducted. The fine tuning of traffic flow model parameters lead to a remarkable reduction in errors of speed but with little contribution to fit-to-counts. The calibration approach for the behavioral model parameters were not considered in the study.

2.3.2 Simultaneous demand-supply calibration

Recently, DTA off-line calibration frameworks that are capable of simultaneously calibrating demand and supply have received significant attention due to their ability to fully consider the interactions between demand models and supply models as well as their huge savings in computation time. In these frameworks, the DTA calibration problem is formulated as an optimization problem. The objective function, which evaluates the goodness-of-fit of simulated traffic quantities to their observed values (e.g., flow, speed, travel time, etc.), is minimized during the calibration process by adjusting the parameters in both the demand component and the supply component of the DTA system.
This stochastic optimization framework was first proposed by Balakrishna (2006). The author applied this simultaneous demand-supply calibration methodology in DynaMIT, the mesoscopic simulation-based DTA system. Parameters considered in the decision vector included time-dependent OD flows, coefficients in travel behavior models, coefficients in speed-density functions, and segment capacities. Goodness-of-fit to sensor flows and segment speed data are used in the objective function. As in most cases, the number of measurements is far less than the number of parameters to estimate, the distances between estimated parameter values and their a priori, or historical values are also included as a part of the objective function. In this work, three algorithms were tested as the solution algorithm of the stochastic optimization problem: the Box-Complex (Box, 1965), stable noisy optimization by branch and fit (SNOBFIT) (Huyer and Neumaier, 2008), and simultaneous perturbation stochastic approximation (SPSA) (Spall, 1998). A synthetic case study and a full scale real world case study in Los Angeles, California were used to test the simultaneous demand-supply calibration framework as well as the three different solution algorithms. The results showed that SPSA was the most suitable algorithm due to its extremely high efficiency and satisfactory convergence performance. Moreover, the results also indicated that this simultaneous demand-supply framework outperformed the traditional iterative framework, which was used as the reference case.

Balakrishna et al. (2007) extended this methodology to the off-line calibration of microscopic traffic simulation models by applying it in MITSIMLab. SPSA was used as the solution algorithm. A case study in network of Lower Westchester County, New York was used to demonstrate the advantage of this methodology over the iterative framework. However, this network has a very high coverage rate of sensors, which is uncommon in real-world applications.

Other than traditional aggregate traffic data, such as loop detector counts, emerging traffic surveillance devices provide a rich source of disaggregate traffic data. For example, Automatic Vehicle Identification (AVI) technology provides point-to-point traffic data that has great potential to improve the calibration of DTA systems. The use of AVI data was limited to OD estimation problem. Vaze et al. (2009) first in-
corporated AVI point-to-point travel times in the off-line calibration of DynaMIT using the simultaneous framework that estimate demand and supply at the same time. SPSA, GA, and a Monte-Carlo simulation based technique (Particle Filter) were tested as the solution algorithm. SPSA and GA outperformed Particle Filter in a small network and SPSA was found to be scalable and therefore suitable for larger problems. Results indicate that using both traditional sensor counts and AVI data to do the calibration significantly improved the accuracy of the model. Again, the simultaneous approach outperformed the sequential approach in this study. The key contribution of this study is that it demonstrates the extendibility of the framework to incorporate any kind of traffic data, aggregate or disaggregate, from any source.

Appiah and Rilett (2010) presents a framework that jointly calibrates OD flows and driving behaviors in microscopic supply simulators using aggregate intersection turning movement counts. With the vehicle trajectory information inherent in intersection turning counts, this methodology does not require the availability of historical OD matrices. Similar to Balakrishna (2006), the framework formulates the calibration as an optimization problem. However, a GA was adopted as the solution algorithm. A case study in a small arterial network using VISSIM, a microscopic simulation software package was conducted in this study. The results showed appropriate final accuracy but the GA algorithm showed high computational cost as it took the algorithm 2 months and 18,000 iterations to converge, given the small scale of the problem.

The previous studies demonstrated the framework only on networks with relatively small scale or simple structure (for example, expressway corridor). Ben-Akiva et al. (2012) applied the simultaneous calibration methodology to a large-scale highly congested urban network in the city of Beijing. The network consists of 1,698 nodes connected by 3,180 directed links in an area of approximately 35 square miles. There are as many as 3000 OD pairs in the study network. Time dependent OD flows, route choice parameters, speed-density relationships and segment capacities were calibrated. Sensor counts in expressways and travel times from GPS equipped floating cars were used as measurements in the calibration. The results showed significant
improvements in terms of goodness-of-fit to observed quantities comparing to their initial values. However, these improvements came from both model refinement (e.g., more sophisticated route choice model specification, special treatment to short links in supply model, etc.) and systematic calibration. It is unknown how much the systematic joint calibration contributed to the improvement.

2.4 Summary

In the past decades, extensive works have been done in the field of separately estimating either the demand models or the supply models in DTA systems. The theoretical significance of such problems as well as the challenges have kept this field extremely active over time, especially the estimation of dynamic origin-destination flows. However, to implement a DTA system in a real world network, joint calibration of demand and supply is crucial for the system to be able to accurately replicate the traffic dynamics and provide consist predictions. Researchers and practitioners have realized the importance of joint calibration frameworks. Attempts have first been made to iteratively calibrate demand and supply models in DTA systems while fixing the other part. Despite their successful applications, the sequential nature of such frameworks makes them computationally inefficient and unable to capture the complex interactions between demand and supply. Recently, simultaneous demand-supply calibration frameworks have been proposed and received attention. They are flexible to incorporate different types of data and can be applied to different types of DTA systems (mesoscopic, microscopic). Experiments show that the simultaneous demand-supply calibration framework lead to better results than the iterative framework.

In the simultaneous demand-supply calibration frameworks, the DTA calibration problem is formulated as a stochastic optimization problem. Different solution algorithms have been tested and SPSA outperformed others in most of the cases. Researchers argue that SPSA can reach similar accuracy as other algorithms do in small networks while being more scalable in terms of computing time. Scalability makes SPSA an ideal algorithm for large-scale problems in terms of computational cost sav-
ing. However, no direct evidence has shown that with the increase of problem scale, SPSA will reach results with the same good quality. In the following chapter, the shortcomings of SPSA in the context of simultaneous DTA off-line calibration are discussed and an improved solution algorithm is proposed with further analyses and applications in later chapters.
Chapter 3

Methodology

This chapter begins with rigorous formulations of the DTA off-line calibration problem under the simultaneous demand-supply framework. An in-depth review of the SPSA algorithm and its applications in DTA off-line calibration is presented next. Then, the shortcomings of applying SPSA in the DTA off-line calibration problem are illustrated with a simple example network. Finally, an enhanced algorithm, Weighted SPSA, or W-SPSA, is introduced and the ways to estimate the key element of this algorithm, the weight matrix, are discussed.

3.1 General problem formulation

Let the time period of interest be divided into intervals \( \mathcal{H} = \{1, 2, ..., H\} \). The DTA off-line calibration problem is formulated using the following notations:

- \( \boldsymbol{x} \): time-dependent DTA parameters, e.g., OD flows, \( \boldsymbol{x} = \{x_h\}, \forall h \in \mathcal{H} \)
- \( \boldsymbol{\beta} \): other DTA parameters, e.g., supply model parameters
- \( M^o \): Observed aggregate time-dependent sensor measurements, \( M^o = \{M^o_h\} \)
- \( M^s \): Simulated aggregate time-dependent sensor measurements, \( M^s = \{M^s_h\} \)
- \( \boldsymbol{x}^a \): A priori time-dependent parameter values, \( \boldsymbol{x}^a = \{x^a_h\} \)
- $\beta^a$: A priori values of other DTA parameters

- $G$: Road network, $G = \{G_h\}$

The off-line calibration problem is formulated as an optimization problem to minimize an objective function over the parameter space:

$$
\text{minimize}_{x, \beta} z(M^o, M^s, x, \beta, x^o, \beta^a) = \sum_{h=1}^{H} \left[ z_1(M^o_h, M^s_h) + z_2(x_h, x^o_h) \right] + z_3(\beta, \beta^a)
$$  

(3.1)

subject to:

$$M^o_h = f(x_1, ..., x_h; \beta; G_1, ..., G_h)$$  

(3.2)

$$l_{x_h} \leq x_h \leq u_{x_h}$$  

(3.3)

$$l_{\beta} \leq \beta \leq u_{\beta}$$  

(3.4)

Equation 3.1 is the objective function for the minimization problem, where $z_1$ measures the goodness-of-fit between observed sensor measurements and simulated sensor measurements, and $z_2$ as well as $z_3$ measures the distance between estimated values and a priori values. Equation 3.2 represents the relationship between simulated measurement values and model inputs. $f$ is the DTA system. Equations 3.3 and 3.4 specify the boundaries for the parameter values to estimate.

$z_1$, $z_2$ and $z_3$ depend on the specific goodness-of-fit measures chosen. Under the generalized least squares framework, the objective function in equation 3.1 becomes:

$$
\text{minimize}_{x, \beta} z(M^o, M^s, x, \beta, x^o, \beta^a) = \sum_{h=1}^{H} \left[ \epsilon'_M \Omega^{-1}_M \epsilon_M + \epsilon'_{x_h} \Omega^{-1}_{x_h} \epsilon_{x_h} \right] + \epsilon'_\beta \Omega^{-1}_\beta \epsilon_\beta
$$  

(3.5)

42
where

$$
\begin{align*}
\epsilon_{M_h} &= M_h^0 - M_h^s \\
\epsilon_{x_h} &= x_h - x_h^a \\
\epsilon_\beta &= \beta - \beta^a
\end{align*}
$$

and $\Omega_{M_h}, \Omega_{x_h}, \Omega_\beta$ are variance-covariance matrices.

The subscripts of intervals (i.e., $h$) highlight the time dependent feature of the DTA calibration problem. It provides the intuition that the problem is a dynamic traffic assignment problem. However, it’s unnecessary when we formulate the problem in a more general way. For a more straight-forward illustration of the solution algorithm, we further generalize the framework by removing the interval subscripts. The more generalized notations are defined as:

$$
\begin{align*}
\theta &= \begin{bmatrix} x_1 \\ \vdots \\ x_H \\ \beta \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_P \end{bmatrix} \\
\epsilon_M &= \begin{bmatrix} M_1^0 - M_1^s \\ \vdots \\ M_H^0 - M_H^s \end{bmatrix} = \begin{bmatrix} M_1^0 - M_1^s \\ \vdots \\ M_D^0 - M_D^s \end{bmatrix} = \begin{bmatrix} \epsilon_{M1} \\ \vdots \\ \epsilon_{MD} \end{bmatrix} = F_M(\theta; M^0; G) \\
\epsilon_\theta &= \begin{bmatrix} \theta_1 - \theta_1^a \\ \vdots \\ \theta_P - \theta_P^a \end{bmatrix} = \begin{bmatrix} \epsilon_{\theta1} \\ \vdots \\ \epsilon_{\thetaP} \end{bmatrix} = F_\theta(\theta; \theta^a; G)
\end{align*}
$$

$\epsilon_M$ is a vector of deviations between observed measurement values and simulated measurement values. $F$ is a function that maps $\theta$, $M^0$ and the road network $G$ to $\epsilon_M$. $\epsilon_\theta$ is a vector of deviations between a priori parameter values and estimated parameter values. In simulation based DTA systems, $F$ is the simulator and related
deviation calculations, where $\theta^a$ is a vector of the a priori values of the elements in $\theta$. The length of $\theta$ and $\epsilon_\theta, P$, equals the total number of parameters to calibrate (one OD flow in two different intervals are considered as two different parameters). The length of $\epsilon_M, D$, equals the total number of measurements.

The further generalized problem formulation is:

$$\begin{align*}
\text{minimize} \quad & z(\theta) = \epsilon_M' \Omega_M^{-1} \epsilon_M + \epsilon_\theta' \Omega_\theta^{-1} \epsilon_\theta \\
\text{subject to:} \quad & \epsilon_M = F_M(\theta; M^0; G) \\
& \epsilon_\theta = F_\theta(\theta; \theta^a; G) \\
& l_\theta \leq \theta \leq u_\theta
\end{align*}$$

(3.12)

To make the objective function in 3.12 identical as the objective function in 3.5, $\Omega_M$ should be a block diagonal matrix with $\Omega_{M_h}$ as its diagonal elements. The same applies to $\Omega_\theta$, which is a block diagonal matrix with $\Omega_{x_h}$ and $\Omega_{\beta}$ lie at its diagonal. However, in a more general formulation that consider all the correlations across different time intervals and different types of parameters, $\Omega_M$ and $\Omega_\theta$ are general variance-covariance matrices.

In next section, a widely applied solution algorithm, SPSA, is reviewed in detail under this problem formulation.

### 3.2 The SPSA algorithm

Stochastic approximation (SA) methods are a family of iterative stochastic optimization algorithms used in error function minimization when the objective function has no known analytical form and can only be estimated with noisy observations. It iteratively traces a sequence of parameter estimates that converge to zero of the objective
function’s gradient.

The general iterative form of SA is:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k)$$  \hspace{1cm} (3.16)

where $\hat{\theta}_k$ is the estimate of the decision vector in the $k_{th}$ iteration of the algorithm and $\hat{g}_k(\hat{\theta}_k)$ is the estimated gradient at $\hat{\theta}_k$. $a_k$ is a usually small number that gets smaller as $k$ becomes larger, known as the $k_{th}$ step size in a gain sequence:

$$a_k = \frac{a}{(A + k + a)^\alpha}$$  \hspace{1cm} (3.17)

where $a$, $\alpha$ and $A$ are algorithm parameters.

Different approaches have been proposed to approximate the gradient. In the finite-difference (FD) schemes, the gradient is estimated by perturbing the parameters in the decision vector one at a time, evaluating the object function values and computing the gradient as:

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{z(\hat{\theta}_k + c_k e_i) - z(\hat{\theta}_k - c_k e_i)}{2c_k}$$  \hspace{1cm} (3.18)

where $\hat{g}_{ki}(\hat{\theta}_k)$ is the $i_{th}$ element in the gradient vector $\hat{g}_k(\hat{\theta}_k)$. $e_i$ is a vector with the length equal to that of the decision vector and with a 1 at $i_{th}$ location (zeros elsewhere). Each parameter is perturbed with an amplitude of $c_k$ to two opposite directions.

$$c_k = \frac{c}{(k + 1)^\gamma}$$  \hspace{1cm} (3.19)

where $c$ is the and $\gamma$ are algorithm parameters.

This approach is capable of estimating high quality gradient vectors in non-analytical problems with noisy observations. It is, however, not efficient for large-scale problems. The number of objective function evaluations within one algorithm iteration is $2P$, where $P$ is the total number of parameters in the decision vector. In
simulation-based DTA systems, one objective function evaluation involves the run-
ing of the simulation from beginning to end, which may take minutes. A mid-sized
DTA model usually consists of thousands of model parameters to calibrate, which
lead to days of run time just for one iteration of the algorithm. Therefore, applying
FDSA in DTA off-line calibration problems is not feasible in terms of computational
overhead.

Spall (1992, 1994, 1998, 1999) proposes an innovative solution to this problem:
simultaneous perturbation stochastic approximation (SPSA). SPSA efficiently esti-
mates the gradient by perturbing all the parameters in the decision vector \( \theta \) simulta-
taneously and the approximation of gradient needs only two function evaluations
regardless of the number of parameters:

\[
\hat{g}_{ki}(\hat{\theta}_k) = \frac{z(\hat{\theta}_k + c_k \Delta_k) - z(\hat{\theta}_k - c_k \Delta_k)}{2c_k \Delta_{ki}}
\]  (3.20)

where \( \hat{g}_{ki}(\hat{\theta}_k) \) is the \( i_{th} \) element in the gradient vector \( \hat{g}_k(\hat{\theta}_k) \). \( c_k \) is the perturbation
amplitude (same as in the FD approach). \( \Delta_k \) is a random perturbation vector,
generated through a Bernoulli process (or using other appropriate distributions) with
values of +1 and -1 with equal probabilities.

SPSA provides a huge saving of computational time due to its constant number of
perturbations for the gradient approximation. In terms of convergence performance,
Spall (1998) argues that SPSA follows a path that is expected to deviate only slightly
from that of FDSA. In other words, SPSA performs as good as FDSA (Figure 3-1)
while having a P-fold time saving. With no or little noise, FDSA is expected to follow
the true descent to the optimal. SPSA may have approximated gradients that differ
from the true gradients, but they are almost unbiased. With larger noise, neither
FDSA and SPSA follow the deepest decent directions but SPSA stays in a path close
to the optimal.

The detailed step-by-step SPSA workflow is described below:

1. Set the current step \( k = 0 \), so that the initial values in the decision vector
   \( \hat{\theta}_k = \hat{\theta}_0 \) (usually the historical values or the most recent calibrated values are
used. Decide the values of the algorithm parameters $\alpha$, $\gamma$, $a$, $A$, and $c$ based on the characteristics of the specific problem.

2. Evaluate the initial objective function value $z_0$ by running the simulator with $\hat{\theta}_0$ as the input parameters.

3. Update $k = k + 1$. Calculate $a_k$ and $c_k$ based on equation 3.17 and equation 3.19.

4. Generate the independent random perturbation vector $\Delta_k$.

5. Evaluate the objective function values at two points: $\hat{\theta}_k + c_k \Delta_k$ and $\hat{\theta}_k - c_k \Delta_k$. Parameter boundaries are imposed before the objective function evaluation.

6. Approximate the gradient vector using equation 3.20.

7. Calculate $\hat{\theta}_{k+1}$ using equation 3.16.

8. If converge, stop the process. If not, return to step 3.

SPSA and its variations have been applied extensively in the field of DTA calibration. Ma et al. (2007) compare the performance of SPSA against the genetic algorithm (GA) and the trial-and-error iterative adjustment algorithm (IA) for the
calibration of a microscopic simulation model in a northern California network and conclude that SPSA can achieve the same level of accuracy as the other two do but it has a significantly shorter running time. Huang et al. (2010) applied SPSA for the calibration of dynamic emission models. This research uses a microscopic traffic simulator and the aggregate estimation ARTEMIS as a standard reference. Lee and Ozbay (2008) propose a Bayesian calibration methodology and applied a modified SPSA algorithm to solve the calibration problem of a cell transmission based macroscopic DTA model. In this formulation, the probability distributions of model parameters are considered instead of their static values. Paz et al. (2012) calibrate all the parameters in CORSIM models simultaneously using SPSA and demonstrate its effectiveness.

These applications of SPSA, however, are all limited to networks with small scale in terms of size and number of time intervals. In a case study for the entire expressway system in Singapore, which is discussed in detail in Chapter 5, it was found that although SPSA kept its computational efficiency, its performances in terms of convergence rate and long run accuracy deteriorated significantly when the problem scale increased. The errors stopped to decrease at relatively high values. Different values of algorithm parameters were tested and adaptive step sizes \( (a_k) \) were implemented. However, no significant improvement was made. The next section analyzes the reasons for SPSA's performance deterioration with an example toy network.

### 3.3 Shortcomings of the application of SPSA to DTA calibration

For a better illustration of the shortcomings of SPSA and the improvement idea, we assume the inverted variance-covariance matrix \( \Omega_M^{-1} \) is the identity matrix and \( \Omega_\theta^{-1} \) is a matrix with all zeros. This is equivalent to the situation where we don't consider the deviations between estimated parameter values and their \( a \text{ priori} \) values. At the same time, we consider the deviations between simulated and observed measurement
values in an OLS way. The analysis and methodology can be easily extended to the general formulation in equation 3.12.

The simplified objective function is:

\[ z(\theta) = \sum_{j=1}^{D} \varepsilon_{Mj}^2 \]  \hspace{1cm} (3.21)

We denote

\[ \varepsilon_{Mk}^+ = F_M(\hat{\theta}_k + c_k \Delta_k; M^0; G) \]  \hspace{1cm} (3.22)
\[ \varepsilon_{Mk}^- = F_M(\hat{\theta}_k - c_k \Delta_k; M^0; G) \]  \hspace{1cm} (3.23)

which are the deviation vectors obtained from the two perturbations. \( \varepsilon_{Mk}^+ \) is the \( j \)th element in \( \varepsilon_{Mk}^+ \) and \( \varepsilon_{Mk}^- \) is the \( j \)th element in \( \varepsilon_{Mk}^- \).

Based on our simplified objective function in equation 3.21, equation 3.20 can be rewritten as

\[ \hat{g}_{ki}(\hat{\theta}_k) = \frac{\sum_{j=1}^{D} [(\varepsilon_{Mkj}^+)^2 - (\varepsilon_{Mkj}^-)^2]}{2c_k \Delta_{ki}} \]  \hspace{1cm} (3.24)

The reason for the performance deterioration of SPSA was found to be related to a gradient approximation error that increased rapidly with the problem scale. The source of this error was not the stochasticity in the DTA models, or the inconsistency in the observed data due to measurement error, but the way SPSA estimated gradients. In each iteration, the gradient estimation process essentially tries to find a direction and amplitude for each parameter value in the decision vector to move. This is achieved by comparing the influences to the system caused by perturbing each of the parameter value in two opposite directions. Given our formulation in equation 3.21, in SPSA the influences caused by perturbing the value of a specific parameter are determined by a scalar: the sum of all the distances between model outputs and corresponding observed measurements. As all the parameters are perturbed at the same time, the change in a measurement value may or may not be caused by this
specific parameter. This may not be a major issue in systems where each parameter is highly correlated to most of the measurements, because in that case, the change in each parameter is responsible for the change of almost every measurement value.

However, in a real-world traffic system, correlations between model parameters and measurements are often sparse. In the spatial dimension, most of the model parameters tend to have a relatively local effect to the system. For example, by changing the OD flow of one OD pair, only traffic volume measurements along the paths between this OD pair will be influenced directly. In the temporal dimension, changing the value of the OD flow at 7:30 will have little influence to the network measurements after 8:30, and will have absolutely no influence to the measurements before 7:30. Therefore, in the DTA off-line calibration problem with a large scale network and a large number of intervals, a large number of uncorrelated measurements introduce a great amount of disturbing signal, which makes it very hard to estimate the actual influence from the perturbation of each of the parameter value to determine a good direction and amplitude to move.

We use an example network (see Figure 3-2) to better illustrate the problem.

![Figure 3-2: An example network for the illustration of estimation error in SPSA](image)

Assume the length of an interval is 15 minute and the simulation period consists of 20 intervals. Assume the longest travel time in this network is 30 minutes, which is two intervals. 6 sensors provide time-dependent counts data as shown in the figure. Therefore, we have in total 120 measurements. Among all the candidate OD flows to calibrate, we consider OD pair from A to B and its flow in the 10th interval in this analysis. Suppose this specific OD flow is the $i_{th}$ element in $\theta$. In SPSA, at $k_{th}$
iteration the element in the gradient vector for this OD flow is approximated as

$$\hat{g}_{ki}(\hat{\theta}_K) = \sum_{j=1}^{120} \left[ (\epsilon^+_{Mkj})^2 - (\epsilon^-_{Mkj})^2 \right] \frac{2c_k \Delta_{ki}}{(3.25)}$$

The numerator tries to approximate the influence on measurements caused by changing this specific OD flow, where the term $(\epsilon^+_{Mkj})^2 - (\epsilon^-_{Mkj})^2$ is decided by the change in the $j$th simulated measurement. However, in this problem, in a specific interval, only counts from sensor 3 and 4 are caused by the change by this OD flow. The changes in sensors 1, 2, 5, 6 are caused by other parameters and introduce disturbing signal. At the same time, the changes of measurement values from intervals before the 10th interval is totally irrelevant to the change of the OD flow in the 10th interval. Based on our longest trip assumption, the changes of measurement values from intervals after the 12th interval is also irrelevant to the change of the OD flow in the 10th interval. Therefore, in the numerator we sum up values from 120 measurements but as much as 116 of them are irrelevant to the change of the current OD flow and a great amount of noise is introduced to the gradient estimation. In other words, the number used to compute the approximated gradient consists of only 3% useful signal and 97% of irrelevant noise!

In real world DTA applications, in order to provide useful information for planners and travelers, the scale of network and the number of intervals to consider (usually across an entire day) is much larger than this small toy network. The gradient approximation problem will be more serious. For reference of network sparsity in real world applications, in the entire expressway network of Singapore, on average an OD pair is correlated to only 15% of the sensors.

To solve this problem, a weighted gradient approximation method and the corresponding solution algorithm, W-SPSA, is proposed in the next section.
3.4 The Weighted SPSA algorithm

A weighted gradient approximation method is introduced to exclude the negative influence of irrelevant measurements in the gradient approximation process of SPSA. In this approach, measurements are considered in a weighted manner based on their relevance to a parameter:\(^1\):

\[
\hat{g}_{ki}(\hat{\theta}_k) = \frac{\sum_{j=1}^{D} w_{ji}[\epsilon_{Mkj}^2 - \bar{\epsilon}_{Mkj}^2]}{2c_k\Delta_{ki}} = \frac{1}{2c_k\Delta_{ki}} \mathbf{W}_i' \begin{bmatrix} (\epsilon_{Mk1})^2 - (\bar{\epsilon}_{Mk1})^2 \\ \vdots \\ (\epsilon_{MkD})^2 - (\bar{\epsilon}_{MkD})^2 \end{bmatrix}
\]

(3.26)

where \(w_{ji}\) is the element at the \(j_{th}\) row and \(i_{th}\) column of a \(D \times P\) weight matrix. \(\mathbf{W}_i\) is the \(i_{th}\) column of the matrix. As introduced in the previous section, \(D\) is the number of deviations (measurements plus historical parameter values) and \(P\) is the number of parameters.

For a change in parameter \(\theta_i\), \(w_{ij}\) represents the relative magnitude of change in measurement \(j\), compared to other measurements. By putting more weights on relevant measurements and less/no weights on less relevant measurements/irrelevant measurements, we can effectively reduce the estimation error and therefore provide a better gradient approximation.

---

\(^1\)Although the objective function value is not utilized directly in the gradient estimation process, it is used during the calibration process to evaluate the goodness-of-fit after each algorithm iteration.
Under the GLS formulation in 3.12, equation 3.26 becomes

$$
\hat{g}_{ki}(\hat{\theta}_k) = \frac{E_M + E_\theta}{2c_k \Delta_{ki}}
$$

(3.28)

where

$$
E_M = [(\epsilon^+_{MK})' \text{diag}(W_{MK}^{o \frac{1}{2}})] \Omega_M^{-1} [(\epsilon^+_{MK})' \text{diag}(W_{MK}^{o \frac{1}{2}})]' -
$$

$$
[(\epsilon^-_{MK})' \text{diag}(W_{MK}^{o \frac{1}{2}})] \Omega_M^{-1} [(\epsilon^-_{MK})' \text{diag}(W_{MK}^{o \frac{1}{2}})]'
$$

(3.29)

$$
E_\theta = [(\epsilon^+_{\theta k})' \text{diag}(W_{\theta i}^{o \frac{1}{2}})] \Omega_\theta^{-1} [(\epsilon^+_{\theta k})' \text{diag}(W_{\theta i}^{o \frac{1}{2}})]' -
$$

$$
[(\epsilon^-_{\theta k})' \text{diag}(W_{\theta i}^{o \frac{1}{2}})] \Omega_\theta^{-1} [(\epsilon^-_{\theta k})' \text{diag}(W_{\theta i}^{o \frac{1}{2}})]'
$$

(3.30)

where $W_{MK}^{o \frac{1}{2}}$ is the element-wise square root of the $i_{th}$ column of the weight matrix for measurements $W_M$. $\text{diag}(W_{MK}^{o \frac{1}{2}})$ is the diagonal matrix built from $W_{MK}^{o \frac{1}{2}}$. $W_\theta$ is the weight matrix for historical values. Typically $W_\theta$ is an identity matrix, because changing the value of a parameter only influences the deviation between this specific parameter’s estimated value and its historical values. The definitions of $\epsilon^+_{\theta k}$ and $\epsilon^-_{\theta k}$ are similar to those of $\epsilon^+_{MK}$ and $\epsilon^-_{MK}$.

These relevance measurements can be obtained from the adjusted Jacobian matrix $J$, where each element is replaced by the absolute value from the traditional Jacobian matrix. Absolute value is used because we are only interested in the magnitude.

$$
J = 
\begin{bmatrix}
\frac{\partial c_1}{\partial \theta_1} & \cdots & \frac{\partial c_1}{\partial \theta_1} & \cdots & \frac{\partial c_1}{\partial \theta_p} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial c_p}{\partial \theta_1} & \cdots & \frac{\partial c_p}{\partial \theta_1} & \cdots & \frac{\partial c_p}{\partial \theta_p} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\frac{\partial c_p}{\partial \theta_1} & \cdots & \frac{\partial c_p}{\partial \theta_1} & \cdots & \frac{\partial c_p}{\partial \theta_p}
\end{bmatrix}
$$

(3.31)

The weight matrix $W$ is a local approximation of the Jacobian matrix at a specific point $\theta_k$. Due to the lack of closed-form analytical relationship between parameters and measurements, a local linear approximation of $\frac{\partial c_j}{\partial \theta_i}$ is usually used.
\[ W|_{\theta_k} = \hat{J}|_{\theta_k} \] (3.32)

Weighted SPSA, or W-SPSA, is the proposed solution algorithm for the off-line calibration problem that applies the weighted gradient estimation in the stochastic path searching process. Figure 3-3 shows the general work flow of W-SPSA. Typically the DTA off-line calibration problems are highly non-linear and the Jacobian matrix changes with current parameter values. Therefore, the weight matrix should be re-estimated during the calibration process along with the change of current estimated parameter values. The frequency of this process depends on the specific problem, the stage of the calibration process, and the amount of available computational power. The dash line from off-line calibration back to weight matrix estimation indicates that the results from the previous off-line calibration stage are usually used as the input of weight matrix estimation. For example, the current values of parameters to calibrate, simulated network conditions, etc. However, this re-estimation process is not necessary under a special case of W-SPSA: SPSA (the weight matrix is an all-one matrix). The next section discusses different approaches to estimate the weight matrix and their pros and cons.

3.5 Estimation of weight matrices

The accurate estimation of weight matrices is the key to successfully applying W-SPSA for the off-line calibration of DTA models. A perfect weight matrix estimation approach should be easy to implement, efficient to execute, and it should be able to provide accurate estimation for different types of parameters and measurements. Four different approaches are proposed:

- Analytical approximation
- Numerical approximation
3.5.1 Analytical approximation

Analytical approximation is an approach to estimate the weights based on known analytical relationships (or approximated linear relationships at the current point) between parameters and measurements. Network knowledge, including network topology, paths choice set, equilibrium link travel time, route choice model, etc., is usually required as inputs of this approach. However, the complete forms of these relationships are typically unknown and the required inputs change with the model parameter values. Therefore, an approximated linear relationship is usually used with the current estimated parameter values.

For example, to calculate the weights between OD flows and link traffic flows (usually measured by loop detectors in the form of sensor counts), dynamic assignment matrices are usually used. The assignment matrix stores the way in which each OD
flow is assigned to different links (sensors). Figure 3-4 shows an example network with three nodes (A, B, and C) shown as circles and four sensors (1, 2, 3, 4) shown as boxes. Lines between nodes are available paths. Suppose we only consider the OD flow from A to B in the decision vector for two intervals. There are in total two parameters to calibrate: the OD flow at interval 1 and the OD flow at interval 2, or $\theta_1$ and $\theta_2$, respectively. We have 4 sensors for two intervals, which result in 8 measurements ($M_i$, $i = 1,...,8$). $M_1$ to $M_4$ correspond to measurements from the 4 sensors at the first interval. The weight matrix for this problem is therefore a $8 \times 2$ matrix. Each element in the matrix specifies the relative correlation between a parameter and a measurement.

![Figure 3-4: An example network for the illustration of analytical weight matrix estimation](image)

Assume under the current traffic condition and route choice behaviors, at the first interval, 100 vehicles leave A for B. 70 of them choose the straight path with sensor 3 and 4, because it’s shorter. 30 of them choose the path with sensor 2. As sensor 2 and sensor 3 are close to point A, all of the vehicles can pass them and therefore be recorded within the first interval. None of the vehicles are able to pass sensor 4 in the first interval and they all pass sensor 4 at the second interval. Assume that the vehicles that leave node A at the second interval have exactly the same behavior as those leaving at the first interval. The fractions of different OD flows being captured
by (assigned to) different sensor measurements are summarized in Table 3.1.

<table>
<thead>
<tr>
<th></th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$M_4$</th>
<th>$M_5$</th>
<th>$M_6$</th>
<th>$M_7$</th>
<th>$M_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.1: Assignment fractions of different OD flows to different measurements

These fractions can perfectly capture the relative correlation (weight) between parameters and measurements under the current condition. For example, changing OD flow at the first interval by some number will lead to 0 change in measurement 1 and 4, 0.3 unit change in measurement 2 and 0.7 unit change in measurement 3 and 8. Therefore, in the gradient approximation, weighing $M_2$ with 0.3, $M_3$ and $M_8$ with 0.7, others with 0 will accurately reflect the “real” changes of the measurements caused by changing $\theta_1$. The corresponding weight matrix for this example problem is:

$$W = \begin{bmatrix} 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \end{bmatrix}$$ (3.33)

There are two approaches to obtain the assignment matrix. One is to use the simulator to track and count the number of cars in each OD pass different sensors at different intervals. The other is to analytically calculate the fractions using network topology, path choice set, current route choice model and equilibrium travel time. The second one is more flexible and potentially can provide more accurate results. For other types of parameter-measurement correlations, similar ideas can be applied but more sophisticated analytical derivations are required to capture the usually indirect and non-linear relationships.

The analytical approximation approach has the advantage of computational efficient, as at most 1 simulation run is required for each weight matrix estimation/update. It’s also able to provide very accurate weights if the relationships between parameters and measurements at the current point are relatively straightforward and linear. However, for parameter-measurement pairs that have complicated indirect relationships, this approach can hardly generate weights for them without significantly increasing
the complexity of analytical approaches used.

### 3.5.2 Numerical approximation

The numerical approximation approach uses the simulator to approximate the Jacobian matrix at the current point through numerical experiments. The flowchart of a typical numerical experiment is shown in Figure 3-5.

Assume that $n = 1, 2, ..., N$ is the current experiment and $N$ is the total number of independent experiments. $P$ is the total number of parameters and $D$ is the total number of measurements (the weights for historical parameter values are typically calculated directly). $M_{ni}^{s+}$ ($M_{ni}^{s-}$) is the simulated measurement value of the $i_{th}$ measurement using the parameter values after the ”+”(”-”) perturbation. In one experiment, all the parameters are perturbed at the same time near the current value to two opposite directions. $\Delta \theta_{ni}$ is randomly generated and independent across different experiments (it typically has a pre-defined amplitude and a randomly generated
The changes of simulated measurement values under each of the two perturbed parameter vectors are evaluated by running the DTA model. Each element in the Jacobian matrix is approximated as the change in a measurement value divided by the change in a parameter. After running \( N \) independent experiments, the estimated weights are obtained by averaging the result from each experiment:

\[
\frac{\partial M_j^S}{\partial \theta_i} \approx \frac{\sum_{n=1}^{N} \frac{\Delta M_{nj}^S}{\Delta \theta_{ni}}}{N} \tag{3.34}
\]

The estimation result is expected to improve with the increase of \( N \) and to be unbiased.

Under this framework, different types of parameters and measurements are treated in a same manner. At the same time, no explicit analytical network knowledge is required in the estimation. Therefore, this approach covers all different kinds of parameter-measurement pairs using a single, and simple method. However, the capability of obtaining accurate results depends highly on a big enough number of experiments and each experiment requires two simulation runs, which lead to intensive computational overhead. Fortunately, this approach can be easily parallelized, as the experiments are completely independent. If enough computers are available, no complex parallelization algorithm is needed to speed up this process significantly.

### 3.5.3 SPSA

A special case of the weight matrix is the “all-one” matrix, where all the weights are set to 1. In other words, for each parameter, all the measurements are supposed to have the same degree of relevance to this parameter. With this weight matrix, the W-SPSA algorithm becomes the original SPSA algorithm. Therefore, SPSA can be viewed as a special case of W-SPSA. Or, W-SPSA is the generalized SPSA.

This approach has the advantage of not relying on the estimation of weight matrix. It's the simplest method to implement and most computational efficient as no weight matrix estimation or update is required. However, as discussed in previous sections, when the network scale is large, the correlations between parameters and
measurements are extremely sparse, using the “all-relevant” assumption results in a highly noisy gradient estimation process, which leads to extremely slow rate of objective function value reduction. Therefore, this approach is only considered to be applied when the weight matrix is very hard to obtain or at the final stage of the calibration process when the goodness-of-fit has already been improved significantly through other methods\(^2\).

### 3.5.4 Hybrid method

To better take advantage of the pros of each estimation method, a hybrid weight matrix can be built and adjusted during the calibration process:

\[
W_h = w_a W_a + w_n W_n + w_o W_o
\]  

(3.35)

where \(W_h\) is the hybrid weight matrix, which equals the weighted sum of a weight matrix estimated using analytical approach \((W_a)\), a weight matrix estimated from numerical approach \((W_n)\) and an all-one weight matrix \((W_o)\). The weights depend on the relative confidence on each part and can change in different stages of the calibration process.

Another way to generate a hybrid weight matrix is to divide the weight matrix into different parts and using the most appropriate estimation approach for each part. For example, for the weights of OD flow-sensor counts, the analytical approach can be used. For the weights of supply parameters-sensor counts, numerical approach can be used, because the analytical relationship is very hard to derive. For the weights of route choice parameters-sensor counts, all one matrix is good enough because the route choice parameters influence the whole network across the whole simulation period and the relationship between these parameters and traffic measurements are hard to capture even using the numerical approach.

In empirical applications, high quality weight matrices are usually not easy to

\(^2\)Experiments in Chapter 5 show that even after achieving a good enough goodness-of-fit, applying SPSA in large networks does not improve the objective function further.
get through applying only one approach. This hybrid method provides some insights about possible ways to use different weight matrix estimation methods at the same time. The best way to estimate weight matrices is problem-specific and requires a good understanding of the characteristics of the problem, in-depth analysis of the parameters and measurements, and some engineering judgment.
Chapter 4

Synthetic Tests

In this chapter, a synthetic test system is built to systematically compare the performance of SPSA and W-SPSA under various problem settings. Moreover, the characteristics of W-SPSA and weight matrix estimation approaches are investigated and analyzed to provide insights and guidance to the application of W-SPSA in the off-line calibration of real DTA systems.

4.1 Experimental design

In the synthetic test system, time-dependent OD flows are calibrated using time-dependent traffic counts in different links. Instead of using a DTA system on a real world road network to model the traffic dynamics, linear functions are used to map OD flows to sensor counts. The true values of the OD flows are randomly generated numbers with magnitudes comparable to real world cases. The initial (seed) OD flows are generated by randomly perturbing the true OD flows. The observed sensor counts are computed using the true OD flows and randomly generated linear relationships:

\[
M_j = \sum_{i=1}^{P} \beta_{ji} \theta_i
\]  

where \( \theta = \theta_1, \theta_2, ..., \theta_P \) are time-dependent OD flows with a total number of \( P \) and \( \mathbf{M} = M_1, M_2, ..., M_D \) are time-dependent sensor counts with a total number of \( D \).
\( \beta = \{\beta_{ji}\} \) are coefficients of the linear functions, which are generated randomly based on specific rules. The OD flow in one interval is assumed to be only related to sensor counts in the same interval. In other words, \( \beta_{ji} = 0 \) if OD flow \( i \) and sensor count \( j \) are from different intervals. Based on the definition of weights in W-SPSA, \( \beta \) is exactly the weight matrix and does not change at different points due to the linearity of the functions. In the experiments in this chapter, \( \beta \) is called the “perfect weight matrix”.

The dimension of the problem is 1000 OD pairs and 100 sensors, with, at every interval, one value for each OD flow and sensor count. Different number of intervals will be tested. The ratio between the number of OD pairs and sensors is comparable to the real world case studies (e.g., in Balakrishna (2006), the Los Angeles network has 1029 OD pairs and 203 detectors; in the Singapore expressway network discussed in Chapter 5, there are 4106 OD pairs and 216 sensors after cleaning out malfunctioning sensors). Parameter correlation quantifies the degree a parameter correlates to all the measurements. Specifically in this test, for a given OD flow \( \theta_0 \), the parameter correlation equals the number of non-zero \( \beta_{jo} \)s divided by the total number of \( \beta_{jo} \)s (i.e., the fraction of sensor counts that are correlated with this OD flow). The Network correlation is the average over all parameter correlations in the network. In this system, given a network correlation, all parameters have different parameter correlations generated randomly with a mean value that equals the network correlation. The parameter correlations are limited to a range centered at the network correlation. The non-zero \( \beta_{jis} \) are chosen randomly and the value for each non-zero \( \beta_{ji} \) is a uniformly distributed random number ranging from 0.1 to 1. For example, if the chosen network correlation is 0.2, each OD flow is randomly correlated to 10-30 sensors counts and the weights (\( \beta_{ji} \)) are randomly generated values between 0.1 and 1 (the weights for uncorrelated sensors are 0).

In the synthetic tests and the case study in the next section, the root mean square normalized error (RMSN) was used to measure the discrepancy between the simulated measurement values (or estimated parameter values) and the observed measurement.
values (or true parameter values):

\[ RMSN = \sqrt{\frac{D \sum_{j=1}^{D} (M_j - \hat{M}_j)^2}{\sum_{j=1}^{D} M_j}} \] (4.2)

where \( M_j \) is the \( j \)th observed (or true) value and \( \hat{M}_j \) is the \( j \)th simulated (or estimated) value. \( D \) is the total number of measurements (plus historical parameter values).

Notice that, in equation 3.12, we use both discrepancies between simulated measurements and observed measurements, and deviations between estimated parameter values and historical parameter values. It is the general way for DTA calibration when high quality historical values are available. In the experiments, however, only the discrepancy between simulated measurements and observed measurements are considered in the objective function and RMSN calculation.

The experiments are done in the Matlab (R2011b) environment. Part of the algorithm parameters in both SPSA and W-SPSA were determined based on Balakrishna (2006) and preliminary tests: \( \alpha = 0.602, \gamma = 0.101, A = 50 \). For \( a \) and \( c \), which are the perturbation and advance step size, linear search was performed to find their best values for each case the experiments.

### 4.2 Results

#### 4.2.1 Scalability

This experiment tests the scalability of SPSA and W-SPSA by comparing their performance under different numbers of intervals, assuming 1000 OD pairs and 100 sensors per interval. The network correlation is 10% and each OD flow is correlated with a random number of sensors between 1 and 20. Random weights are assigned to the sensors between 0.1 to 1. In W-SPSA, perfect weight matrices are assumed known and used. The results are shown in Figure 4-1, where the horizontal axis represents the number of algorithm iterations during the calibration process, and the vertical axis represents the RMSN after a specific number of iterations.
Figure 4-1: Scalability performance of SPSA and W-SPSA in terms of RMSN with number of algorithm iterations

W-SPSA outperforms SPSA in terms of convergence rate and final achieved accuracy independently of the number of intervals. The convergence rate of SPSA and W-SPSA decrease with the increase of the number of intervals (so that the scale and sparsity of the problem). However, W-SPSA achieves similarly good long run accuracy with different number of intervals but SPSA has a significantly worse accuracy performance.

The results show that while SPSA is scalable in term of computational time (two objective function evaluations regardless of the problem scale), it is not scalable enough in terms of convergence and accuracy performance. W-SPSA achieves significantly better scalability in both computational performance and accuracy performance. The additional calculation in the gradient approximation process of W-SPSA is negligible compared to the running time of the simulator.

4.2.2 Network correlation

As discussed in Chapter 3, the deterioration of SPSA’s performance is a result of the sparse correlations in the network. This experiment tests the influence of network correlation to the performance of SPSA and W-SPSA. Only 1 interval is considered
in this experiment, with 1000 ODs and 100 sensors. Perfect weight matrix is assumed to be known and used in these experiments. Figure 4-2 shows the results of this test.

Figure 4-2: Performance of SPSA and W-SPSA in terms of RMSN with different degrees of network correlation

The blue bars represent the best achieved RMSN using SPSA with different network correlations and the red bars represent the best achieved RMSN using W-SPSA. The performance of SPSA improves when the network correlation increases. For W-SPSA, the difference is less significant. It achieves the best result with 5% correlation and the worst result with 50% correlation. Further experiments and studies are needed to find the possible reasons for this pattern. The difference between the performance of SPSA and W-SPSA becomes smaller when the degree of network correlation increases. With a 100% network correlation, W-SPSA is only slightly better than SPSA, because SPSA assumes the weights are all 1, while with perfect weight matrix, W-SPSA knows the accurate numbers of the weights. However, the influence of such information is very small. This will be discussed in detail in the next sub-section.

In real world networks, the network correlations are usually less than 20%. If more intervals are considered in the simulation, the actual overall correlation across all the intervals will be much smaller. Therefore, in real world applications, if the scale of the network is large and multiple intervals are considered, W-SPSA will very
likely be a better choice and pre-computing the network correlation is not necessary.

4.2.3 Weight matrix

Perfect weight matrices are extremely hard to get, if not impossible, in real world applications. Therefore, to successfully apply W-SPSA, we must understand how much the performance of W-SPSA is influenced by the quality of weight matrices and how to obtain weight matrices with higher quality. In the experiments in this part, four intervals are considered with 1000 ODs and 100 sensors in each interval. The network correlation is 10% within each interval.

The accurate values of weights are usually hard to calculate due to the lack of accurate estimation of current network conditions and the approximations in the analytical functions or in the numerical approach. Figure 4-3 shows the performance of W-SPSA with a weight matrix that only has 0-1 information instead of accurate weights between 0 and 1, where 0 means a specific parameter is not related to a measurement or the correlation is smaller than a threshold, and 1 means the parameter-measurement pair is correlated. We calculated the 0-1 weight matrix based on the weight matrix by turning all the non-zero weights to 1.

![Figure 4-3: Performance of SPSA and W-SPSA with different weight matrices](image)

The performance of W-SPSA with 0-1 weight matrix is compared with SPSA and
W-SPSA with perfect weight matrix. According to the results, the lack of accurate values in the weight matrix has a limited influence to the performance of W-SPSA in terms of both convergence rate and long run accuracy. This characteristic is also illustrated in Figure 4-2 when the network correlation is 100%. The only difference between SPSA and W-SPSA under that scenario is that W-SPSA has accurate weights while SPSA only have 0-1 information (all one, actually). This observation indicates that in real applications, we can use approximated weight matrices with only 0s and 1s. Huge efforts can be saved because finding the accurate values of the weights is not necessary, if possible.

For numerically estimated weight matrices (see Chapter 3.5.2, numerical approximation), there is rarely any weight that is literally 0. Therefore, although the correlations between parameters and measurements are sparse, the estimated weight matrix cannot be saved directly as a sparse matrix. Saving and using the full weight matrix introduce great computational burden in terms of RAM space and running time. With the finding that 0-1 weight matrix can serve as good as the accurate weight matrix, the numerically estimated weight matrices can be processed as a sparse matrix by setting all the weights smaller than a threshold to 0, which will help save computational resources significantly without compromising the performance much.

Another experiment was performed to test the influence of unknown weights. Figure 4-4 shows the best achieved RMSN when different percentage of the weights are known (only 0-1). The unknown weights are set as 1.

The results show that the performance of W-SPSA becomes worse significantly when there are more unknown weights. When 100 % of the weights are unknown, W-SPSA turns to SPSA. In real world applications, using the analytical approach usually can provide high quality weights but only for part of the parameter-measurement pairs. Simply setting the remaining weights to 1 is easy to implement but will compromise the performance of the algorithm. Therefore, using the numerical approach to estimate the remaining weights is necessary (this is one way to generate a hybrid weight matrix (see Chapter 3.5.4, hybrid methods)).

In the numerical approximation approach, there are two important algorithm
parameters to decide: the number of iterations and the threshold under which the weights are set to zero. Figure 4-5 shows the final RMSN achieved by using different weight matrices. SPSA represents the original SPSA algorithm, or all one weight matrix. Perfect represents using the perfect weight matrix. 100, 500, 2500 represent using numerically estimated weight matrices that are estimated with 100, 500, 2500 iterations of the estimation algorithm given a fixed threshold (0.4). The results show that more iterations in the approximation process lead to better final achieved RMSN using W-SPSA. After a certain number of iterations, the result is comparable to that using perfect weight matrix.

Figure 4-6 shows the improvement of the sensitivity and precision of the estimated weight matrix compared to the perfect weight matrix, given a fixed threshold (0.4). Sensitivity and precision are defined as:

\[
\text{sensitivity} = \frac{\text{true}^1}{\text{true}^1 + \text{false}^0} \tag{4.3}
\]

\[
\text{precision} = \frac{\text{true}^1}{\text{true}^1 + \text{false}^1} \tag{4.4}
\]

where \text{true}^1 represents the number of correctly identified 1s from the perfect weight matrix under the current threshold. For example, if an element in the perfect weight
matrix is 0.4, in the estimated weight matrix it is 0.35, the threshold is 0.3, this element is estimated correctly as 1. *false*0 is when the estimated weight is 0 but it is actually 1 in the perfect weight matrix. Intuitively, sensitivity reflects the ability to detect 1s in the perfect weight matrix, and precision evaluates the reliability of the estimated 1s. The results show that sensitivity and precision increase when the number of iterations increases. However, after a certain number of iterations, the improvement becomes limited.

Figure 4-7 shows the trade-off between sensitivity and precision by comparing them under different thresholds. The number of estimation iterations is 1000. In the plot, sensitivity keeps decreasing with the increase of threshold. However, the change of precision is not monotonic. It increases with the threshold before reaching its highest value when the threshold is 0.6 and then decreases. This indicates that, regardless of whether sensitivity or precision is more important to the problem, the threshold should not be set too high.

In conclusion, weight matrices with only 0 and 1 are sufficient for applying W-SPSA successfully. Whenever there are unknown weights, using the numerical method to fill these blanks is better than simple filling with 1s. For the numerical approximation approach, more iterations in the estimation process lead to better results, but
Figure 4-6: The influence of number of iterations in the numerical weight matrix approximation approach to sensitivity and precision

Figure 4-7: Sensitivity and precision under different threshold
making the number of iterations extremely large will introduce little benefit while significantly increasing the computational overhead. For the threshold, if the problem requires detecting more small correlations and therefore improve the final accuracy it should be small. On the other hand, if the current goal is to speed up the convergence process and only consider the important correlations, it should be set as a larger value, but never too large (more than 0.5, for example).

4.2.4 Sensitivity to noise

In real world applications, there is usually noise in the DTA system and in the data acquisition process, which influences the DTA calibration process in different ways based on its type and magnitude. In the DTA system, given exactly the same inputs, the outputs may be different due to the stochasticity in the models. In the data acquisition process, there are measurement errors which make the observed data different from real traffic conditions. These measurement errors come from a variety of sources, including defective equipments, immature techniques, and other human related mistakes.

Figure 4-8 shows the influence of different levels of noise from the simulator on the calibration process using W-SPSA. 2% noise in the simulator is implemented by adding a uniformly distributed error from -2% to +2% to the simulated measurement values before the gradient estimation and goodness of fit evaluation process. The same procedure applies to 5% noise. This magnitude of noise is comparable to that in real simulation-based DTA systems. For example, the simulator noise in DynaMIT in the Singapore expressway network case study was 3%. The figure shows that higher level of noise influences both the convergence rate and the long run accuracy. At the same time, noise from the simulator makes the curve less smooth.

Figure 4-9 shows the influence of different levels of noise from measurements on the calibration process using W-SPSA. The 10% noise in measurement is implemented by adding a uniformly distributed error from -10% to +10%. The same procedure applies to 20% noise. In real world applications, the noise in data is usually much higher than that from the simulator due to the approximation and aggregation in
the data acquisition process and the errors in sensors. However, it is hard to verify the exact noise level in the observed data because usually no true data is available. According to the figure, noise in data influences both the convergence rate and long run accuracy. The data set with noise becomes inconsistent so that a perfect fit is even theoretically impossible. The quality of the data, together with the noise in the simulator, decides the best achievable goodness-of-fit. Therefore, interpreting the measurement of fit without understanding the level of noise is a unsound exercise.

4.3 Summary

In this chapter, a synthetic test system is built to test the performance of SPSA and W-SPSA in terms of convergence rate, long run accuracy and scalability. Several important characteristics of W-SPSA are investigated in order to provide guidance to its application to real world DTA off-line calibration problems.

W-SPSA is proven to outperform SPSA in all different problem scales and different network correlations in terms of convergence rate and long run accuracy. The performance advantage of W-SPSA becomes larger when the problem scale increases and when the parameter-measurement correlations are more sparse. W-SPSA is scal-
W-SPSA has a characteristic that is important in real DTA system calibration problems: the accurate values of weights only have little influence on the final calibration result. In other words, only 0-1 information is sufficient to generate a high quality weight matrix.

In real applications, there are usually weights that are extremely hard to calculate using an analytical method. Using numerical approximation to fill these blanks can lead to better results than filling them with all 1s. In the numerical approximation process, two key parameters are required: the number of iterations to estimate weights, and the threshold under which the weight is set to 0. More iterations lead to better results but the improvement rate slows down significantly when the number of iterations is larger than a certain value. A smaller threshold should be used when the purpose is to capture more subtle correlations and to make the process more unbiased. A larger threshold should be sued when the purpose is to speed up the convergence process and to make the process more efficient. However, the threshold should not be larger than 0.5.

Noise in both the simulator side and the measurement side will influence the
convergence rate, long run accuracy and smoothness of the RMSN decreasing process. Noise data introduce inconsistency and a perfectly fit to these data is theoretically impossible. The best achievable fit is decided by the inconsistency in the data and the noise in the simulator. Therefore, without understanding the data quality and the stochasticity of the simulator, evaluating the calibration process by simply referring to the value of final RMSN may not be feasible.
Chapter 5

Case Study

In this chapter, a case study of the entire expressway network in Singapore is discussed to demonstrate the performance of W-SPSA in a real-world large-scale DTA system. First, the DTA model and candidate parameters being calibrated are introduced. Then, network configuration and available field traffic data are detailed. Moreover, concerns on raw data quality are raised and a data cleaning process is introduced. Next, the calibration process is discussed with a focus on empirical considerations and modifications when applying W-SPSA in this case study. Calibration results are presented in the last part of this chapter.

5.1 Model and parameters

DynaMIT (Dynamic network assignment for the Management of Information to Travelers) is used in the case study to model the network and traffic dynamics of the entire Singapore expressway system. DynaMIT is a simulation-based DTA system with a microscopic demand simulator and a mesoscopic supply simulator. The demand simulator is capable of accurately modeling the time-dependent OD flows, pre-trip decisions and en-route choices of individual drivers given real time information. The supply models simulate the traffic dynamics, the formation, spill-back and dissipation of queues and the influence of incidents. The sophisticated interactions between demand and supply allows the system to estimate and predict network traffic conditions.
in a realistic manner. More detailed discussions about the features, framework and implementation of DynaMIT can be found at Ben-Akiva et al. (2010).

In the demand side, a route choice model and a set of time-dependent OD flows are calibrated. DynaMIT models drivers' behavior when selecting a path given a set of alternatives using a Path-size Logit model (Ramming, 2001). Path-size Logit model considers the overlap between different paths and is therefore able to more accurately model the route choice behavior. The probability of choosing a path given a set of alternatives is calculated as:

\[ P(i) = \frac{e^{V_i + \ln PS_i}}{\sum_{j \in P} e^{V_j + \ln PS_j}} \] (5.1)

where \( P(i) \) is the probability of selecting path \( i \). \( PS_i \) is the size of path \( i \). \( P \) is the choice set, or the set of alternative paths between an OD pair. \( V_i \) is the systematic utility of path \( i \), which is a function of the characteristics of drivers and the attribute of the path. The parameters in this function is usually estimated using disaggregate data before the calibration and adjusted during the joint demand-supply calibration process.

Time-dependent OD flows are an important input to the DTA system. The total number of time-dependent OD flows to calibrate equals the number of OD pairs in the network times the number of time intervals in the simulation. Reliable historical OD flows as initial values of the estimated OD flows are crucial for the success of calibration. However, in this case study, such OD flows are not available. The initial values are decided arbitrarily yet based on local transportation engineers' intuition.

In the supply side, there are three major models: the moving part in a segment, which is modeled by a speed-density function, the queuing model at the end of a segment, which is decided by segment capacities, and the segment performance under incident, which is captured by capacity reduction factors. In this case study, incidents are not considered.

In the moving part of a segment, vehicles move at a speed decided by the current segment density and a segment specific speed-density function:
\[ v = \max \left\{ v_{\min}, v_{\max} \left[ 1 - \left( \frac{k - k_{\min}}{k_{\text{jam}}} \right)^\beta \right]^\alpha \right\} \] (5.2)

where \( v \) is the movement speed of vehicles under the current density \( k \). \( v_{\max}, v_{\min}, k_{\min}, k_{\text{jam}}, \alpha \) and \( \beta \) are model parameters to calibrate. \( v_{\max} \) is the free-flow speed. \( v_{\min} \) is the minimum speed on a segment. The calculated speed is always higher or equal to \( v_{\min} \). \( k_{\min} \) is the minimum density beyond which the speed starts to drop. \( k_{\text{jam}} \) is the jam density. \( \alpha \) and \( \beta \) are parameters that control the shape of the speed-density function curve. In this case study, each segment has its own speed-density relationship function. The initial value of these parameters are estimated by fitting field data.

In the queuing part of a segment, vehicles move to the next segment or form (join) a queue based on capacity calculations. If the current output flow exceeds the segment capacity, or vehicles are unable to move to the next segment due to not enough space, vehicles start to queue. The initial values of capacities are decided based on the Highway Capacity Manual and slightly adjusted according to local filed data.

5.2 Network and data

5.2.1 Network overview

The entire road network in Singapore is shown in Figure 5-1 (Source: OpenStreetMap). In this case study, the expressway system is extracted and modeled in DynaMIT, including expressway links and on-ramps, off-ramps that connect the expressway system to local roads. Figure 5-2 is the computer representation of the expressway system in DynaMIT. The network has accurate and detailed representation of length, geometry and lanes of each segment.

There are in total 831 nodes connected by 1040 links in the expressway network. Each link is made up of several segments based on the geometry, and there are 3388 segments. 4106 OD pairs are chosen among the 831 \( \times \) 830 possible node combinations.
Figure 5-1: Singapore road network (OpenStreetMap)

Figure 5-2: Computer representation of the entire Singapore expressway system in DynaMIT
to make sure each origin is an on-ramp, where vehicles enter the expressway system from local roads, and each destination is an off-ramp, where vehicles depart the network. Other heuristic rules are used to eliminate unreasonable long or detour trips between on-ramps and off-ramps. The simulation time period is from 5am to 10am, or 60 intervals with the length of each interval being 5 minute. The first two hours (24 intervals) are used for network warm-up.

The scale of the calibration problem is therefore:

- 1 route choice model parameter (travel time)
- $4,106 \times 36 = 147,816$ time dependent OD flows
- $3,388 \times 6 = 20,328$ speed-density function parameters
- 3,388 segment capacities

5.2.2 Data overview and cleaning

Data overview

The Land Transport Authority of Singapore (LTA) provides speed and flow data in real time for use in DynaMIT. The real time data feed are sent into the DynaMIT system every 5 minutes for on-line calibration purposes and they are archived to be used in the off-line calibration.

Flow data is provided from EMAS system (Expressway Monitoring and Advisory System). The flow data is obtained from fixed cameras which are mounted on street lamps at distances of approximately 500 to 1000 meters (see Figure 5-3 for example). The cameras are able to count vehicles passing on the road below. Every 5 minutes the cameras report the flow back to the central system. The central system then estimates the flow on the road segment using specific algorithms, even for those without sensors on them. There are in total 338 “sensors” that provide flow measurement every 5 minutes.

Speed data is provided by the LTA for each segment (3388 in the DynaMIT network) on the expressway. It is derived from probe vehicles equipped with GPS.
The exact method for estimating the speed is proprietary and the details have not been made known to us. However it is known that taxis provide the bulk of the raw data used to make the calculations.

Data cleaning

As discussed in Chapter 4, the quality of data influences the final achievable accuracy of the calibration process. Inconsistent data make it harder to fit. More importantly, fitting to inaccurate, or even erroneous data is meaningless in terms of applying the calibrated model to estimate and predict real world traffic conditions. In real world applications, sensor data in all cities are prone to faults due to the unknown random factors in the acquiring process. For example, in a case study in Beijing, the author discusses the data inconsistency and its influence to calibration (Wei, 2010). Therefore, evaluating the quality of data and filtering out wrong data or malfunctioning sensors are important procedures before starting the off-line calibration process.

We investigated the flow data carefully due to the fact that they are fusion data
generated by unknown algorithms and the exact location of some cameras are not available to us. Three consistency tests were done to detect possible inconsistencies within the data. Figure 5-4 shows the three cases that were covered in our tests. The solid circles represent nodes. A line between two nodes is a link. Rectangular boxes represent sensors. The arrows show the direction traffic moves.

In case 1, there is no on-ramp or off-ramp between the two sensors, which means they are supposed to give similar flow patterns and values. If there is a big discrepancy between the measured flow at sensor 1 ($S_1$) and that at sensor 2 ($S_2$) in most intervals, these two sensors are determined to be malfunctioning. Figure 5-5 shows the measured flows from two sensors in a scenario similar to case 1. The specific day and sensor location/ID are not shown here due to the consideration of a Non Disclosure Agreement with the data provider.

In case 2, there is an on-ramp between sensor 1 and sensor 2. The sensors are determined to be malfunctioning if the flow measured by sensor 2 is significantly smaller than the flow measured by sensor 1 in most intervals. Case 3 is similar to case 2 while sensor 2 is supposed to have a lower measured flow. Figure 5-6 shows the measured flows from two sensors in a scenario similar to case 2.

An initial calibration was done using all the 338 sensors. Then, an inconsistency
Figure 5-5: An example of two inconsistent sensors under case 1 scenario

Figure 5-6: An example of two inconsistent sensors under case 2 scenario
check was done. For each recognized inconsistent sensor pair, the sensor gives worse value (comparing with the simulation output) was labeled as malfunctioning and removed. There are 216 reliable sensors left after the cleaning process.

After this cleaning process, there were still sensor failures that were not recognized using the three basic tests. At the same time, due to the lack of incident information, there were data inconsistencies introduced by incidents. For example, without information about a major accident, it is impossible to reproduce the significant reduction of speed and flow during the calibration process. To reduce the influence of incidents and unrecognized malfunctioning sensors, the field traffic data were averaged across the 31 days in August 2011. The averaged data is not “true” data for any specific day, but they reflect the averaged trends and patterns of the network, which are exactly what the off-line calibration aims to capture.

5.3 Calibration process and empirical considerations

This section focuses on the technical aspects when implementing and applying SPSA and W-SPSA for this case study.

5.3.1 Weight matrix calculation

In this case study, the weight matrix is divided into three parts and each part is calculated using numerical approximation methods.

The weights between OD flows and sensor counts, segment speeds are computed based on network topology, latest estimated route choice model and time-dependent link travel times. Given an OD pair and a time interval, among all the vehicles that leave the origin within this interval, the proportion of these vehicles that pass a specific segment / sensor is calculated using the route choice model and the travel times in this time period. For longer trips the travel times in the next few time periods are also considered. An example of this method can be found at Section 3.5,
analytical approximation. These proportions are used as weights between this OD flow and sensor counts/speeds from the sensors/segments along the paths between the OD pair. The weights for measurements outside the alternative paths set of this OD pair are set to 0. Additionally, a weight factor needs to be decided between sensor counts and segment speed, because when calculating the gradient, for a specific OD flow, the errors from related sensor counts and segment speeds are summed up directly while changing the OD flow may have influence to sensor counts and speeds with different magnitude. In this case study, more weights are put on sensor counts because it is believed that changing the OD flows have more influence on counts than segment speed. The weights are updated during the calibration process using the latest estimated route choice model and time-dependent link travel times.

The weights between speed-density parameters, segment capacities and sensor counts, segment speeds are set in a simple way: 1 if the speed-density parameters and segment capacities are at the same segment as the count and speed data; 0 for all other situations. This is an extremely simplified way to capture the relationships. The supply parameters of a specific segment may also have significant enough influence on other neighboring segments. A better way to calculate the weights would be using the numerical approximation method (see Section 3.5, numerical approximation). Similar to the previous part, a weight is decided to distinguish the different magnitudes of influence to counts and speeds, while more weights are put on segment speeds.

The weights between the route choice model parameter and all the measurements are set to 1.

Because the weights are estimated in a relatively simple way to just capture the most significant relationships, there are concerns that this may lead to results that are not optimal due to neglecting other relationships. To solve this problem, all-one matrix (the SPSA) is used after reaching convergence with the computed weight matrix in order to capture all the potential relationships between parameters and measurements.
5.3.2 Empirical considerations

Parameter boundaries

Due to the lack of high quality of a priori parameter values, the deviations between estimated parameter values and a priori parameter values were not included in the objective function. However, to prevent parameter values that are too far away from the initial range, strict boundaries were applied during the calibration process. For example, without setting parameter boundaries, in order to fit the speed data, the calibration process will result in very high minimum speed or very low maximum speed for some segments. Over-fitting the data by setting unreasonable parameter values will lead to poor prediction performance. Therefore, setting strict and reasonable boundaries is a crucial first step for the calibration process.

Ratio versus non-ratio

In Balakrishna (2006), the author suggests a ratio perturbation method when applying SPSA. The author argues that because of the significant difference in magnitude of different parameter values, perturbing them with a same value is not applicable. For example, most of the OD flows are between 0 to a few hundreds, while the capacity parameter is a value usually between 0 and 10 (vehicle per second). Multiple perturbation step size magnitudes have to be decided for all the different parameters. A more convenient way is to perturb all the parameters with a same step size magnitude but in a ratio way. For example, the initial values for two parameters are 10 and 100. When the original non-ratio perturbation is applied and the perturbations are +5 for both of them, the values after perturbation become 15 and 105. When the ratio perturbation is applied, the initial ratios are 1. After a perturbations of +0.1, the ratios become 1.1 and 1.1, and the corresponding parameter values become 11 and 110.

This method works well when all the parameters have similar values. For example, the maximum speeds for all segments are between 45 and 70 miles per hour. The maximum value is less than twice of the minimum value. However, the method does
not work well when the distribution of the same type of parameters has a large range. For example, the maximum OD flow can be a few hundreds, while the minimum OD flow can be 1, or even 0. In this case, when doing the ratio based perturbation, the parameters with small values can never have any significant influence because their actual perturbation step sizes are so small (for a 10% perturbation, the actual step size of a OD flow of 5 is 0.5, while that of an OD flow of 100 is 10). Therefore, the estimation of gradient for these parameters are always extremely biased because all the captured changes are actually introduced by other larger parameters that influence the same sensor. In many cases, the OD flows that have small values at the beginning actually have every big values after proper calibration.

In this case study, the original non-ratio perturbation was applied for OD flows and the ratio perturbation was applied for other parameters.

**Fixed versus random**

In the SPSA algorithm, within each iteration, after determining a perturbation step size, the perturbation directions are generated randomly based on a selected distribution (Bernoulli in many cases). It was found in this case study that, when using W-SPSA, at the beginning stage of the calibration process, if the start parameter values are believed to be biased towards a specific direction, a fixed perturbation leads to much faster improvement of the objective function value. More specifically, the initial OD flows were found to be mostly smaller than true values (according to the simulated counts). Therefore, in each iteration, instead of perturbing all the OD flows based on a randomly generated vector (i.e., some + some - then some - some +), all the OD flows are perturbed in a same direction (i.e., all + then all -). This fixed perturbation was applied at the very beginning stage and only for OD flows. It dramatically accelerated the improvement of fit-to-counts. After a few iterations, random perturbation was applied for all parameters.

This finding is based on empirical experiments with the expressway network and dataset in this case study, no theoretical proof was done to prove its effectiveness for all cases.
Simplified step size

In the SPSA algorithm, two step sizes are used in each iteration: the perturbation step size \( (c_k) \) and the advance step size \( (a_k) \). They change with \( k \), the index of current iteration in a complex non-linear way. Although there are guidelines and heuristic rules to help decide the algorithm parameter values, it still needs a lot of trial-and-error to find some good values and it is always unknown if better values exist. Moreover, the meanings of the algorithm parameters are not very intuitive.

When applying W-SPSA for this case study, the calculation of the step sizes were simplified. No complex calculation of the advance step is required. The process advances the parameter values with full step size. In other words, after the two perturbations, if the sum of errors after perturbation 1 is smaller than that of perturbation 2, the parameter value is directly set to that in perturbation 2. Two strategies were tested to deal with the situation when there is no improvement in both of the perturbations. In the first strategy the parameter value was set to the one with better result, while in the second one it remained unchanged. Experiments showed that the second strategy achieved better convergence rate and final objective function value.

At the same time, no complex calculation of the perturbation step is required. There are two intuitive algorithm parameters that control the perturbation step size: 1) the initial step size, and 2) the reduce rate. The step sizes are reduced with specific rates decided for different parameters. Moreover, in the case study, two ways were tested for the perturbation step size reduction. In the first method, the perturbation step sizes were reduced after each iteration, while in the second method, the step size for each parameter reduced separately: for a parameter, if the corresponding summed error did not improve within the two perturbations, the perturbation step size for this parameter was reduced with the given rate. Otherwise it remained the same. The second method showed better performance.

This finding is based on empirical experiments with the expressway network and dataset in this case study, no theoretical proof was done to prove its effectiveness for all cases. Reducing the step size when there is no improvement might be effective.
when the initial values are close to the global optimum. Otherwise, the process is very likely to be trapped in a local optimum. Approaches to prevent local optimum by increasing the step size under certain situations can be studied further.

**Partial perturbation**

In this specific case study, there are more than 4000 OD pairs for each interval and many of them are with very small flows. If the perturbation was done on all of them in each iteration, there will be serious gridlock forming in the network and the calibration process will be influenced significantly. For example, if the perturbation step size is 10 for each OD, although there will be +10 and -10 for different OD flows, the actual number of vehicles entering the network will increase significantly in both of the two perturbations, because there are a great number of OD flows that are less than 10 and the lower bound for OD flow is 0.

Two methods can be used to address this problem. The first one only perturbs the OD flows that are larger than the current step size. This is not a good idea, because as we mentioned before, some OD flows with very small values may actually have very large values. If they are not perturbed at all, there will be no chance to increase them to the true values. The second idea randomly selects a proportion of the OD flows to perturb in each iteration. This method may lead to worse convergence rate, but it is unbiased. The second method was applied in this case study.

### 5.4 Results

Both SPSA and W-SPSA were implemented and applied in this case study. The W-SPSA results are based on implementing all the technical considerations in the previous section. However, many of the changes could not be applied to SPSA. For example, fixed perturbation does not make sense for SPSA because it can only choose one direction from the two perturbation directions and advance towards that direction with a specific step size, while W-SPSA’s advance direction can be totally different from the two perturbation directions. With fixed perturbation, the two
perturbation directions are of extremely bad quality if considering them as candidate advance directions. Therefore, two versions of SPSA were implemented. The first one is without any modification and the second one is after implementing non-ratio OD perturbation and partial perturbation.

Unfortunately, although great efforts were put on finding good parameter values for SPSA and running it to do the calibration, no meaningful improvement was achieved in terms of fit-to-counts and fit-to-speeds. Two reasons might explain the ineffectiveness of SPSA in the case study: 1) the expressway network is very sparse in terms of spatial correlations between parameters and measurements, and the number of intervals is very large, which lead to significant approximation errors (noises from irrelevant measurements) and the algorithm could not find meaningful gradients under this situation; 2) the algorithm parameters are still not set with good enough values. The second reason reveals another problem of SPSA: it has too many parameters to configure and it is very hard to compare this algorithm with others because no one can prove the current set of algorithm parameters is good enough. At the same time, only after a big enough number of iterations can we tell if the current set of parameters are better or worse than the previous one due to the stochastic nature of the algorithm, which leads to an extremely long trial-and-error process. On the contrary, with the modifications, W-SPSA requires only a few very intuitive algorithm parameters that can be set and tested very quickly.

W-SPSA achieved significant improvement within only a few tens of iterations. As mentioned in the weight matrix calculation part, experiments were done to apply SPSA after running W-SPSA in order to try capturing all the potential correlations between parameters and measurements. However, no meaningful improvement was achieved. The gradient estimation errors are so large that the influence of irrelevant measurements are significantly larger than the influence of omitted correlations.

Table 5.1 shows the overall calibration results using W-SPSA. Fit-to-counts is calculated as the RMSN between the simulated counts and observed counts over all sensors and intervals. Fit-to-speeds is calculated as the RMSN between the simulated speeds and observed speeds over all segments and all intervals. The fit-to-counts
measurement was improved by 49.3% and the fit-to-speeds was improved by 49.0%.

Table 5.1: Overall calibration results

<table>
<thead>
<tr>
<th></th>
<th>Before calibration</th>
<th>After calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit-to-counts</td>
<td>0.426</td>
<td>0.216</td>
</tr>
<tr>
<td>Fit-to-speeds</td>
<td>0.321</td>
<td>0.164</td>
</tr>
</tbody>
</table>

Fit-to-counts and fit-to-speeds by time interval are shown in Table 5.2 and Table 5.3, respectively.

Figure 5-7 shows the fit-to-counts at three different time intervals: 7:30-7:35, 8:30-8:35, and 9:00-9:05. The x-axis corresponds to the observed sensor counts in vehicles per 5 minutes. The y-axis corresponds to the simulated sensor counts in vehicle per 5 minutes after off-line calibration. Blue dots represents observed and simulated counts at a specific sensor. The red 45 degree line represents the perfect fit where the simulated counts exactly matches the observed counts. Most of the dots are very close to the 45 degree line, which indicates good fit.

While Figure 5-7 provides an intuitive way to evaluate how good the calibration results fit the observed data at different intervals, Figure 5-8 compares the cumulative counts at different sensors across the whole simulation period. The x-axis corresponds to the simulation interval ID, where 1 is the first interval after the warm-up period (7:00-7:05) and 36 is the last interval (9:55-10:00). The y-axis represents the cumulative counts at a sensor. It is found from the results that most of the sensors showed good fit, as shown in Figure 5-8. However, there were sensors at which the observed counts are constantly larger than simulated counts, or simulated counts are constantly larger than observed counts, as shown in Figure 5-9. This is somehow different from what is expected: the errors at a sensor across different intervals have a relatively random distribution, instead of biased to one direction. These results indicate the possibility of some systematic error in the network and data. For example, in the simulation, due to a missing on-ramp between two specific sensors, the simulated counts at these two sensors are always similar. However, in the data, if there is a big difference between the observed flow from these sensors (because of the on-ramp), the calibration process will make the flow at these two sensors to be somewhere between
Table 5.2: Fit-to-counts by time interval

<table>
<thead>
<tr>
<th>Time period</th>
<th>Before calibration</th>
<th>After calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:00-7:05</td>
<td>0.526</td>
<td>0.246</td>
</tr>
<tr>
<td>7:05-7:10</td>
<td>0.521</td>
<td>0.229</td>
</tr>
<tr>
<td>7:10-7:15</td>
<td>0.515</td>
<td>0.224</td>
</tr>
<tr>
<td>7:15-7:20</td>
<td>0.507</td>
<td>0.217</td>
</tr>
<tr>
<td>7:20-7:25</td>
<td>0.488</td>
<td>0.219</td>
</tr>
<tr>
<td>7:25-7:30</td>
<td>0.495</td>
<td>0.220</td>
</tr>
<tr>
<td>7:30-7:35</td>
<td>0.479</td>
<td>0.213</td>
</tr>
<tr>
<td>7:35-7:40</td>
<td>0.472</td>
<td>0.228</td>
</tr>
<tr>
<td>7:40-7:45</td>
<td>0.461</td>
<td>0.212</td>
</tr>
<tr>
<td>7:45-7:50</td>
<td>0.443</td>
<td>0.228</td>
</tr>
<tr>
<td>7:50-7:55</td>
<td>0.431</td>
<td>0.219</td>
</tr>
<tr>
<td>7:55-8:00</td>
<td>0.419</td>
<td>0.231</td>
</tr>
<tr>
<td>8:00-8:05</td>
<td>0.414</td>
<td>0.223</td>
</tr>
<tr>
<td>8:05-8:10</td>
<td>0.412</td>
<td>0.219</td>
</tr>
<tr>
<td>8:10-8:15</td>
<td>0.401</td>
<td>0.219</td>
</tr>
<tr>
<td>8:15-8:20</td>
<td>0.404</td>
<td>0.221</td>
</tr>
<tr>
<td>8:20-8:25</td>
<td>0.396</td>
<td>0.216</td>
</tr>
<tr>
<td>8:25-8:30</td>
<td>0.389</td>
<td>0.208</td>
</tr>
<tr>
<td>8:30-8:35</td>
<td>0.400</td>
<td>0.200</td>
</tr>
<tr>
<td>8:35-8:40</td>
<td>0.419</td>
<td>0.218</td>
</tr>
<tr>
<td>8:40-8:45</td>
<td>0.439</td>
<td>0.229</td>
</tr>
<tr>
<td>8:45-8:50</td>
<td>0.458</td>
<td>0.199</td>
</tr>
<tr>
<td>8:50-8:55</td>
<td>0.473</td>
<td>0.218</td>
</tr>
<tr>
<td>8:55-9:00</td>
<td>0.486</td>
<td>0.190</td>
</tr>
<tr>
<td>9:00-9:05</td>
<td>0.490</td>
<td>0.207</td>
</tr>
<tr>
<td>9:05-9:10</td>
<td>0.503</td>
<td>0.201</td>
</tr>
<tr>
<td>9:10-9:15</td>
<td>0.498</td>
<td>0.215</td>
</tr>
<tr>
<td>9:15-9:20</td>
<td>0.501</td>
<td>0.206</td>
</tr>
<tr>
<td>9:20-9:25</td>
<td>0.504</td>
<td>0.207</td>
</tr>
<tr>
<td>9:25-9:30</td>
<td>0.506</td>
<td>0.192</td>
</tr>
<tr>
<td>9:30-9:35</td>
<td>0.504</td>
<td>0.199</td>
</tr>
<tr>
<td>9:35-9:40</td>
<td>0.508</td>
<td>0.212</td>
</tr>
<tr>
<td>9:40-9:45</td>
<td>0.507</td>
<td>0.220</td>
</tr>
<tr>
<td>9:45-9:50</td>
<td>0.502</td>
<td>0.219</td>
</tr>
<tr>
<td>9:50-9:55</td>
<td>0.504</td>
<td>0.244</td>
</tr>
<tr>
<td>9:55-10:00</td>
<td>0.502</td>
<td>0.216</td>
</tr>
<tr>
<td>Time period</td>
<td>Before calibration RMSN</td>
<td>After calibration RMSN</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>7:00-7:05</td>
<td>0.174</td>
<td>0.146</td>
</tr>
<tr>
<td>7:05-7:10</td>
<td>0.172</td>
<td>0.149</td>
</tr>
<tr>
<td>7:10-7:15</td>
<td>0.177</td>
<td>0.148</td>
</tr>
<tr>
<td>7:15-7:20</td>
<td>0.175</td>
<td>0.145</td>
</tr>
<tr>
<td>7:20-7:25</td>
<td>0.182</td>
<td>0.142</td>
</tr>
<tr>
<td>7:25-7:30</td>
<td>0.179</td>
<td>0.141</td>
</tr>
<tr>
<td>7:30-7:35</td>
<td>0.179</td>
<td>0.142</td>
</tr>
<tr>
<td>7:35-7:40</td>
<td>0.178</td>
<td>0.141</td>
</tr>
<tr>
<td>7:40-7:45</td>
<td>0.178</td>
<td>0.138</td>
</tr>
<tr>
<td>7:45-7:50</td>
<td>0.176</td>
<td>0.137</td>
</tr>
<tr>
<td>7:50-7:55</td>
<td>0.181</td>
<td>0.139</td>
</tr>
<tr>
<td>7:55-8:00</td>
<td>0.178</td>
<td>0.139</td>
</tr>
<tr>
<td>8:00-8:05</td>
<td>0.185</td>
<td>0.142</td>
</tr>
<tr>
<td>8:05-8:10</td>
<td>0.187</td>
<td>0.136</td>
</tr>
<tr>
<td>8:10-8:15</td>
<td>0.190</td>
<td>0.127</td>
</tr>
<tr>
<td>8:15-8:20</td>
<td>0.217</td>
<td>0.109</td>
</tr>
<tr>
<td>8:20-8:25</td>
<td>0.237</td>
<td>0.114</td>
</tr>
<tr>
<td>8:25-8:30</td>
<td>0.245</td>
<td>0.112</td>
</tr>
<tr>
<td>8:30-8:35</td>
<td>0.250</td>
<td>0.114</td>
</tr>
<tr>
<td>8:35-8:40</td>
<td>0.260</td>
<td>0.114</td>
</tr>
<tr>
<td>8:40-8:45</td>
<td>0.274</td>
<td>0.115</td>
</tr>
<tr>
<td>8:45-8:50</td>
<td>0.291</td>
<td>0.121</td>
</tr>
<tr>
<td>8:50-8:55</td>
<td>0.299</td>
<td>0.123</td>
</tr>
<tr>
<td>8:55-9:00</td>
<td>0.308</td>
<td>0.131</td>
</tr>
<tr>
<td>9:00-9:05</td>
<td>0.344</td>
<td>0.143</td>
</tr>
<tr>
<td>9:05-9:10</td>
<td>0.368</td>
<td>0.154</td>
</tr>
<tr>
<td>9:10-9:15</td>
<td>0.379</td>
<td>0.158</td>
</tr>
<tr>
<td>9:15-9:20</td>
<td>0.388</td>
<td>0.166</td>
</tr>
<tr>
<td>9:20-9:25</td>
<td>0.403</td>
<td>0.175</td>
</tr>
<tr>
<td>9:25-9:30</td>
<td>0.431</td>
<td>0.191</td>
</tr>
<tr>
<td>9:30-9:35</td>
<td>0.451</td>
<td>0.213</td>
</tr>
<tr>
<td>9:35-9:40</td>
<td>0.453</td>
<td>0.215</td>
</tr>
<tr>
<td>9:40-9:45</td>
<td>0.462</td>
<td>0.222</td>
</tr>
<tr>
<td>9:45-9:50</td>
<td>0.457</td>
<td>0.220</td>
</tr>
<tr>
<td>9:50-9:55</td>
<td>0.462</td>
<td>0.226</td>
</tr>
<tr>
<td>9:55-10:00</td>
<td>0.462</td>
<td>0.226</td>
</tr>
</tbody>
</table>
Figure 5-7: Fit-to-counts at different intervals
the two values given by the two sensor to minimize the errors. Therefore, the errors at one sensor are always constantly positive while the errors at the other sensor are always negative. Inconsistencies in the data will also lead to this result.

![Graphs of Sensor 17 and Sensor 90](image)

(a) Sensor 17  
(b) Sensor 90

Figure 5-8: Cumulative counts at different sensors

### 5.5 Summary

In this chapter, W-SPSA is implemented, modified, and applied in a real world large scale case study to validate its performance and scalability. DynaMIT, a mesoscopic DTA system is calibrated for the entire expressway network of Singapore. The field data, especially the sensor counts data are analyzed and processed before the cali-
Figure 5-9: Some sensor errors that may caused by systematic network/data issues
ibration process. Several empirical considerations and modifications of the W-SPSA algorithm for this case study are discussed in detail, including how the weight matrix is calculated for this network and dataset. SPSA and W-SPSA are implemented and applied in the off-line calibration process. W-SPSA outperforms SPSA significantly and achieves great improvement over the reference case. Results show that after the off-line calibration, the DTA system is able to reproduce the observed traffic condition with a high level of accuracy.
Chapter 6

Conclusion

6.1 Summary

The off-line calibration is a crucial process for the successful application of simulation-based dynamic traffic assignment (DTA) systems in real world traffic networks. Among the various frameworks proposed to solve this problem, the simultaneous demand-supply calibration is chosen by the author due to its flexibility to utilize different types of traffic data and its capability of capturing the important interactions between demand models and supply models. Under this framework, the DTA off-line calibration is formulated as a stochastic optimization problem. Simultaneous perturbation stochastic approximation (SPSA) has been reported in many literatures to be the most suitable solution algorithm for this problem due to its scalability in terms of computation time. However, in a case study for the entire expressway network of Singapore, the performance of SPSA was found to deteriorate when the size of the network and the number of simulation intervals increase. No satisfactory results were obtained by applying SPSA or its different modified versions.

This thesis first analyzes the reasons for the performance deterioration of SPSA. It is found that the sparsity between parameters and measurements in a traffic network introduces significant approximation error in SPSA's gradient estimation process. Two types of network sparsity are identified: 1) spatial sparsity, and 2) temporal sparsity. Spatial sparsity is caused by the relatively local influence of parameters...
in the traffic network. One parameter is usually only able to influence a very small proportion of all the measurements in the network. Temporal sparsity is caused by the fact that a parameter in one interval never influences measurements in previous intervals and only influences measurements in a limited number of future intervals. Although parameters and measurements are extremely sparsely correlated in a real world traffic network, SPSA considers all the measurements when estimating the gradient, or the influence brought by one specific parameter in the model. The signals from irrelevant measurements introduce noise, which lead to errors in the gradient estimation process. It is estimated that in a real world application with a reasonable number of intervals, more than 99% of the signals are actually noise, which makes the estimation of gradients impossible.

To address the problem of gradient estimation errors, an enhanced algorithm is proposed in this thesis: Weighted SPSA, or W-SPSA. Rather than using information from all the sensors, W-SPSA estimates the gradient only using information that is relevant to a specific parameter and weighs this information based on the degree of relevance. A weight matrix that stores all the weights for the gradient estimation needs to be estimated and updated during the calibration process. Several methods to estimate the weight matrix are proposed, including analytical approximation, numerical approximation, all-one matrix, and hybrid methods. Their advantages and shortcomings are discussed. It is also noted that the original SPSA algorithm is a special case of the W-SPSA algorithm when all the weights are set to 1.

Then, a synthetic test system is built to compare the performance of SPSA against W-SPSA and to demonstrate W-SPSA's characteristics. W-SPSA outperforms SPSA in terms of both convergence rate and long run accuracy. It is found that although SPSA is scalable in terms of computation time, its convergence and accuracy performance deteriorates significantly when the scale of the problem increases. On the other hand, W-SPSA is found to be scalable in terms of both computational time and accuracy. Moreover, under all network correlation levels, W-SPSA is found to outperform SPSA. Several characteristics of W-SPSA are discovered, including its insensitivity to accurate values in the weight matrix (only 0-1 is sufficient for good results). Ways
to estimate the weight matrices are discussed, specifically, guidelines for applying the numerical approximation method are proposed based on experimental results. The noises in the simulator and data are found to influence the calibration process and final results.

Finally, a case study is presented with the entire expressway network in Singapore using DynaMIT, a mesoscopic simulation-based DTA system. More than 160,000 time-dependent model parameters are calibrated simultaneously using field sensor counts and speed data. A data cleaning process is introduced before detailed discussions on the calibration process. Empirical technical considerations when applying the W-SPSA algorithm for this case study are discussed. Several modifications to the existing algorithm are implemented and tested considering the specific problem characteristics in this case study. Results show that W-SPSA outperforms SPSA significantly in this real world case study by achieving great improvement in terms of goodness-to-fit to the observed sensor counts and speed data. Although there are still systematic errors that come from data inconsistencies and network coding errors, the overall results show that the calibrated system is capable of replicating the real world traffic conditions with a high level of accuracy.

6.2 Future research directions

- Parallel W-SPSA
  The off-line calibration using W-SPSA can be parallelized at two different levels. At the upper level, within an iteration, the two runs of the simulator are completely independent. Running them with two computers/processors at the same time and collecting the outputs for the gradient estimation afterwards can lead to significant saving of computation time. At the lower level, within the gradient estimation process, the calculations of each element in the gradient vector, the perturbation of each parameter (if partial perturbation and adaptive step size are implemented) are independent and can be parallelized. Although this is a relatively small time saving compared to parallel running of
the simulator, it is still considerable when the problem scale is large.

- Applying numerical weight matrix approximation in real world case studies
  In the case study in this thesis, the weight matrix is estimated solely using the analytical method. The process is highly simplified for some parameter-measurement pairs (e.g., OD flows vs. speeds, supply parameters vs. sensor counts), which may lead to the omission of important correlations. The numerical method is believed to be a better candidate to estimate the weights between such parameter-measurement pairs. However, due to the limitation of computational resource, the computationally intensive numerical method is not applied in this case study. Although its effectiveness is proved in the synthetic tests, it is suggested that it should be applied and tested in real world case studies in the future.

- Applying weighted gradient approximation in the on-line calibration process
  The weighted gradient approximation idea in W-SPSA is also applicable in the on-line calibration process. Specifically, with the state-space formulation of the on-line calibration problem, the Extended Kalman Filter algorithm is widely applied (Antoniou, 2004). However, this method has the issue of scalability due to the extremely heavy computational burdens associated with the Jacobi matrix estimation. With the simultaneous perturbation and weighted gradient approximation idea, the Jacobi matrix estimation process can be significantly accelerated while remains high level of accuracy.

- Incorporating novel data types into the off-line calibration
  Nowadays more and more emerging types of data are available to transportation researchers and agencies, which provide an opportunity to understand and estimate the influence of factors that are rarely considered in previous studies. For example, with the abundance of information on the Internet and text mining techniques, the information for special events (e.g., concerts, sports games), including their location, time, and potential popularity can be obtained and archived automatically. Using these data, the influence of such events to de-
mand and supply can be modeled in the DTA system and calibrated for real time applications.
Bibliography


FHWA (2013a). CORSIM software. 
URL: http://ops.fhwa.dot.gov/trafficanalysis/tools/corsim.htm

FHWA (2013b). Transmodeler software. 
URL: http://www.caliper.com/transmodeler/default.htm


