Computationally Efficient Offline Demand Calibration Algorithms for Large-scale Stochastic Traffic Simulation Models

by

Chao Zhang

B.E., Zhejiang University (2011)
M.S., University of Massachusetts Lowell (2013)

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

Doctor of Philosophy in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

Author .....
Department of Civil and Environmental Engineering
August 17, 2018

Certified by .....
Carolina Osorio
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by .....
Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee

Computationally Efficient Offline Demand Calibration Algorithms for Large-scale Stochastic Traffic Simulation Models

by

Chao Zhang

Submitted to the Department of Civil and Environmental Engineering
on August 17, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Transportation

Abstract

This thesis introduces computationally efficient, robust, and scalable calibration algorithms for large-scale stochastic transportation simulators. Unlike a traditional "black-box" calibration algorithm, a macroscopic analytical network model is embedded through a metamodel simulation-based optimization (SO) framework. The computational efficiency is achieved through the analytical network model, which provides the algorithm with low-fidelity, analytical, differentiable, problem-specific structural information and can be efficiently evaluated. The thesis starts with the calibration of low-dimensional behavioral and supply parameters, it then addresses a challenging high-dimensional origin-destination (OD) demand matrix calibration problem, and finally enhances the OD demand calibration by taking advantage of additional high-resolution traffic data. The proposed general calibration framework is suitable to address a broad class of calibration problems and has the flexibility to be extended to incorporate emerging data sources.

The proposed algorithms are first validated on synthetic networks and then tested through a case study of a large-scale real-world network with 24,335 links and 11,345 nodes in the metropolitan area of Berlin, Germany. Case studies indicate that the proposed calibration algorithms are computationally efficient, improve the quality of solutions, and are robust to both the initial conditions and to the stochasticity of the simulator, under a tight computational budget. Compared to a traditional "black-box" method, the proposed method improves the computational efficiency by an average of 30%, as measured by the total computational runtime, and simultaneously yields an average of 70% improvement in the quality of solutions, as measured by its objective function estimates, for the OD demand calibration. Moreover, the addition of intersection turning flows further enhances performance by improving the fit to field data by an average of 20% (resp. 14%), as measured by the root mean square normalized (RMSN) errors of traffic counts (resp. intersection turning flows).

Thesis Supervisor: Carolina Osorio
Title: Associate Professor of Civil and Environmental Engineering

Acknowledgments

First and foremost, I am wholly indebted to Professor Carolina Osorio, my research advisor, who has been supportive of my Ph.D. research with her immense knowledge, rigorous scientific attitude, encouraging advice, and continuous guidance. The accomplishment of this research and the writing of this thesis would not have been possible without her. She has been a top role model for my development as a scientist, and I could not have imagined having a better advisor and mentor for the past five years.

My sincere gratitude to the rest of my thesis committee: Professor Nigel Wilson and Professor John Williams, for their challenging questions, insightful comments, and endless encouragement. Their advice has helped me to broaden my research horizon, to better present my work, and to rethink my research from a practical perspective.

My heartfelt thanks also go to Professor Gunnar Flötteröd at the Royal Institute of Technology (KTH) for preliminary discussions on this work, Professor Kai Nagel and Dominik Ziemke at the Technical University of Berlin (TU Berlin) for providing the data, and the U.S. National Science Foundation (NSF Grant No. 1334304) for partially funding this research. Without their precious support, it would not be possible to conduct this research.

I thank my fellow labmates in Osorio lab: Linsen Chong, Tianli Zhou, Jing Lu, Kevin Zhang, Timothy Tay, Evan Fields, and Xiao Chen; my colleagues Zhan Zhao, Yin Wang, Haizheng Zhang, Nate Bailey, Yafei Han, Yan Leng, Chiwei Yan, and Yiwen Zhu; and my friends Siyuan Lu and Vivek Sivathanu at MIT for the stimulating discussions on research and career, for the friendly advice on life decisions, and for truckloads of good times we shared in the last five years.

I thank Marilyn Levin and Thalia Rubio at the Writing and Communication Center (WCC) for their constructive feedback on my writing and caring advice on both career and life; Jake Livengood at the Global Education and Career Development (GECD) for the valuable time he spent with me discussing how I can achieve my career goals; and Roberta Pizzinato, Eunice Kim, Katie Rosa, Max Martelli, Kathleen

Briana, Kiley Clapper, and Sarah Smith, the administrative staff at the Department of Civil and Environmental Engineering (CEE), for their generous assistance.

I would also like to extend my sincere appreciation to my former advisors: Professor Yuanchang Xie and Professor Nathan Gartner at the University of Massachusetts Lowell, who initiated me into the research area of Transportation, stimulated my interests in pursuing a Ph.D. in it, and have supported me continuously. Also I thank my former colleague and friend Tugba Arsava for her kindness, support, generosity, and the positive influence she has had on me.

Last but not least, I would like to thank my parents Fengxian Shi and Guangqi Zhang for giving me the life every child deserves and all the sacrifices they've made and my girlfriend Zhuo Chen for supporting me spiritually throughout writing this thesis and my life in general.

Contents

Chapter 1

Introduction

1.1 Motivation and Objective

Traffic congestion is a severe and growing problem worldwide, especially in urban areas. One solution is to expand traffic infrastructure, but this requires public land, sufficient funds, and costly labor, and this investment may even worsen the situation. A more feasible and cost-effective solution is to deploy models to strategically manage traffic over space and time. Transportation professionals and the research community rely on these models, with different ranges of complexity, comprehensiveness, and potential usefulness, to optimize traffic management strategies, evaluate potential impacts, and forecast future traffic conditions. Compared to analytical traffic models, traffic simulation models describe complex traffic dynamics through detailed representations of demand. The most detailed models represent the traffic system at the scale of individual travelers. This comes at the expense of increased model complexity and computational inefficiency. Barceló (2010) provides a review of the state-of-the-art for traffic simulation models.

These high-resolution models typically have a large number (e.g., thousands) of input parameters. The process of setting optimal values for the input parameters so as to obtain a close match between the simulation-based performance metrics and the field traffic measurements, such as flow, speed, and occupancy, is called calibration. Calibration is a critical prerequisite for simulators prior to their application. System-

atically calibrating traffic simulators is important, yet mathematically difficult.

Designed to achieve asymptotic convergence, traditional calibration algorithms are computationally inefficient general-purpose algorithms that require a large number of simulation evaluations. On the one hand, these algorithms are generic and can be applied to any simulation model (e.g., finance, biomechanics, and maritime). On the other hand, they treat the simulator as a "black-box", capturing little to no structural information of the underlying problem. Given very limited computational budgets in practice, these algorithms are of little use, so there is a need for computationally efficient calibration algorithms that can identify solutions with good performance at low computational cost (i.e., within a few simulation evaluations).

In this thesis, we formulate the demand calibration of large-scale traffic simulation models as a simulation-based optimization (SO) problem, and develop computationally efficient metamodel SO approaches. The metamodels embed transportation problem-specific structural information derived from low-fidelity analytical network models, which significantly boosts the computational efficiency and the quality of the solutions of the calibration algorithms. In other words, the proposed methods benefit from both the realistic representation of congested traffic systems provided by traffic simulation models and from the computational tractability of analytical traffic models. This thesis is structured as follows. Chapter 2 provides an extensive overview of the state-of-the-art demand calibration algorithms and data sources and introduces the SO framework, upon which the algorithms of the following chapters are developed. Chapter 3 designs an algorithm for a low-dimensional route choice behavioral parameter calibration problem, which is then further applied to calibrate supply parameters in Chapter 4; then, we present in Chapter 5 a more challenging high-dimensional origin-destination (OD) demand matrix calibration. In the end, the proposed methodology is further extended to incorporate additional high-resolution data sources in Chapter 6. As a first step, we demonstrate the extensibility of the proposed methodology by exploring the possibility and benefits of integrating a new data source, namely turning traffic flows at intersections.

1.2 Contributions

This thesis formulates computationally efficient demand calibration algorithms for high-resolution large-scale stochastic traffic simulation models. By doing so, It contributes both theoretical and practical advances in transportation.

Theoretically, the broad family of metamodel SO methods have been used to efficiently address large-scale urban traffic management problems while using inefficient yet detailed stochastic high-resolution simulation models. This thesis extends metamodel SO ideas to the field of model calibration. The main contributions of this thesis are summarized as follows.

Computational efficiency We propose calibration algorithms that can identify points with good performance within a few algorithmic iterations. Therefore, they are computationally efficient algorithms that reflect well the computational conditions under which calibration problems are addressed by both the transportation research and practice communities. This is achieved by designing algorithms that exploit the transportation-specific structure of the calibration problem. More specifically, the proposed approach solves at every iteration of the calibration algorithm, an analytical (i.e., not simulation-based) approximate calibration problem. This analytical problem is solved by using information from an analytical network model. The latter is computationally efficient. It is formulated as a system of analytical and differentiable equations. Hence, it can be evaluated with a variety of efficient solvers. This is key for the efficiency of the calibration algorithms.

The proposed approach resorts to the use of a derivative-free algorithm. In other words, it does not rely on estimates of the derivatives of the simulation-based objective function. This further contributes to the efficiency of the algorithm.

Analytical network models The analytical network model, which provides problem-specific structural information, is the key to improving computational efficiency of the calibration algorithms. This contributes to a largely unresolved methodological challenge which is the formulation of tractable analytical measurement

equations that link available surveillance field data to the simulator's calibration parameters. Specifically, we formulate analytical network models that are simple, tractable, and specifically suited for each given calibration problem. In Chapter 3, the analytical network model with endogenous assignment is formulated based on a probabilistic queueing-based analytical traffic model, which is then reduced to one with exogenous assignment in Chapter 4. In Chapter 5, the analytical network model with exogenous assignment is further simplified to a system of linear equations. In Chapter 6, the analytical network model with endogenous assignment is formulated as a system of nonlinear equations.

Robustness The use of the analytical network models improves the robustness of the calibration algorithms to both the quality of initial points and to the stochasticity of the simulation models under various demand and supply scenarios. Hence, the performance of the proposed algorithms is similar for various initial points, as well as for various algorithmic runs with the same initial conditions.

Scalable algorithms The proposed algorithms are suitable for the efficient calibration of large-scale networks. This is achieved with the formulation of scalable analytical network models. For example, the analytical network model in Chapter 3 is defined as a system of nonlinear equations with a dimension that scales linearly with the number of links. Moreover, the analytical network model in Chapter 5 is formulated as a simple system of linear equations, the dimension of which scales linearly with the number of links and independently of other link attributes (e.g., link length) and of the dimension or structure of the OD matrix.

Stochasticity The algorithms are simulation-based optimization algorithms that account for the simulator's stochasticity, making them suitable for use with either probabilistic or deterministic models.

Capability of handling heterogeneous data The proposed calibration framework is capable of handling multiple heterogeneous data inputs. This enables the pro-

posed method to take advantage of additional high-resolution data types from emerging technologies (e.g., automatic vehicle identification (AVI)) to address the calibration problems more effectively. This further sheds light on the development of enhanced calibration algorithms that leverage big data with higher resolution in the future.

Added value for general-purpose SO algorithms Given the good performance of the proposed algorithms under tight computational budgets, they can be used as techniques to identify good initial solutions to launch or initialize traditional general-purpose calibration algorithms. They can serve to accelerate the convergence of general-purpose algorithms. This is particularly important when using the general-purpose algorithms under tight computational budgets, which are by design sensitive to the initial conditions.

In more practical terms, the proposed computationally efficient calibration algorithms enable the transportation community to more effectively and consistently calibrate inefficient high-resolution stochastic traffic simulation models. The proposed calibration algorithms are scalable, efficient, and robust compared to general-purpose "black-box" calibration algorithms. The proposed methodology is validated on real case studies. The scalability is demonstrated through case studies based on a large-scale network and field data in the metropolitan area of Berlin, Germany. The network includes 24,335 links and 11,345 nodes with traffic count data from 346 sensors. The proposed algorithms are shown to be computationally efficient and applicable to large-scale real scenarios.

1.3 Structure

This thesis focuses on the problem of offline demand calibration by dealing with different types of input parameters, such as behavioral parameters and OD demand matrices. It is structured around four journal papers. The outline of this thesis is presented as follows and for each chapter reference to the publication is provided.

Chapter 2 describes the context and the background of this research, reviews the state-of-the-art in demand model calibration, and introduces the general metamodel SO framework, which lays the foundation for the rest of this thesis.

Chapter 3 develops the metamodel SO demand calibration algorithm based on the metamodel SO framework introduced in Section 2.3 and demonstrates it through a low-dimensional route choice behavioral parameter calibration problem. The methodology and the case studies presented in this chapter have been published as:

Zhang, C., Osorio, C., and Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B: Methodological*, 97, 214-239.

Osorio, C., Flötteröd, G., and Zhang, C. (2015). A metamodel simulation-based optimization approach for the efficient calibration of stochastic traffic simulators. *Transportation Research Procedia*, 6, 213-223.

Chapter 4 further extends the algorithm developed in Chapter 3 to the calibration of two parameters, namely flow capacity and space capacity, in the supply models. The methodology and case studies shown in this chapter have been presented and published as:

Zhang, C., Osorio, C., and Flötteröd, G. (2016). An efficient algorithm for the supply calibration of large-scale stochastic traffic simulators. In *Proceedings of the Symposium of the European Association for Research in Transportation (hEART).*

Chapter 5 tackles the high-dimensional problem of calibrating OD demand matrices. The general methodology presented in Section 3 is extended for this calibration problem.

Chapter 6 enhances the OD demand calibration algorithm presented in Chapter 5 by leveraging several heterogeneous data sources including high-resolution emerg-

ing data. As an example, intersection turning traffic flow information is considered and its added value is illustrated through a case study on a real network.

Chapter 7 summarizes the main conclusions and implications of this thesis, recommends areas of future work on this topic, and provides additional insights on related problems, .

Appendices provide supplementary information regarding the description of the SO calibration algorithms, additional experimental results, and the implementation details for Chapters 3 through 6.

Chapter 2

Literature Review and Methodology

2.1 Background

Traffic models describe the dynamics of the transportation system through demand and supply models, which interact with each other. Overall, they can be categorized into: analytical models and simulation models. Analytical models are flow-based methods that describe the interactions of traffic at an aggregate level, thus are less expensive to evaluate but less realistic. Their applications are usually limited to small networks under strong assumptions (e.g., simplified travelers' behaviors). With the rapid evolution of transportation infrastructure and the introduction of next-generation mobility (e.g., autonomous vehicles), higher-resolution models, such as simulation models, are needed to capture the ever-increasing complexity of the transportation system. In traffic simulation models, the demand models estimate activity or trip patterns (i.e., origin-destination (OD)) and simulate the behaviors of individual travelers (e.g., travel mode, route choice, car following, and lane changing). On the other hand, supply models represent in detail the capacity of network elements (e.g., link, node). Examples of traffic simulation models are DynaMIT (Milkovits *et al.*, 2010), MITSIMLab (Ben-Akiva *et al.*, 2010), DYNASMART (Zhou and Mahmassani, 2006), Aimsun (Barceló, 2010), VISSIM (PTV, 2008), and MATSim (Horni *et al.*, 2016).

In this thesis, we use the MATSim traffic simulation model (Multi-Agent Trans-

port Simulation). However, the methodology developed in this thesis is suitable for a broad family of transportation simulation models. MATSim is a large-scale multi-agent activity-based traffic simulation model. Each iteration of the simulation in MATSim mimics the journeys of individual travelers (i.e., agents) given the transportation infrastructure in one virtual day. Each agent has socio-demographic attributes (e.g., age, gender, employment status, home location, and car availability) and performs a series of activities and trips. By interacting with other travelers, each traveler tries to make the best possible choices (e.g., varying departure time, travel mode, and route choice) based on a utility function. The next day, travelers make decisions aiming to achieve a better utility based on their travel experience from the previous day. The simulation stops once the total utility cannot be further increased in the day-to-day replanning process (i.e., the equilibrium state). Figure 2-1 shows the simulation process in MATSim.

Calibration, which is to infer model inputs and parameters based on traffic data, is the key to enable the successful application of high-resolution simulation models. A complete calibration of traffic simulators involves estimating input parameters for both demand models and supply models. In each single model embedded in the traffic simulator, there are numerous input parameters to be determined. Supply calibration determines parameters of supply models such as free-flow speed, jam density, and flow capacity. Demand calibration includes, for instance, the estimation of behavioral pa-

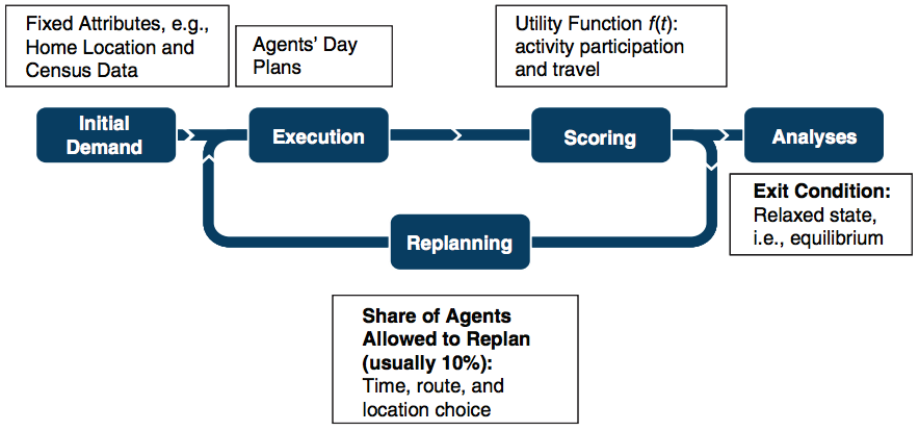


Figure 2-1: A typical cycle of MATSim simulation. Source: Horni *et al.* (2016)

rameters and of OD demand matrices. The latter is needed for planning, monitoring, and management of traffic. It contains the expected number of trips made during a certain time period between origin zones and destination zones. The problem of OD demand calibration is one of the main problems in modern traffic management systems and intelligent transportation systems and it has received a large amount of attention from researchers and practitioners alike. This thesis focuses on the topic of demand calibration of simulation-based road transportation models.

The calibration of traffic simulation models is challenging. Traffic simulation models are often stochastic and can involve sampling for every traveler from a variety of disaggregate behavioral models (e.g., choice models such as departure time, mode, route, lane-changing, etc.). Thus, a single run of the simulator involves drawing, for each of the thousands or hundreds of thousands of travelers, from a set of behavioral distributions. Given a sample of behavioral choices, a traffic flow model is used to propagate the travelers throughout the network. Thus, the mapping between the calibration input parameters and the objective function of a calibration problem is an intricate function. In particular, the aforementioned mapping is usually non-convex and contains multiple local optima. Thus, it is difficult to identify good solutions. The stochasticity of the simulator requires the use of optimization algorithms that account for the lack of both: (i) a closed-form expression of the objective function, and (ii) exact function evaluations (since the functions can only be estimated via simulation). Hence, the traditional approach to calibration has been the use of simulation-based optimization (SO) algorithms.

This thesis focuses on the calibration of simulators that are computationally costly to evaluate. The high computational cost can be due to: (i) the simulation of high levels of demand along with a high-resolution representation of demand (e.g., disaggregate representation of travelers), (ii) the simulation of a large-scale network, (iii) the use of a stochastic simulator requiring the evaluation of numerous simulation replications, and (iv) the desire to evaluate performance under equilibrium conditions, which requires running sequentially multiple simulation-based assignment iterations (Nagel and Flötteröd, 2012).

2.2 State-of-the-art Demand Calibration

The calibration of traffic simulators has evolved rapidly over the past decade, fueled by the need for applications from real-time traffic operations to long-term transportation system planning. The problem of model calibration has been extensively studied within the transportation community.

2.2.1 Methods for Demand Calibration

Many early studies on calibration resorted to computationally inefficient manual adjustment and trial-and-error enumeration based on engineering judgment or prior experience rather than a systematic approach (e.g., Chu *et al.*, 2003; Hourdakis *et al.*, 2003; Gomes *et al.*, 2004). In accordance with the types of transportation models described in Section 2.1, the formulations and solution approaches of numerous existing algorithms for systematic model calibration mainly fall into two categories: the analytical approach and the simulation-based approach. Balakrishna (2006) provides a survey of both analytical and simulation-based calibration problems and algorithms. Additional reviews on demand calibration algorithms are available in Djukic (2014); Zhang *et al.* (2017) and Tympakianaki (2018). In the literature, analytical demand calibration problems are commonly referred to as demand estimation. Seminal work on demand estimation includes Cascetta and Nguyen (1988); Yang *et al.* (1992); Cascetta *et al.* (1993) and Yang (1995).

The recent research on demand calibration is summarized in Table 2.1. Each row of the table corresponds to one paper. For each paper, the table indicates whether the problem is formulated as: an OD demand calibration or a joint supply-demand calibration problem, an online or an offline problem, and a time-dependent problem. The table also indicates the scale of case studies, in terms of the number of nodes and the number of links, of the largest and/or real-world network, as well as the dimension of the calibration vector. Table cells are left empty whenever the dimension is not directly reported in the paper. All papers use field traffic counts, which are the most widely available type of traffic data, for calibration. The last column of the

table indicates the papers that consider additional types of field data. As indicated in this table, there have been few studies that have used the algorithms to calibrate large-scale network models. To the best of our knowledge, the work of Flötteröd *et al.* (2011) and of Verbas *et al.* (2011) considers the largest-scale network instances with over 24,000 nodes and over 60,000 links. Regarding additional field data, various new data types have been considered, such as the most popular link condition measurements (e.g., speeds, densities) collected by conventional sensors and the emerging higher-resolution data sources (e.g., subpath travel times, route choice probabilities) collected by advanced transportation infrastructure.

Online calibration problems are commonly formulated based on state-space models with the state defined as the calibration parameters, which is then solved using Kalman filtering techniques (e.g., Antoniou, 2004; Bierlaire and Crittin, 2004; Zhou and Mahmassani, 2007; Antoniou *et al.*, 2007; Huang, 2010). This thesis focuses on the offline demand calibration of stochastic traffic simulators. The most common approach to offline simulation-based OD demand calibration has been the use of general-purpose SO algorithms, such as simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992) and genetic algorithms (GA) (Holland, 1975). As a derivative-based SO algorithm, SPSA efficiently estimates first-order derivative information through simultaneous perturbation of all the parameter vector and thus requires only two computations of the objective function per algorithmic iteration. It has been used, for instance, in Balakrishna *et al.* (2007); Vaze *et al.* (2009); Lee *et al.* (2009); Cipriani *et al.* (2011); Ben-Akiva *et al.* (2012) and Nigro *et al.* (2018). The transportation research community has also recently proposed various extensions of the SPSA algorithm to enhance efficiency, including Cipriani *et al.* (2011); Antoniou *et al.* (2015); Lu *et al.* (2015) and Tympakianaki *et al.* (2015). As a derivative-free random search algorithm, GA is another commonly used general-purpose SO algorithm. It has been used, for instance, in Kim *et al.* (2001); Stathopoulos and Tsekeris (2004); Kattan and Abdulhai (2006) and Vaze *et al.* (2009) for addressing low-dimensional calibration problems. Other derivative-free algorithms applied for calibration include the Box-Cox algorithm (Box, 1965) used in Kunde (2002) and the SNOBFIT (Stable

Table 2.1: Recent demand calibration literature overview

	OD demand	Joint supply-demand	Online	Offline	Time-dependency	Nodes	Links	Dimension	Additional field data
Kim *et al.* (2001)	✓			✓		9	14	8	
Tavana (2001)	✓			✓	✓	178	441		
Zhou *et al.* (2003)	✓			✓	✓	31	80		
Antoniou (2004)		✓	✓		✓	15 on/off-ramps		80	Speeds, densities
Bierlaire and Crittin (2004)	✓		✓		✓	296	618	627	
Jha *et al.* (2004)	✓[1]			✓	✓	1,479	3,756	41,906	
Eisenman and List (2004)	✓			✓				10	OD flows
Antoniou *et al.* (2006)	✓		✓		✓	10	10	6	Subpath flows, subpath travel times
Kattan and Abdulhai (2006)	✓			✓	✓	30	50	400	
Nie (2006)	✓			✓		17	23	4	Link-to-link counts, path travel times
Zhou and Mahmassani (2006)	✓			✓	✓	31	80		Link-to-link counts
Balakrishna *et al.* (2007)		✓		✓	✓	243	606		
Hazelton (2008)	✓			✓	✓	21	50	1,190	
Zhang *et al.* (2008)	✓			✓	✓	29 on/off-ramps		928	Subpath travel times
Lee and Ozbay (2009)	✓			✓	✓	A one-way freeway			Speeds
Vaze *et al.* (2009)		✓		✓	✓	825	1,767	6,470	Subpath travel times
Huang (2010)		✓	✓		✓	56	85	638	Speeds, densities
Cipriani *et al.* (2011)	✓			✓	✓	221	734		Speeds
Flötteröd *et al.* (2011)	✓			✓	✓	24,180	60,492	187,484	
Frederix *et al.* (2011)	✓			✓	✓	39	56		
Verbas *et al.* (2011)	✓			✓	✓	28,406	68,490	10^6-10^8	
Barceló *et al.* (2012)	✓			✓	✓	142	232	85	Subpath travel times
Ben-Akiva *et al.* (2012)		✓		✓	✓	1,698	3,180	69,093	Link travel times
Zhou *et al.* (2012)	✓			✓	✓	4 on/off-ramps			Link trave times, link densities
Cao *et al.* (2013)	✓			✓	✓	57	138		Link speeds
Cipriani *et al.* (2014)	✓			✓	✓	8	8		Speeds, Path trave times
Djukic *et al.* (2015)	✓			✓	✓			3,249	Link travel times, link densities
Lu *et al.* (2015)		✓		✓	✓	831	1,040	373,646	
Tympakianaki *et al.* (2015)	✓			✓	✓	N.A.	1,101	1,848	
Nigro *et al.* (2018)	✓			✓	✓	16 paths		12	Path trave times, route choice probabilities
Tympakianaki (2018)	✓			✓	✓	N.A.	1,101	1,848	

[1] Demand parameters of driver behavior and route choice models are also calibrated.

Noisy Optimization by Branch and FIT) algorithm (Huyer and Neumaier, 2008) used in Balakrishna *et al.* (2007).

The generality of these SO algorithms stems from the fact that they treat the simulator as a "black-box". The main implication of this is that they are designed to achieve asymptotic (i.e., large-sample size) convergence properties. They are not designed to identify good solutions within a few algorithmic iterations, i.e., they are not computationally efficient. The repeated evaluation of high-resolution simulation models involves extremely costly computer runs, even though offline applications do not require real-time computation. Nonetheless, they are used under very limited computational budgets in practice. Given the high computational costs involved in evaluating the simulation models, there is a need for calibration algorithms that can identify solutions with good performance at low computational cost. It is also challenging to design SO algorithms that remain efficient for high-dimensional problems. In particular, OD demand calibration problems are of high dimension, e.g., in the order of thousands or tens of thousands of decision variables. To sum up, there is a pressing need for algorithms that can identify good solutions within a few algorithmic iterations and can be applied to calibrate high-resolution yet computationally expensive simulation models of large-scale transportation networks.

There have been recent efforts to design scalable demand calibration algorithms. One approach is based on the use of dimensionality reduction techniques, such as sensitivity analysis (e.g., Ge *et al.*, 2014; Ge and Menendez, 2014; Ciuffo and Azevedo, 2014; Ciuffo *et al.*, 2015; Ge and Menendez, 2016; Zhong *et al.*, 2016) and principal component analysis (PCA) (e.g., Djukic, 2014; Prakash *et al.*, 2017). To address the scalability issues, Frederix *et al.* (2014) propose a spatial network decomposition approach for offline demand calibration.

General-purpose SO algorithms exploit limited, or no, problem-specific structural information (e.g., at most they are based on numerical linearizations). Another line of research to derive computationally efficient algorithms is to exploit the structure of the underlying calibration problem. It aims to improve both scalability and computational efficiency by embedding problem-specific structural information that is

derived from an analytical network model. One recent work that does exploit problem structure within the calibration algorithm is that of Flötteröd *et al.* (2011). It formulates and embeds within the algorithm an analytical approximation of the first-order derivative of the simulator's measurement equation. This leads to significant reductions in the computational requirements of the algorithm. This thesis follows this line of research.

2.2.2 Data Sources for Demand Calibration

Conventional data sources such as link traffic counts collected by inductive loop detectors and historical demand information estimated from census reports, possibly outdated, have been the most common data sources used for demand calibration in the theoretical literature and in practical applications. Traffic counts are typically scarce (due to the limited number of loop detectors) and thus lead to a significantly under-determined problem. In addition, traffic counts may fail to effectively discern between a congested and uncongested traffic state of a link, because of the non-monotonic nature of the flow-density relationship. As a result, a good fit to traffic counts does not necessarily indicate a high quality calibration. In search of a physically plausible solution, researchers have continued to use additional information (column "Additional field data" in Table 2.1) on link conditions, such as speeds, travel times, and densities, to restrict or bind the solution space of the problem (i.e., reduce the number of local minima). Examples of such data sources included in demand calibration are, for example, speeds (Balakrishna, 2006; Lee and Ozbay, 2009; Cipriani *et al.*, 2011; Frederix *et al.*, 2011); link travel times (Ben-Akiva *et al.*, 2012); speeds and densities (Antoniou, 2004; Huang, 2010); and link travel times and densities (Zhou *et al.*, 2012). These additional measurements of link conditions can be obtained from conventional detectors (e.g., inductive loop detectors) and they provide information regarding traffic congestion. They can be simply included in the objective function (offline) or the measurement equations (online) of the calibration problem and have shown significant added value to the effectiveness of calibration.

Demand calibration keeps evolving by taking advantage of both methodologi-

cal advances and additional data types from emerging technologies. The new generation of calibration algorithms will be supported by big data (e.g., social media data, cell-phone traces, and global positioning system (GPS) trajectories) with higher resolution. Automatic vehicle identification (AVI) systems and automatic vehicle location (AVL) systems are examples of emerging technologies for probe vehicle re-identification and tracking. AVI technology, such as in-vehicle traffic sensors (e.g., GPS, global system for mobile communications (GSM)), roadside cameras, and radio frequency identification (RFID) tag/plate readers, allows the detection of vehicles across multiple locations as they traverse the network. AVL technology, such as electronic-toll collection devices, infrared cameras, Bluetooth, and Wi-Fi, may also provide (partial) direct measurement of OD flows (over a fraction of OD pairs), point-to-point travel times and traffic counts, route choice probabilities, vehicle paths, and turning flows. Recent demand calibration work uses path travel times, subpath travel times, route choice probabilities, turning flows, OD split fractions, and/or OD flows collected by on-board mobile/GPS devices (e.g., Caceres *et al.*, 2007; Yamamoto *et al.*, 2009; Cao *et al.*, 2013; Nigro *et al.*, 2018), Bluetooth/Wi-Fi devices (e.g., Barceló *et al.*, 2010, 2012; Djukic *et al.*, 2015), roadside cameras (e.g., Asakura *et al.*, 2000; Mishalani *et al.*, 2002), and tag/plate readers (e.g., Dixon and Rilett, 2002; Eisenman and List, 2004; Zhou and Mahmassani, 2006; Antoniou *et al.*, 2006; Sun and Feng, 2011; Cipriani *et al.*, 2014). Reviews of heterogeneous traffic data used for calibration are presented in Djukic (2014) and Tympakianaki (2018). This thesis aims to develop computationally efficient algorithms for demand calibration that leverage several heterogeneous data sources.

Conventional detectors, such as inductive loop detectors, collect aggregate information regarding link conditions, such as traffic counts, speeds, and densities, for all vehicles. However, their deployment is only limited to specific locations (i.e., a small subset of links), which barely cover the entire network, due to high installation and maintenance costs. Compared to conventional data types, advanced technologies, such as GPS, roadside cameras, and RFID tag/plate readers, enable the collection of more detailed data, such as path travel times, route choice probabilities, and turning

flows. They contain information for the entire or partial journey and provide additional information on route choice behavior or individual mobility that can integrate conventional link condition data. As a result, they can be useful in reducing the complexity of the calibration problem. However, they may provide biased measurements since only a fraction of traffic is captured (e.g., probe vehicles). In addition, they suffer from issues such as limited network coverage, misidentification, and fears in data protection and privacy (Djukic, 2014, Section 2.3.1). In terms of cost, roadside video or image cameras are costly, which prevents them from large deployment. As more cost-effective ways for advanced data collection, RFID, Bluetooth, and GPS just require the installation of on-board electronic devices, although they currently cannot effectively cover the entire transportation network due to the low market penetration of required devices (e.g., smart mobile phones) (Herrera *et al.*, 2010). The latter leads to the current use of GSM data, such as the call detail records (CDR), by leveraging the existing communication infrastructure for demand calibration (Iqbal *et al.*, 2014). GSM data are usually incomplete and are not as high-resolution as GPS data. Nevertheless, these issues will be ameliorated as these technologies increasingly penetrate the market in the near future. Mimbela and Klein (2000, Chapter 3) provide a comprehensive discussion on the pros and cons of various vehicle detection and surveillance technologies.

The methodology in this thesis is able to leverage additional high-resolution data to enhance the performance of the calibration algorithm. Specifically, Chapter 6 of this thesis uses turning traffic flows at intersections to enhance an SO calibration and demonstrates that the enhanced calibration algorithm further improves the quality of solutions, maintains the computational efficiency of the calibration algorithm, is robust to both the initial points and the stochasticity of simulators, and is capable of handling multiple available data sources. This sheds light on the development of enhanced calibration frameworks for large-scale stochastic traffic simulation models that leverage big data in the future.

2.3 Metamodel Simulation-based Optimization (SO) Framework

This section presents the general metamodel simulation-based optimization (SO) framework, upon which the methodology of the following chapters is developed.

In this thesis, the offline demand calibration is formulated as an SO problem and addressed using a metamodel SO approach. The metamodel SO calibration algorithm is based on that of Osorio and Bierlaire (2013), which is in turn based on the derivative-free trust-region (TR) algorithm of Conn *et al.* (2009).

The broad family of SO problems considered can be formulated as follows:

$$\min_{x \in \mathcal{X}} f(x, z; \hat{q}). \tag{2.1}$$

Problem (2.1) consists of two components: a simulation-based objective function f, and a feasible region $\mathcal{X}$. The objective function f is not available in closed-form, it can only be estimated via simulation. It depends on: the calibration parameters x (e.g., behavioral parameters, supply parameters, and demand matrices), endogenous simulation variables z (e.g., link flows, travel times) and exogenous simulation parameters $\hat{q}$ (e.g., network topology). The feasible region $\mathcal{X}$ is defined by a set of constraints assumed analytical (rather than simulation-based), differentiable, and of general-form (e.g., non-convex).

For a review of metamodel SO methods, we refer the reader to Osorio (Chap. 5, 2010). The main idea underlying metamodel SO algorithms is to address the simulation-based Problem (2.1) by recursively solving a set of analytical problems. At iteration k the SO algorithm solves an analytical problem with the following form:

$$\min_{x \in \mathcal{X}} m_k(x; \beta_k). \tag{2.2}$$

The main idea is to construct, at every iteration, an analytical approximation (m_k in (2.2)) of the simulation-based objective function (f in (2.1)) using a low fidelity

analytical traffic model. The function m_k is known as the metamodel. It is often a parametric function, with the iteration-specific parameter vector, β_k, being fitted based on simulated observations.

For a given iteration k, the metamodel SO algorithm loops through two primary parts as depicted in Figure 2-2: (1) metamodel fitting and (2) subproblem optimization. In metamodel fitting (Step 1), coefficients of metamodel β_k are fitted based on simulation observations obtained up until the current iteration. For instance, to fit the parameters the algorithm of Osorio and Bierlaire (2013) solves a weighted least squares problem that minimizes the distance between the simulated objective function estimates and the metamodel predictions. A description of how the metamodel parameters are fitted is given in Appendix C.3. Then, the algorithm identifies new points to be simulated (Step 2). Figure 2-2 distinguishes between two types of points, which correspond to Steps 2a and 2b, respectively. In Step 2a, the metamodel $m_k(x)$ is used instead of the original simulation-based objective function (2.1) to perform optimization, which we refer to as subproblem optimization, and to derive a solution (known as a trial point). A second type of point to simulate are model improvement points (Step 2b), which are identified based on general sampling

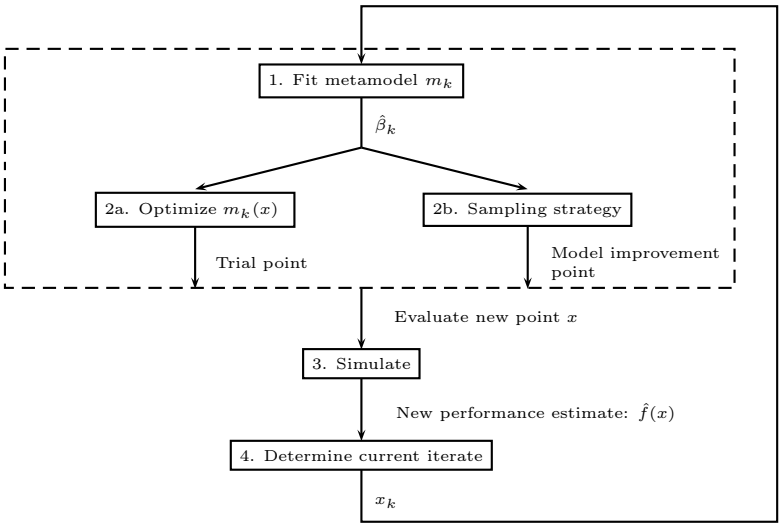


Figure 2-2: Metamodel simulation-based optimization framework. Adapted from Chong and Osorio (2018)

strategies (e.g., uniformly and randomly sampling from the feasible region $\mathcal{X}$). Step 3 simulates the newly identified point x to obtain a new performance estimate $\hat{f}(x)$ and the new simulation observation is added to the prior simulation observations to re-fit the metamodel parameters. Step 4 determines whether to update the current iterate x_k (i.e., the best point found so far). These steps are iterated: as new points are evaluated by the simulator, new simulation observations become available and the accuracy of the metamodel can be improved, leading to trial points with improved performance derived by the subproblem optimization. The aforementioned steps are solved recursively until, for instance, the computational budget (e.g., the total number of simulation runs) is depleted. The focus of this thesis is on the formulation of novel metamodels that can lead to trial points with good simulation-based performance, and can do so under tight computational budgets (i.e., with a few simulation observations).

In order to use this framework for calibration, the main challenge is to formulate a metamodel that: (i) provides a good approximation of the mapping of the decision vector (i.e., calibration parameters) to the objective function, and (ii) is also computationally efficient such that Problem (2.2), which needs to be solved at every iteration of the algorithm, can be solved efficiently. The following chapters contribute by formulating suitable metamodels that achieve these two goals for specific calibration problems.

Metamodels can be classified as either: (i) functional models, which are general-purpose functions (e.g., polynomials) suitable to approximate an arbitrary function f; (ii) physical models, which are problem-specific functions. In other words, their functional form depends on the specific problem.

We use the metamodel idea of Osorio and Bierlaire (2013) that combines functional models and physical models. At iteration k the metamodel is defined by:

$$m_k(x;\beta_k) = \beta_{k,0} f_A(x) + \phi(x;\beta_k). \tag{2.3}$$

Equation (2.3) defines m_k as a linear combination of a general-purpose parametric

function (e.g., a polynomial) denoted by ϕ and a problem-specific approximation of f (defined in (2.1)) denoted by f_A. The role of the general-purpose function ϕ is to ensure asymptotic algorithmic properties and that of the problem-specific function f_A is to enable computational efficiency. The first scalar element of the coefficient vector β_k is denoted by $\beta_{k,0}$. Based on (2.3), the metamodel can be interpreted as a problem-specific approximation (f_A) which is corrected by a scaling factor ($\beta_{k,0}$) and an additive general-purpose correction term (ϕ).

The problem-specific approximation (f_A) is derived from an analytical macroscopic network model. Hence, the analytical problem solved at every iteration is defined by:

$$\min_{x \in \mathcal{X}} m_k(x; \beta_k) \tag{2.4}$$

$$h(x, \tilde{v}; \tilde{q}) = 0, \tag{2.5}$$

where h denotes the analytical macroscopic network model, with endogenous variables x and $\tilde{v}$ (e.g., expected link queue-lengths) and exogenous parameters $\tilde{q}$ (e.g., network topology). This problem differs from Problem (2.2) in the presence of an additional set of constraints (2.5). This set of constraints represents the analytical macroscopic network model used.

The key to achieving computational efficiency is in formulating a problem-specific approximation f_A that is a good approximation of the true, unknown, simulation-based objective function f. Hence, for a given transportation problem the main challenge is in the formulation of a suitable analytical network model (function h of (2.5)) that satisfies the following requirements:

(i) It leads to a good analytical approximation (f_A) of the simulation-based objective function (f);

(ii) It is a scalable analytical network model, such that large-scale networks can be addressed;

(iii) It is computationally efficient to solve. Every iteration of the SO algorithm requires solving Problem (2.4)-(2.5), which contains the analytical network model

as a set of constraints. Hence, the analytical network model needs to be computationally inexpensive to evaluate;

(iv) It maps the decision variables (i.e., calibration input parameter vector) to available traffic measurements, at least partially.

The broad family of metamodels defined by (2.3) has already been used to efficiently address large-scale urban traffic management problems (e.g., Chen *et al.*, 2012; Osorio and Chong, 2015; Osorio and Nanduri, 2015a,b; Chong and Osorio, 2018; Osorio and Selvam, 2017; Osorio and Atastoy, 2017), behavioral and supply parameter calibration problems (Zhang *et al.*, 2017, 2016a), and a high-dimensional OD demand calibration problem (Zhang and Osorio, 2018) while using inefficient yet detailed stochastic microscopic traffic simulators. Metamodel approaches have also been used recently for addressing various transportation problems, such as in Chen *et al.* (2016), where a pricing optimization problem is addressed based on a large-scale mesoscopic network model. As in traditional simulation literature, the metamodel of Chen *et al.* (2016) is a general-purpose (also known as a functional) metamodel. This comes with the advantage of being a general-purpose methodology, which can be directly applied to a problem regardless of its formulation (e.g., the choice of the objective function, the decision variables, the underlying network structure). Nonetheless, general-purpose metamodels lack problem-specific structural information, and hence are not designed to be computationally efficient.

Chapter 3

Travel Behavioral Model Calibration of Stochastic Traffic Simulators

The contents of this chapter were published as: Zhang, C., Osorio, C., and Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B: Methodological*, 97, 214-239.

3.1 Introduction

This chapter focuses on the demand calibration of simulation-based road transportation models and illustrates the use of the proposed algorithm with the calibration of a single demand parameter (i.e., a route choice behavioral parameter). This chapter formulates the calibration problem as a simulation-based optimization (SO) problem and uses a metamodel SO algorithm. The chapter formulates a novel metamodel suitable for demand calibration problems. The analytical metamodel combines information from the simulator and information from an analytical differentiable and tractable network model that relates the calibration parameters to the simulation-based objective function. The algorithm is first validated on a toy network and then used to address a calibration problem for a large-scale Berlin metropolitan network.

The rest of this chapter is organized as follows. Section 3.2 presents the proposed methodology, which is developed based on the metamodel SO framework in Sec-

tion 2.3, followed by case studies on both a toy network and the Berlin metropolitan network (Section 3.3). We conclude with a brief discussion (Section 3.4). Appendix A details the SO algorithm. Appendix B.1 contains additional details of the numerical results of Section 3.3.

3.2 Methodology

The general framework discussed in this chapter is suitable to address a broad class of calibration problems (e.g., demand, supply, time-independent, time-dependent, etc.). Section 3.2.1 formulates a general calibration problem. In order to provide a more detailed formulation of the proposed framework, Section 3.2.2 focuses on one specific calibration problem which is a time-independent one-dimensional demand calibration problem. Section 3.2.2 formulates the calibration problem as a metamodel SO problem. It then formulates a suitable metamodel to address the calibration problem.

3.2.1 Problem Formulation

We consider the calibration of travel demand parameters from link flows. Origin-destination (OD) pairs are trip production and attraction points connected by a set of routes in an urban network. Let τ and i index the simulation time intervals of duration T and the network links, respectively, and let $n = 1 \dots N$ index the individual trip-makers (i.e., simulated travelers or agents) in the system. Denoting by $\Delta_{ni\tau} \in \{0, 1\}$ the stochastic binary indicator of traveler n crossing link i in time interval τ, the stochastic simulated link flow rate $F_{i,\tau}$ on link i in time interval τ is

$$F_{i,\tau} = \frac{1}{T} \sum_{n=1}^{N} \Delta_{ni\tau} \tag{3.1}$$

with the expectation

$$E[F_{i,\tau}] = \frac{1}{T} \sum_{n=1}^{N} \delta_{ni\tau}, \tag{3.2}$$

where $\delta_{ni\tau} = E[\Delta_{ni\tau}]$ is the probability that traveler n crosses link i in time interval τ (Flötteröd and Bierlaire, 2009). This probability is in turn dependent on the network conditions z, in particular travel times of the various routes, which in turn depend on the travel behavior (e.g., departure time, route choice), requiring typically to solve (3.2) iteratively. This is subsequently expressed by writing $\delta_{ni\tau} = \delta_{ni\tau}(\theta; z)$ with θ a parameter vector of the underlying behavioral model. Here, θ represents a general calibration parameter vector. It can include, for instance, coefficients of attributes of behavioral models, such as the travel time coefficient of a route choice model or of a departure time choice model. In a transportation simulator that simulates the movement of individual travelers (microscopic and often mesoscopic), these iterations can be interpreted as a learning process over subsequent days, where in each day some travelers update their travel decisions (typically route choice, in some models also time and mode choice) based on the most recent network conditions z, followed by a simulation of the corresponding vehicle flows through the network, which in turn yields updated network conditions.

Let $y_{i,\tau}$ be the number of vehicles counted in the field on link i in time interval τ. A traditional nonlinear least squares formulation of the calibration problem is then to minimize the following objective function:

$$f(\theta) = \min_{\theta \in \Theta} \sum_{i \in \mathcal{I}} \sum_{\tau} \left(y_{i,\tau} - E[F_{i,\tau}(\theta; z)] \right)^2 . \tag{3.3}$$

The summation considers all time intervals τ and all links i that belong to the set of links with measurements available, denoted by $\mathcal{I}$. The feasible region Θ is defined analytically, and often consists of simple bound constraints.

Formulation (3.3) illustrates the main challenges of the calibration problem described in Section 2.1. The function $E[F_{i,\tau}(\theta; z)]$ has no closed-form expression avail-

able, since z (e.g., the travel times) and $E[F_{i,\tau}(\theta; z)]$ (i.e., the expected link flows) can only be estimated by evaluating the stochastic simulation model. Thus, Problem (3.3) cannot be solved using traditional analytical and deterministic optimization approaches. The function $E[F_{i,\tau}(\theta; z)]$ is a nonlinear function that describes intricate spatial-temporal traffic phenomena in the stochastic traffic simulator and lacks sound mathematical properties such as convexity.

The purpose of this chapter is to propose an efficient calibration algorithm for such difficult problems. The main idea is to embed within the algorithm analytical structural problem-specific information. In particular, we propose to formulate and provide the algorithm with an analytical approximation of the relationship between θ and $E[F_{i,\tau}(\theta; z)]$. We expect this analytical information to enable computational efficiency.

Developing such an analytical approximation is a challenging problem because the approximated mapping involves the highly nonlinear and stochastic *network loading map* of path flows on network conditions, comprising in the simulation context all difficulties that come along with real traffic flow dynamics in urban networks (including, e.g., multi-lane flows, spillbacks, flow interactions in intricate intersections).

3.2.2 Metamodel Formulation

The novel methodology proposed in this chapter is valid for a general class of calibration problems. In order to illustrate and implement a specific instance of it, we focus, hereafter, on: (i) a time-independent calibration problem (i.e., we consider a single time interval), (ii) the calibration of a single behavioral parameter, a scalar θ, which governs route choice. Recall, that Section 2.3 presented a general calibration problem with a general calibration vector denoted by x. This section considers a specific calibration problem. Hence, θ now represents a scalar that denotes the travel time coefficient of a route choice model.

The specific SO problem is then formulated as:

$$f(\theta) = \min_{\theta} \sum_{i \in \mathcal{I}} \left(y_i - E[F_i(\theta; z)]\right)^2 \tag{3.4}$$

$$\theta_L \leq \theta \leq \theta_U, \tag{3.5}$$

where θ_L (resp. θ_U) denotes a lower (resp. upper) bound. The purpose of the metamodel is to approximate the simulation-based performance metric $E[F_i(\theta; z)]$, which denotes the expected flow on link i. Let $m_{i,k}$ denote the analytical approximation of $E[F_i(\theta; z)]$ at iteration k of the SO algorithm. Given an expression for $m_{i,k}$, the SO algorithm would then solve a series of analytical problems of the form:

$$\min_{\theta} \sum_{i \in \mathcal{I}} \left(y_i - m_{i,k}(\theta; \beta_{i,k})\right)^2 \tag{3.6}$$

$$\theta_L \leq \theta \leq \theta_U \tag{3.7}$$

$$h(\theta, \tilde{v}; \tilde{q}) = 0, \tag{3.8}$$

where h denotes the analytical network model used to analytically approximate the expected link demands.

The metamodel is formulated for link i and iteration k as:

$$m_{i,k}(\theta; \beta_{i,k}) = \beta_{i,k,0}\lambda_i(\theta) + \beta_{i,k,1} + \beta_{i,k,2}\theta, \tag{3.9}$$

where θ is the (scalar) behavioral parameter, $\lambda_i(\theta)$ is the expected demand for link i approximated by the analytical network model, and $\beta_{i,k}$ is a three-dimensional vector of metamodel parameters: $\beta_{i,k} = [\beta_{i,k,0}, \beta_{i,k,1}, \beta_{i,k,2}]$. The metamodel can be interpreted as the analytical approximation ($\lambda_i(\theta)$) corrected by a scaling factor ($\beta_{i,k,0}$) and an additive linear-in-θ error term (represented by $\beta_{i,k,1} + \beta_{i,k,2}\theta$).

We now present the analytical network model that will yield the analytical approximation of expected link demands ($\lambda_i(\theta)$). The model is a probabilistic and differentiable network model. We map the road network as a probabilistic queueing network. Each link is modeled as a single queue. Hereafter, the terms "link" and

"queue" are used interchangeably. We first introduce notation, we then present the model formulation and comment on its derivation.

d_s expected travel demand for OD pair s;

ℓ_i space capacity of queue i;

l_i length of link i (road length);

v_i maximum speed on link i;

μ_i service rate for queue i;

γ_i expected external demand for queue i;

λ_i expected demand for queue i;

$\tilde{t}_i$ expected travel time for queue i;

$\tilde{n}_i$ expected number of vehicles in queue i;

p_{ij} turning probability from queue i to queue j;

t_r expected travel time on route r;

f_r expected demand on route r;

$\tilde{p}_{sr}$ probability that a traveler on OD pair s takes route r;

θ travel time coefficient of the route choice model;

$\mathcal{S}$ set of OD pairs;

$\mathcal{Q}$ set of queues;

$\mathcal{R}$ set of routes;

$\mathcal{R}_s$ set of routes of OD pair s;

$\mathcal{G}_{ij}$ set of routes that consecutively go through queues i and j;

$\mathcal{H}_i$ set of routes that go through queue i;

$\mathcal{T}_i$ set of routes that start with queue i;

Ψ_r set of links on route r.

$$
\begin{cases}
f_r = \sum_{s \in \mathcal{S}} d_s \tilde{p}_{sr} & \forall r \in \mathcal{R} & \text{(3.10a)} \\
\tilde{p}_{sr} = \dfrac{e^{\theta t_r}}{\sum_{j \in \mathcal{R}_s} e^{\theta t_j}} & \forall s \in \mathcal{S}, \forall r \in \mathcal{R}_s & \text{(3.10b)} \\
t_r = \sum_{i \in \Psi_r} \tilde{t}_i & \forall r \in \mathcal{R} & \text{(3.10c)} \\
\tilde{t}_i = \dfrac{l_i}{v_i} + \dfrac{\tilde{n}_i}{\lambda_i} & \forall i \in \mathcal{Q} & \text{(3.10d)} \\
\tilde{n}_i = \dfrac{\tilde{\rho}_i}{1-\tilde{\rho}_i} - \dfrac{(\ell_i+1)\tilde{\rho}_i^{\ell_i+1}}{1-\tilde{\rho}_i^{\ell_i+1}} & \forall i \in \mathcal{Q} & \text{(3.10e)} \\
\tilde{\rho}_i = \dfrac{\lambda_i}{\mu_i} & \forall i \in \mathcal{Q} & \text{(3.10f)} \\
\lambda_i = \gamma_i + \sum_{j \in \mathcal{Q}} p_{ji} \lambda_j & \forall i \in \mathcal{Q} & \text{(3.10g)} \\
p_{ij} = \dfrac{\sum_{r \in \mathcal{G}_{ij}} f_r}{\sum_{w \in \mathcal{H}_i} f_w} & \forall i \in \mathcal{Q}, \forall j \in \mathcal{Q} & \text{(3.10h)} \\
\gamma_i = \sum_{r \in \mathcal{T}_i} f_r & \forall i \in \mathcal{Q} & \text{(3.10i)}
\end{cases}
$$

Equation (3.10a) describes how expected OD demand is mapped to expected route demand through the route choice model. In other words, it defines the expected route demand as a weighted sum of expected OD demands. Equation (3.10b) is the route choice model. It is a multinomial logit (MNL) model with a single attribute: expected route travel time. This is a simplification of the route choice model used in the simulation models in the application in Section 3.3. For a detailed description of the route choice models of the simulator, see Zhang *et al.* (2016b). Equation (3.10c) defines the expected route travel time as the sum of expected link travel times.

Equation (3.10d) approximates the expected link travel time as the sum of expected free-flow travel time and expected delay. The analytical approximation model used here is based on stochastic point queues, meaning that it does not capture spillbacks and that all link outflow constraints result from the link's downstream bottleneck capacity. Recall that the structural metamodel is, by design, a simplified

approximation of the simulator, which may itself use space queues and capture spill-backs. The expected time it takes a vehicle to traverse a spatial link with a point queue downstream bottleneck is the sum of: (i) the time to join the point queue, which corresponds to term $\frac{l_i}{v_i}$ and represents a constant free-flow travel time, and (ii) the expected delay or time needed to pass the bottleneck, which corresponds to term $\frac{\tilde{n}_i}{\lambda_i}$ and is obtained by modeling the downstream bottleneck capacity as a distributed quantity in order to capture variability in link discharge flows.

The expected delay is based on Little's law (Little, 2011, 1961), assuming an infinite space capacity queue. The expected free-flow travel time is defined as the travel time to travel the link at maximum speed. Equation (3.10e) approximates the expected length of a given queue. It is obtained by considering the queue as an isolated $\mathrm{M/M/1/}\ell$ queue, with finite space capacity ℓ. A derivation of this expression can be found in Appendix A of Osorio and Chong (2015). Equation (3.10f) defines $\tilde{\rho}_i$, which is known as the traffic intensity of the queue, as the ratio of expected demand to expected supply. Equation (3.10g) is a flow conservation equation that relates the expected demand for queue i (denoted by λ_i) to the sum of expected external demand to queue i (denoted by γ_i) and expected demand arising from upstream queues. The expected demand, λ_i, is also referred to as the arrival rate of the queue. The expected external demand, γ_i, represents demand that arises from outside the network, i.e., trips that start at queue i. The turning probability from queue i to queue j (denoted by p_{ij}) is defined by Equation (3.10h) as the ratio of the expected demand from i to j and the expected demand for queue i. The expected external demand for queue i is defined in Equation (3.10i) as the sum of the expected demand over all routes that start at queue i.

For the simulation model used in the application, the set of route alternatives for a given OD pair is endogenous (i.e., it varies with θ and across assignment iterations). However, for the purpose of tractability a fixed (i.e., exogenous) route choice set is considered for the analytical model. For every OD pair, a set of 10 routes is constructed. Details on the derivation of this choice set are given in Section 2.3 of Zhang *et al.* (2016b). Since the analytical model does not capture congestion-

dependent route choice changes, these are captured by the polynomial component (ϕ of Equation (2.3)) of the metamodel. The use of an endogenous, iteration-dependent, route choice set could yield more accurate analytical results, but at a higher computational burden.

In summary, the analytical approximation of $\lambda_i(\theta)$ of Equation (3.9) is obtained by evaluating the analytical network model defined by the system of nonlinear differentiable Equations (3.10). The exogenous parameters of this system of equations are: $\theta, d_s, \mu_i, \gamma_i, \ell_i, l_i, v_i, \mathcal{S}, \mathcal{Q}, \mathcal{R}, \mathcal{R}_s, \mathcal{G}_{ij}, \mathcal{H}_i, \mathcal{T}_i$, and Ψ_r. All other variables are endogenous and are obtained when solving the above system of equations.

As is illustrated with the case studies of Section 3.3, the proposed model (system of Equations (3.10)) works well for scenarios with various levels of congestion, including congested scenarios. This is remarkable given that the model does not account for the occurrence of spillbacks and their impact on the performance of upstream links. Nonetheless, the model accounts for the impact of the link's finite space capacity on the expected link delay. More specifically, the expected link travel time equation (Equation (3.10d)) consists of the sum of expected link free-flow travel time and expected link delay. The delay term is based on the approximation of the expected number of vehicles on the link (Equation (3.10e)) which assumes that each link has a finite space capacity (denoted ℓ_i). Hence, the impact of finite space capacity on the expected delay is accounted for.

Equation (3.8) represents the system of Equations (3.10). The function h is of class C^∞. There exists a variety of algorithms to efficiently solve this differentiable system of nonlinear equations. The dimension of the system of equations scales linearly with the number of links in the network and linearly with the number of OD pairs. This makes it a scalable model suitable for the calibration of large-scale networks.

This metamodel framework is not constrained to the use of the concrete analytical queueing-theoretic network model (3.10) but is compatible with a variety of other analytical network models. We consider this flexibility a strength of the proposed framework. The proposed analytical model is particularly efficient, since its evaluation consists of solving a system of nonlinear equations, the dimensionality of

which scales linearly with the number of links in the network and linearly with the number of OD pairs. This makes it a scalable model suitable for the calibration of large-scale networks. The proposed framework can be used with other analytical network models. Ongoing work studies the use of traffic-theoretic network loading models for simulation-based optimization, such as the model of Osorio and Flötteröd (2014), which is a stochastic model consistent with Newell's deterministic simplified theory of kinematic waves (Newell, 1993). The use of approximate expressions of local, and path marginal cost functions (e.g., Ghali and Smith, 1995; Shen *et al.*, 2007; Qian and Zhang, 2011; Lu *et al.*, 2013) could be of interest. The main challenge in using more traditional traffic-theoretic models is to develop formulations that both: (i) have endogenous user-equilibrium assignment, and (ii) are computationally efficient for large-scale networks.

The modeling of route choice sets is in general an unresolved problem; this is because route choice sets are not directly observable (e.g., Frejinger *et al.*, 2009; Flötteröd and Bierlaire, 2013). Hence we consider it adequate to deploy the simplest possible approach to route choice set generation (where the choice set is exogenous) and to rely on the general-purpose polynomial term in the metamodel to absorb the resulting modeling error.

We briefly describe how the exogenous route choice set is derived. A more detailed description is provided in Section 2.3 of Zhang *et al.* (2016b). The exogenous set consists of 10 route alternatives per OD pair. We consider different behavioral parameter values. For a given value, we run a set of sequential assignment iterations, and extract the set of routes used by the simulator in the last assignment iteration. We group the set of routes extracted from the various behavioral parameter values. Then for a given OD pair, the final set of 10 routes is determined by selecting the set (of 10 routes) with maximal distance-based overlap with the entire set of extracted routes. For a given network, this process is carried out once, prior to calibration. The route choice set is then kept fixed throughout the entire calibration process.

The use of an exogenous choice set contributes to the computational efficiency of the proposed approach. As will be discussed in Section 3.3.3, for the Berlin metropoli-

tan network, the analytical model with exogenous route choice set yields an accurate approximation of the form of the simulation-based objective function. This highlights the negligible effect that the exogenous route choice set has on the analytical model's accuracy. Nonetheless, this observation is network- and problem-specific. A discussion on extensions of this framework to allow for the use of endogenous route choice sets is given in Section 3.4.

3.3 Case Studies

3.3.1 Experimental Design

We apply the proposed approach in two case studies: a hypothetical toy network and a Berlin metropolitan network. The simulator used is MATSim (Horni *et al.*, 2016). The main purpose of both case studies is to evaluate the added value of embedding within the calibration algorithm the problem-specific analytical structural information, which is provided by the analytical network model. For each case study, we compare the performance of two calibration approaches that only differ in the use (or not) of the analytical network model. All other algorithmic details are identical. The first approach is the proposed approach (denoted by algorithm Am), which uses the metamodel defined by (3.9). The second approach (denoted by algorithm $A\phi$) considers a metamodel defined for iteration k and link i as:

$$\phi_{i,k}(\theta; \beta_{i,k}) = \beta_{i,k,1} + \beta_{i,k,2}\theta. \tag{3.11}$$

This metamodel differs from that of (3.9) in the absence of the macroscopic analytical network model. In other words, compared to Am, $A\phi$ uses the same general-purpose metamodel component but has no problem-specific metamodel component.

The network topology characteristics of both networks are summarized in Table 3.1. The main details of the experimental design for each network are displayed in Table 3.2. The first row of Table 3.2 considers the lower and upper bounds for θ, this defines the feasible region, Θ. For the toy network, we consider a set of hypothet-

	Toy network	Berlin network
Number of links	6	24335
Number of nodes	6	11345
Number of OD pairs	1	3635
Expected demand (veh/hr)	1400	172900

Table 3.1: Network attributes

	Toy network	Berlin network
Bounds for θ values (1/hr), $[\theta_L, \theta_U]$	$[-60, 0]$	$[-60, 0]$
True θ values (1/hr), θ^*	$\{-5, -20, -55\}$	N.A.
Initial θ values (1/hr), θ_0	$\{0, -40, -60\}$	$\{0, -40, -60\}$
Computational budget	30	20
Simulation replications	5	10
Simulation assignment iterations	50	100
Total simulation assignment iteration per algorithmic run	7500	20000

Table 3.2: Experimental design

ical θ values, based on which we simulate synthetic traffic counts. The second row of the table displays these hypothetical values, which we refer to as the true values and denote by θ^*. For the Berlin network, θ^* is unknown. Recall that θ is the travel time coefficient of the route choice model. Hence its unit is the inverse of the unit of travel time. In other words, the concrete value of θ depends on the time unit used for travel time. For both networks, the travel times are computed in hours, hence the unit of θ is hr^{-1}.

For a given θ^*, we initialize the SO algorithms with three different initial points, denoted by θ_0. The initial values used are displayed in row 3. Therefore, in total there are 9 different experiments for the toy network and 3 different experiments for the Berlin network. For each experiment, we run each SO algorithm (Am or $A\phi$) 3 times. The need to run an algorithm multiple times for a given experiment is due to the stochastic nature of the traffic simulator. Each algorithmic run consists of a maximum number of points (θ values) to be evaluated. This is known as the computational budget or the sampling budget (displayed in row 4). Once this computational budget is reached, the algorithm is terminated.

For a given θ value, an estimate of the simulation-based objective function (Equa-

tion (3.4)) is obtained by averaging over a set of independent simulation replications. The number of independent simulation replications is displayed in row 5. Each simulation replication consists of a sequential set of simulation assignment iterations (displayed in row 6). For a given simulation replication, the estimate of $E[F_i(\theta; z)]$ is obtained by averaging the observations from the last 5 assignment iterations. Details on how the assignment iterations are initialized are given in Zhang *et al.* (2016b). The last row of the table indicates the total number of simulation assignment iterations per SO algorithmic run. For example, for the toy network, the computation of a single experiment (i.e., one run of the algorithm) requires a total of 7500 simulation assignment iterations: the performance of a total of 30 θ values are estimated, each estimation involves 5 independent replications, each of which requires 50 sequential assignment iterations. This leads to a total of 30*5*50=7500 simulation assignment iterations. Similarly, for the Berlin network each SO run involves 20*10*100=20000 simulation assignment iterations. In other words, we allow for a tight computational budget, which is defined as a small number of iterations of the calibration algorithm. This number is 30 (resp. 20) for the toy (resp. Berlin) network. Each iteration of the calibration algorithm involves an estimation of the simulation-based optimization objective function (f of Equation (3.4)). For each estimation, we carry out a set of sequential assignment iterations. Hence, each estimation involves calling the simulator 5*50=250 (resp. 10*100=1000) times.

Table 3.3 displays the runtime statistics considering the 9 (resp. 3) experiments for the toy (resp. Berlin) network. For the toy network, a single assignment iteration takes an average of 0.06 minutes, leading to an average of 3 minutes per SO iteration and 90 minutes per SO algorithmic run (i.e., one experiment). For the Berlin network an assignment iteration averages 1.2 minutes, an SO iteration 120 minutes, and an algorithmic run 2400 minutes (i.e., 40 hours). The toy network experiments are carried out on a standard laptop with a 4-core Intel i7-3740QM processor and 8GB RAM. The Berlin network experiments are carried out on a server with a 40-core Intel Xeon E5-2660 processor and 64GB RAM.

	Toy network	Berlin network
Average runtime per assignment iteration	0.06	1.2
Average runtime per SO iteration	3	120
Average runtime per experiment	90	2400

Table 3.3: Simulation runtime statistics in minutes

3.3.2 Toy Network

Network Attributes

For the hypothetical toy network, we pick a set of true θ values, θ^* (row 2 of Table 3.2). We use these values to generate the "real" traffic counts via simulation. The topology of the network is displayed in Figure 3-1. Each link consists of a single lane road. Table C.1 of Appendix C details the link properties.

The network has one OD pair (node 1 to node 6) and an expected demand of 1400 vehicles per hour. There are two alternative routes connecting the OD pair, a route to the north which goes through node 3 and a route to the south which goes through node 4. The northern route has a signal controlled intersection at node 3, whereas the southern route is un-signalized. The traffic signal control at node 3 is green for 75 seconds out of the 100 seconds cycle time. The free-flow travel time on the route to the north (resp. south) is approximately 14.9 (resp. 15.5) minutes. The free-flow travel time on the route to the north is shorter and hence it is preferred when there is no congestion. As congestion increases, the route to the south becomes increasingly attractive.

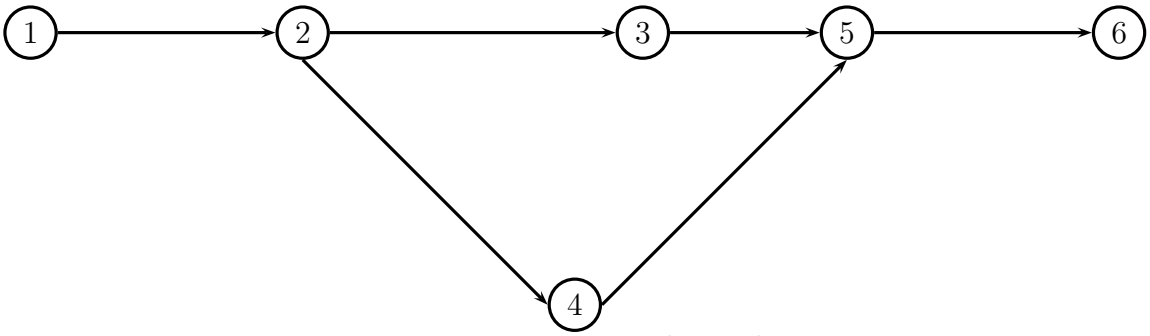


Figure 3-1: Toy network topology

Results

We first consider the experiment with $\theta^* = -5\text{hr}^{-1}$. Figure 3-2 displays the simulation-based objective function and the objective function approximation provided by the analytical traffic model (Equations (3.10)). The simulation-based objective function is defined by Equation (3.4). The simulation-based estimates are displayed as blue circles. For a given point, the estimate is based on 5 simulation replications. The red error bars displayed are 95% confidence intervals. The black curve is the analytically approximated objective function, which is defined as:

$$\sum_{i \in \mathcal{I}} \left(y_i - \lambda_i(\theta)\right)^2 . \tag{3.12}$$

In other words, the analytical approximation is obtained by replacing the simulation-based metric (expected link flow) with the analytical metric. The black curve appears an excellent approximation of the simulation-based objective function.

Figure 3-2 also displays a green range for θ. This range is a set of θ values that have statistically equivalent objective function values to that of θ^*. If an SO method yields θ values within this range, we consider it to have converged. Statistical equivalence is tested with a paired t-test where the null hypothesis assumes equal expectations, while the alternative hypothesis assumes unequal expectations. For this experiment, the equivalent region is $[-6, -4]$ (in units hr^{-1}).

Figures 3-3 to 3-5 each considers different initial points, θ_0. The x-axis displays the iteration of the SO algorithm, and the y-axis displays the current iterate (i.e., the best θ value found so far by the algorithm). The solid black lines correspond to the proposed method, Am, while the dashed red lines correspond to the benchmark method, $A\phi$. Recall that the only difference in the methods is their metamodel formulation, all other algorithmic details are identical. These figures also display the aforementioned equivalent region (in green).

Note that in Figure 3-5, 2 of the red curves overlap from iteration 1 to iteration 14, this occurs for the right-most red curve. For all 3 figures (3-3 to 3-5), the following observations hold.

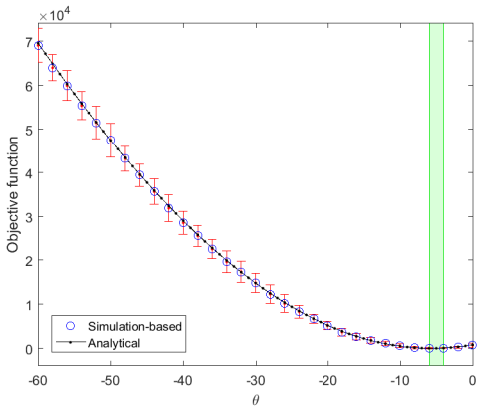


Figure 3-2: Simulation-based and analytical objective functions for $\theta^* = -5\text{hr}^{-1}$

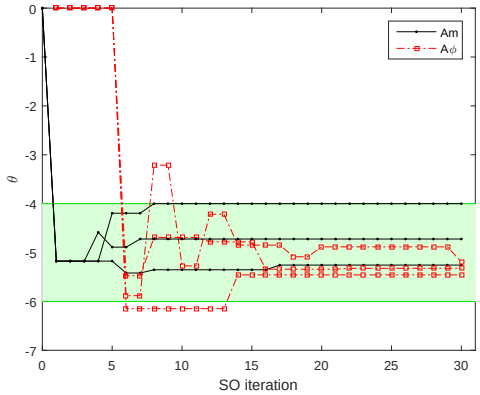


Figure 3-3: Algorithmic solutions versus iterations, for $\theta^* = -5\text{hr}^{-1}$ and $\theta_0 = 0\text{hr}^{-1}$

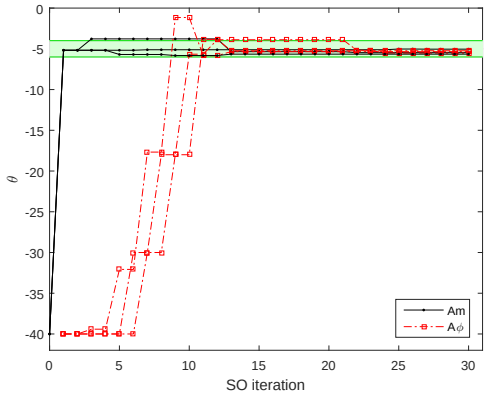


Figure 3-4: Algorithmic solutions versus iterations, for $\theta^* = -5\text{hr}^{-1}$ and $\theta_0 = -40\text{hr}^{-1}$

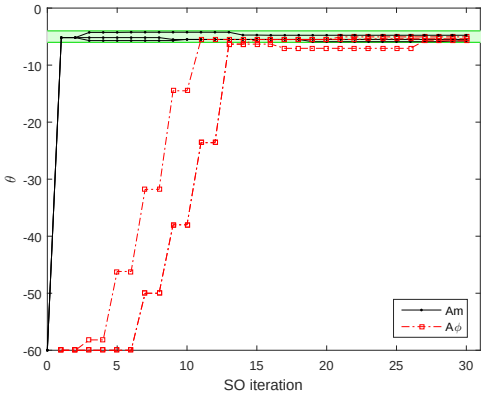


Figure 3-5: Algorithmic solutions versus iterations, for $\theta^* = -5\text{hr}^{-1}$ and $\theta_0 = -60\text{hr}^{-1}$

- All Am and all $A\phi$ runs converge.
- All Am runs converge faster than the $A\phi$ runs.

We now consider a true value of $\theta^* = -20\text{hr}^{-1}$. The different objective functions are displayed in Figure 3-6. Again, the analytical approximation of the objective function provided by the analytical network model (Equations (3.10)) is an excellent approximation of its simulation-based counterpart. The statistically equivalent region for θ is $[-21, -18]$ (in units hr^{-1}). The corresponding experimental results are shown in Figures 3-7 to 3-9. Note that in Figure 3-9, all 3 red curves overlap from iteration 1

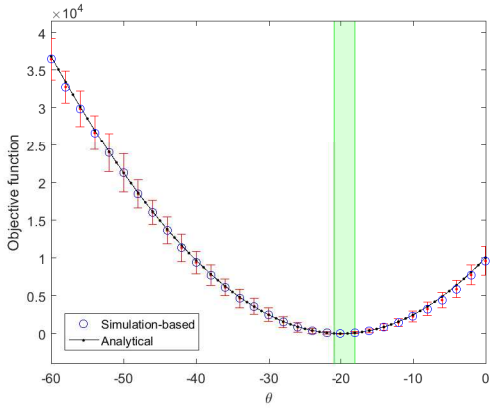


Figure 3-6: Simulation-based and analytical objective functions for $\theta^* = -20\text{hr}^{-1}$

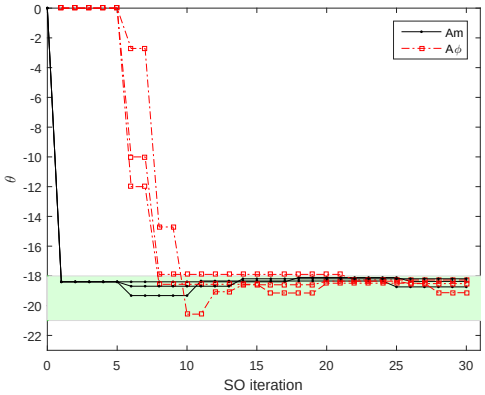


Figure 3-7: Algorithmic solutions versus iterations, for $\theta^* = -20\text{hr}^{-1}$ and $\theta_0 = 0\text{hr}^{-1}$

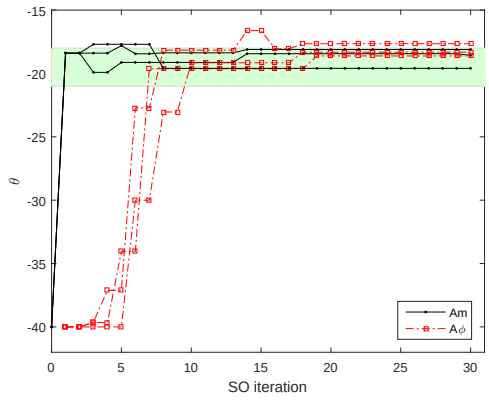


Figure 3-8: Algorithmic solutions versus iterations, for $\theta^* = -20\text{hr}^{-1}$ and $\theta_0 = -40\text{hr}^{-1}$

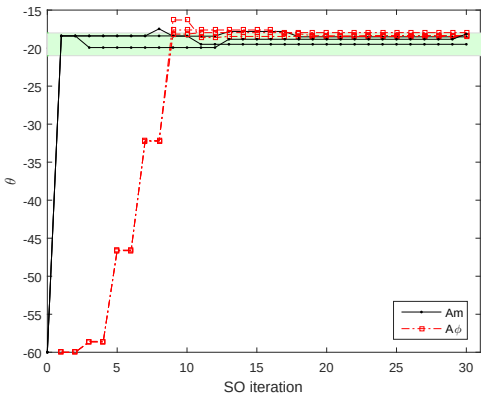


Figure 3-9: Algorithmic solutions versus iterations, for $\theta^* = -20\text{hr}^{-1}$ and $\theta_0 = -60\text{hr}^{-1}$

to iteration 8. For Figures 3-7 to 3-9, all Am runs converge, while 7 out of 9 $A\phi$ runs converge. Of the 2 $A\phi$ runs that do not converge, both had current iterates within the green region, yet exited the region. For the converged runs, convergence tends to be faster for Am than for $A\phi$.

The different objective functions for the experiments with $\theta^* = -55\text{hr}^{-1}$ are displayed in Figure 3-10. As before, the analytical objective function approximation is almost identical to the simulation-based objective function. The statistically equivalent region for θ is $[-57, -54]$ (in units hr^{-1}). The results for these experiments are displayed in Figures 3-11 to 3-13.

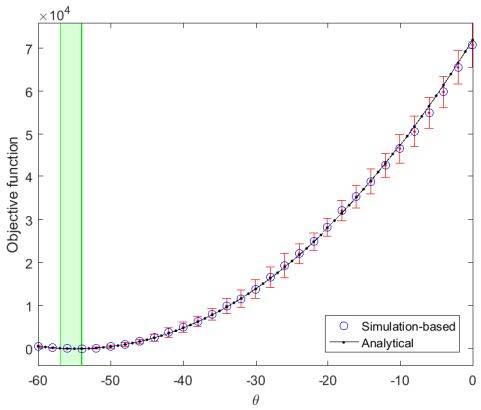


Figure 3-10: Simulation-based and analytical objective functions for $\theta^* = -55\text{hr}^{-1}$

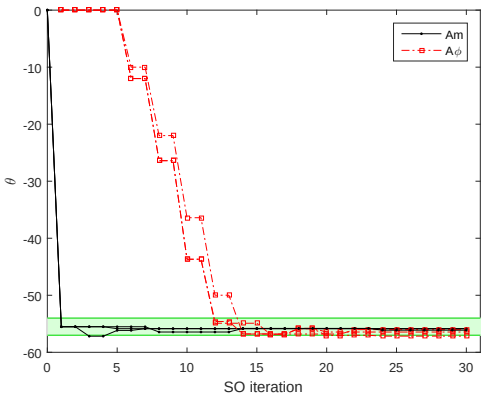


Figure 3-11: Algorithmic solutions versus iterations, for $\theta^* = -55\text{hr}^{-1}$ and $\theta_0 = 0\text{hr}^{-1}$

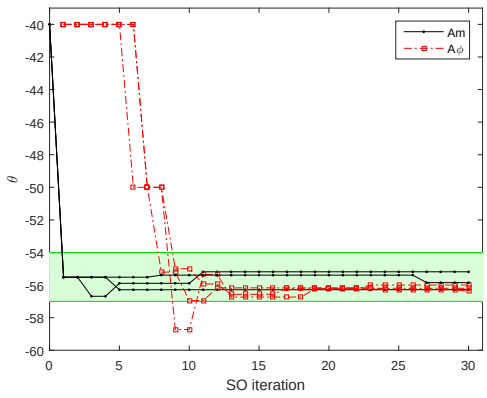


Figure 3-12: Algorithmic solutions versus iterations, for $\theta^* = -55\text{hr}^{-1}$ and $\theta_0 = -40\text{hr}^{-1}$

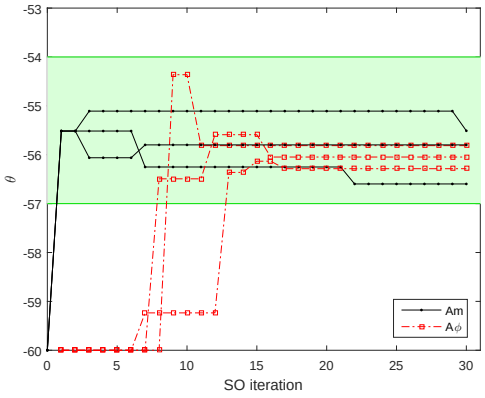


Figure 3-13: Algorithmic solutions versus iterations, for $\theta^* = -55\text{hr}^{-1}$ and $\theta_0 = -60\text{hr}^{-1}$

Note that in Figure 3-11, 2 of the red curves overlap from iteration 1 to iteration 11, this occurs for the left-most red curve. As with the previous experiments, all Am runs converge. Eight of the nine $A\phi$ runs also converge.

From the perspective of the values proposed by the algorithms for the calibration parameter, θ, both SO methods yield values with good performance and do so within a few algorithmic iterations. Overall the performance of both methods is similar and good. Overall, Am identifies good solutions faster than $A\phi$, and converges systematically.

Let us compare the performance of both methods in terms of their computational

efficiency. Table 3.4 considers all the above experiments. It displays for each experiment and each method, the number of algorithmic iterations until convergence (i.e., the first time the equivalent region is entered and not exited thereafter). If a method upon termination of the algorithm (i.e., iteration 30) has not converged, then we indicate a value equal to the computational budget (i.e., the maximum number of iterations) of 30. This underestimates the convergence statistics for the non-converged experiments.

This table indicates that for Am and for $\theta^* = -5\text{hr}^{-1}$, 8 of the 9 experiments converge after 1 iteration. For $\theta^* = -20\text{hr}^{-1}$, this happens 6 out of 9 times, and for $\theta^* = -55\text{hr}^{-1}$, this happens 8 out of 9 times. At the first algorithmic iteration, only a single objective function estimate is available (the estimate obtained at the initial value θ_0). Hence, this instantaneous convergence is due to the information provided by the analytical network model. When considering all true values (i.e., all 27 experiments), Am converges on average at iteration 2.7, and in the worst case at iteration 18. Algorithm $A\phi$ converges on average at iteration 14.4 (this average includes the non-converged cases, where we enter an iteration value of 30), at best at iteration 6, and does not converge for 3 experiments. The corresponding numerical values of the solutions derived by each method for each experiment are presented in Table B.1 of Appendix B.

Table 3.5 analyzes the convergence of the methods in terms of their simulation

		Am			$A\phi$		
$\theta^* = -5$	$\theta_0 = 0$	1	1	1	10	6	14
	$\theta_0 = -40$	1	13	1	13	22	11
	$\theta_0 = -60$	1	1	1	11	27	15
$\theta^* = -20$	$\theta_0 = 0$	1	1	1	22	8	10
	$\theta_0 = -40$	8	6	1	7	30	10
	$\theta_0 = -60$	1	18	1	17	11	30
$\theta^* = -55$	$\theta_0 = 0$	1	1	5	30	12	14
	$\theta_0 = -40$	1	1	1	11	9	8
	$\theta_0 = -60$	1	1	1	9	13	8

Table 3.4: Number of algorithmic iterations until convergence for the toy network

		Am			$A\phi$		
$\theta^* = -5$	$\theta_0 = 0$	3	3	3	30	18	42
	$\theta_0 = -40$	3	39	3	39	66	33
	$\theta_0 = -60$	3	3	3	33	81	45
$\theta^* = -20$	$\theta_0 = 0$	3	3	3	66	24	30
	$\theta_0 = -40$	24	18	3	21	90	30
	$\theta_0 = -60$	3	54	3	51	33	90
$\theta^* = -55$	$\theta_0 = 0$	3	3	15	90	36	42
	$\theta_0 = -40$	3	3	3	33	27	24
	$\theta_0 = -60$	3	3	3	27	39	24

Table 3.5: Simulation runtimes until convergence for the toy network (minutes)

runtimes in minutes until convergence. As in the above analysis, if a method has not converged, we indicate the total simulation runtime used until the algorithm was terminated. This underestimates the runtime needed until convergence. The method Am converges on average within 8 minutes of simulation, while $A\phi$ converges on average in 43 minutes. By providing the algorithm with the analytical information, we can converge with an average 81.4% reduction in simulation runtime. This table highlights the computational efficiencies that are achieved when using Am.

3.3.3 Berlin Metropolitan Network

Network Attributes

Figure 3-14 displays the metropolitan Berlin network topology, as modeled in the simulator. As a reference, we also display here the road map of the corresponding network (Figure 3-15). This network represents the metropolitan area of Berlin, Germany. The area includes the city of Berlin and the broader federal state of Brandenburg. It consists of 24,335 uni-directional links, 11,345 nodes, and 3,635 OD pairs. Real (i.e., field) traffic counts from sensors on 346 links are available per hour for 24 hours collected by the Traffic Management Center (in German: *Verkehrsmanagementzentrale*). We focus on the morning peak hour: 8-9 AM, during which the expected demand is 172,900 vehicles. The model considers automobile traffic only. For more data and model details, see Ziemke *et al.* (page 120, Section "Counts", 2015)

Figure 3-14: Metropolitan Berlin simulation network

Figure 3-15: Metropolitan Berlin road network. Map source: http://goo.gl/2kLXzj, down-loaded on 06/06/2017

and Ziemke (pages 57-59, 2013).

Results

Figure 3-16 displays the simulation-based objective function estimates with corresponding 95% confidence intervals obtained from 20 simulation replications. As for the toy network, the green region represents the range of statistically equivalent θ values, which is $[-6, -1]$ (in units hr^{-1}). Based on the finite and small set of simulated points, the θ value at which the minimum objective function is obtained is near -2hr^{-1}. The analytical approximation of the objective function derived from the analytical network model (Equations (3.10)) is displayed in Figure 3-17. Both functions have similar form and seem to be non-convex. Note that these figures have different y-axis limits. Hence, the analytical function closely approximates the form of the simulation-based function, but is not scaled properly. The scaling is corrected by the metamodel (term $\beta_{i,k,0}$ of Equation (3.9)). The minimum of the analytical objective function is obtained at -5hr^{-1}, which is in the green statistically equivalent region.

We proceed as for the toy network: we run experiments for 3 different initial values θ_0. For each initial value, we plot the current iterate (i.e., best θ value identified so

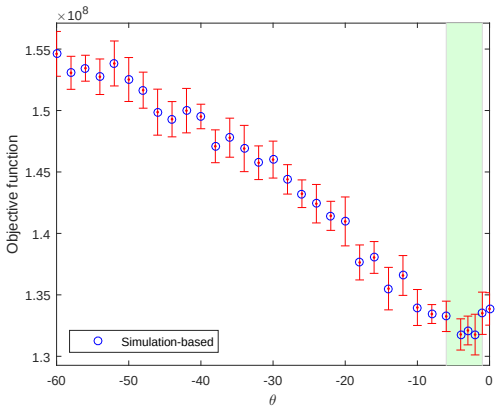


Figure 3-16: Simulation-based objective function for the Berlin network

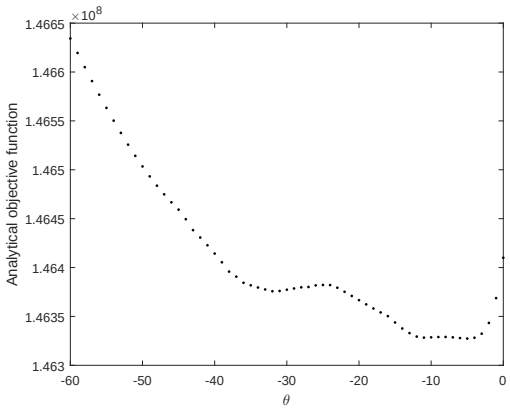


Figure 3-17: Analytical objective function for the Berlin network

far by the algorithm) versus algorithmic iteration. We do this for both SO methods: Am and $A\phi$. The results are presented in Figures 3-18, 3-19 and 3-20.

In these figures, some curves overlap: in Figure 3-18, two dashed red lines overlap at $\theta = 0\text{hr}^{-1}$ for all iterations; in Figure 3-19, two solid black lines overlap at $\theta = -4.7\text{hr}^{-1}$.

In Figure 3-18, all Am runs converge, 1 of 3 $A\phi$ runs converges, the remaining 2 stay at the initial θ value of 0hr^{-1}. In Figure 3-19, all Am runs converge, 1 of the 3 $A\phi$ runs converges, the remaining 2 reach and stay at the lower bound value for θ equal to 0hr^{-1}. In Figure 3-20, all Am runs converge, none of the $A\phi$ runs converge, 2 reach the lower bound and the third stays at the value of -0.4hr^{-1}.

For all 3 figures, the following observations hold.

- All Am runs converge. Seven of the nine runs converge at the first iteration. This is thanks to the analytical network model closely approximating the simulation-based objective function.
- Only 2 of the 9 $A\phi$ runs converge. Of the non-converged runs, two stay at the initial value of $\theta = 0\text{hr}^{-1}$ (Figure 3-18), 4 leave the initial value, reach and stay at the upper bound value of $\theta = 0\text{hr}^{-1}$ (Figures 3-19 and 3-20).
- When comparing the converged runs: all Am runs converge faster than the $A\phi$ runs.

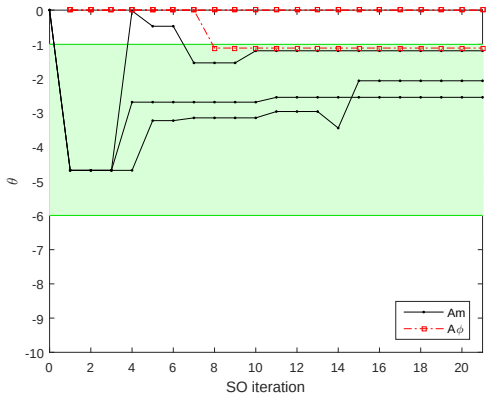


Figure 3-18: Algorithmic solutions versus iterations, for $\theta_0 = 0\text{hr}^{-1}$

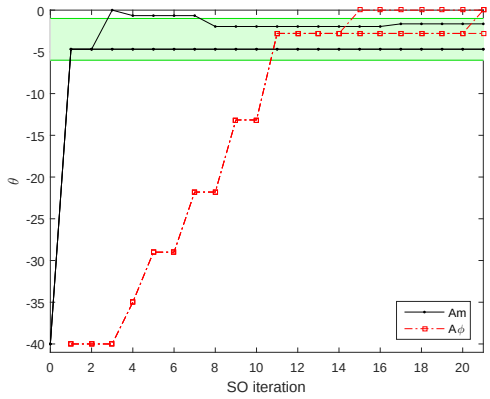


Figure 3-19: Algorithmic solutions versus iterations, for $\theta_0 = -40\text{hr}^{-1}$

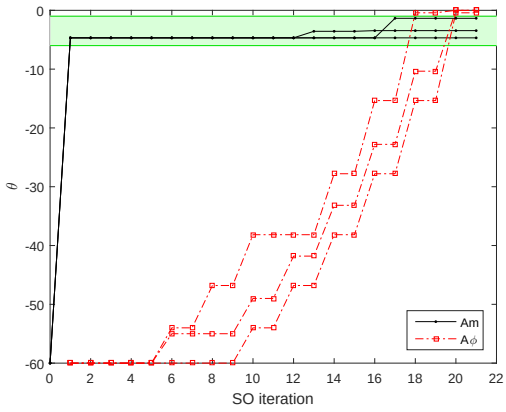


Figure 3-20: Algorithmic solutions versus iterations, for $\theta_0 = -60\text{hr}^{-1}$

We analyze the convergence statistics of the two methods. We proceed as for the toy network. Table 3.6 considers all the above Berlin experiments. It displays for each experiment and each method, the number of iterations until convergence (i.e., the first time the equivalent region is entered and not exited thereafter). If a method has not converged before termination of the algorithm (i.e., iteration 20), we indicate a value equal to the computational budget, which is 20. This underestimates the convergence statistics for the non-converged experiments. This table indicates that method Am converges at iteration 1 for 7 of the 9 experiments. On average it converges at iteration 2.4 and at worst at iteration 8. Method $A\phi$ converges only 2 of the 9 runs. For those two converged experiments, convergence is achieved at iterations 8 and 11. The corresponding numerical values of the solutions derived by

	Am			$A\phi$		
$\theta_0 = 0$	1	7	1	20	20	8
$\theta_0 = -40$	1	1	8	20	20	11
$\theta_0 = -60$	1	1	1	20	20	20

Table 3.6: Number of iterations until convergence for the Berlin network

each method for each experiment are presented in Table B.2 of Appendix B.

The corresponding statistics in terms of simulation runtimes are displayed in Table 3.7. If a given method has not converged upon termination of the algorithm, we indicate the total simulation runtime upon termination (i.e., 2400 minutes). As before, this underestimates convergence runtime for method $A\phi$ which often does not converge. On average method Am converges after 293 minutes (i.e., 4.9 hours), while $A\phi$ does so within 2120 minutes (i.e., 35.3 hours). Method Am achieves an average 86.2% reduction in runtime until convergence, which corresponds in this case study to average savings of 30.4 hours of simulation per experiment.

We now compare the performance of the best solutions proposed by each method with the performance of the value currently used in the Berlin model, the latter was obtained from prior calibration efforts (Ziemke *et al.*, 2015). Let $\tilde{\theta}$ denote this pre-calibrated value -6hr^{-1}.

Recall, that for a given initial value, θ_0, we ran each method 3 times, this leads to 3 solutions. We now compare the performance of these solutions with that of $\tilde{\theta}$. In order to evaluate the performance of a given θ value, we ran 50 simulation replications and obtained 50 observations of the objective function. More specifically, for a given θ value, we ran 100 assignment iterations to obtain 1 objective function estimate. We repeated this process 50 times to obtain 50 independent simulation observations (or

	Am			$A\phi$		
$\theta_0 = 0$	120	840	120	2400	2400	960
$\theta_0 = -40$	120	120	960	2400	2400	1320
$\theta_0 = -60$	120	120	120	2400	2400	2400

Table 3.7: Simulation runtimes until convergence for the Berlin network (minutes)

estimates).

Figure 3-21 displays the cumulative distribution function (cdf) of these 50 observations. Each curve corresponds to a given θ value. The solid black (resp. dashed red) curves correspond to values derived by Am (resp. $A\phi$). The dotted black curve corresponds to the pre-calibrated value $\tilde{\theta}$. The more a curve is shifted to the left, the lower the objective function estimates, i.e., the better its performance. More specifically, for a given x-value, the corresponding y-value on the cdf curve gives the proportion of observations (out of the 50) that have objective function values smaller than x. Figure 3-21 presents the solutions obtained with an initial value of $\theta_0 = 0\text{hr}^{-1}$. 2 of the 3 runs of $A\phi$ yield the same solution (i.e., the algorithm considers for all iterations the initial value, $\theta_0 = 0\text{hr}^{-1}$, as the best value). Thus, only 2 dashed red curves are visible. The right-most dashed red curve represents the 2 identical solutions (i.e., $\theta = 0\text{hr}^{-1}$). All 3 values derived by Am outperform the 3 values obtained by $A\phi$. They also outperform the pre-calibrated value.

Figure 3-22 displays the cdf of each solution obtained when initializing the algorithms with $\theta_0 = -40\text{hr}^{-1}$. Here as well, 2 of the 3 runs of $A\phi$ yield the same solution of 0hr^{-1}. Hence, only 2 dashed red curves are visible, one of which represents 2 identical solutions. The proposed method Am also yields 2 solutions that are similar (both are approximately -4.7hr^{-1}). The same conclusions as above hold: all solutions derived by Am outperform all solutions derived by $A\phi$ and outperform the pre-calibrated value. Figure 3-23 displays the cdf of each solution obtained when initializing the algorithms with $\theta_0 = -60\text{hr}^{-1}$. The method $A\phi$ yields a solution of 0hr^{-1} for 2 of the 3 runs. Hence, only 2 distinct dashed red curves are visible. For this figure, the same conclusions as above hold.

For each initial value θ_0 and each method, we choose only one proposed solution which is defined as that with the smallest objective function average (the average over the 50 simulation replications). In Table 3.8, we give the numerical values of the best proposed solutions for the three sets of experiments. Figure 3-24 presents the cdf of the best solutions for all three initial values. It also displays the cdf of the pre-calibrated value and of the value obtained by solving the problem with only the

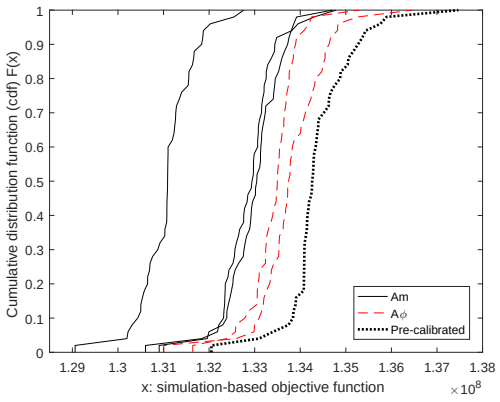


Figure 3-21: Objective function distributions for each solution of each method, when initialized with $\theta_0 = 0\text{hr}^{-1}$

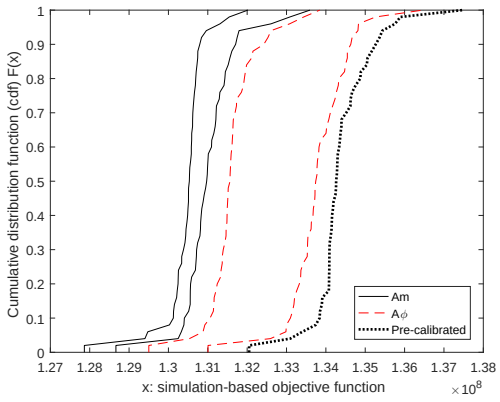


Figure 3-22: Objective function distributions for each solution of each method, when initialized with $\theta_0 = -40\text{hr}^{-1}$

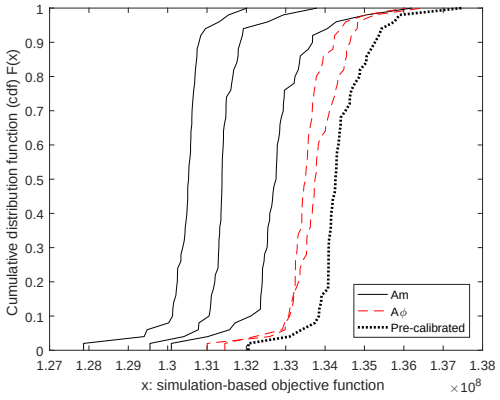


Figure 3-23: Objective function distributions for each solution of each method, when initialized with $\theta_0 = -60\text{hr}^{-1}$

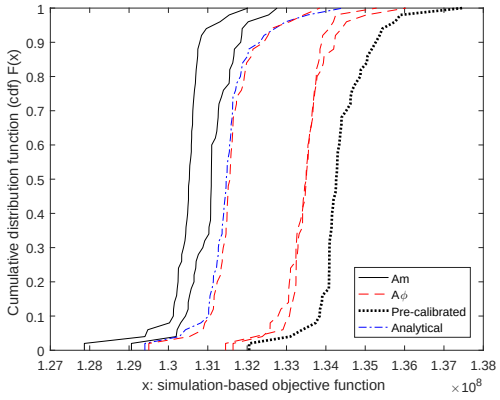


Figure 3-24: Objective function distributions for the best solutions of each method and each initial θ value

analytical network model (i.e., no simulation). The solid black curves correspond to *Am* solutions. Only two curves appear, because *Am* yields the same solution ($\theta = -4.7\text{hr}^{-1}$) for two initial values ($\theta_0 = -40\text{hr}^{-1}$ or $\theta_0 = -60\text{hr}^{-1}$). Hence, the left-most solid black curve represents two solutions. The dashed red curves correspond to the three solutions of $A\phi$. The dash-dotted blue curve corresponds to the solution of the analytical traffic model (i.e., no simulation) and the dotted black curve corresponds to the pre-calibrated value.

Figure 3-24 indicates that the solutions proposed by *Am*, for each of the three initial values, outperform all 3 solutions proposed by $A\phi$. It also outperforms the

	Am	$A\phi$
$\theta_0 = 0$	-2.1	-1.1
$\theta_0 = -40$	-4.7	-2.8
$\theta_0 = -60$	-4.7	-0.4

Table 3.8: Best solution of each method and each initial θ value

pre-calibrated value. This figure also shows that the analytical solution outperforms all $A\phi$ solutions. This indicates the added value of the analytical network model information. All Am solutions outperform the analytical solution. This indicates the value of combining analytical and simulation-based information, rather than merely solving the problem with an analytical-only approach.

Recall that the traffic field data consists of traffic counts from sensors on 346 links. Figure 3-25 compares the performance of the best Am solution ($\theta = -4.7\text{hr}^{-1}$, the left-most curve in Figure 3-24) and the pre-calibrated solution. For each solution and each sensor location, we compute the relative error of the count: $(y_i - \hat{E}[F_i(\theta; z)])/y_i$, where y_i represents the field count at location i and $\hat{E}[F_i(\theta; z)]$ represents the simulated estimate of the count at that location. Figure 3-25 displays two curves, one for each solution (Am or pre-calibrated). The curve is the cumulative distribution function of the relative error. The distribution is over all 346 links. Let us illustrate how to interpret these curves. The solid black curve corresponds to method Am. A vertical line through $x = 0$ intersects the Am curve at $y = 0.17$. This means that under the Am solution, 17% of the sensor locations yield a negative relative difference, i.e., they overestimate 17% of the counts. The pre-calibrated curve intersects this vertical line at $y = 0.07$, meaning that 7% of the locations overestimate the counts. Similarly, a horizontal line through $y = 0.6$ intersects the Am curve at $x = 0.35$. This indicates that 60% of the relative errors are below 0.35. This horizontal line intersects the pre-calibrated curve at $x = 0.44$, which indicates that 60% of the relative errors are below $x = 0.44$. From this figure we can deduce that 31% of the counts have a relative error within $[-0.2, 0.2]$ under the Am value, while this is the case of only 21% of the counts under the pre-calibrated value.

We now carry out a more detailed analysis of the performance of the different

points within the green equivalent region. Figure 3-26 considers the equivalent region $[-6, -1]$ (in units hr^{-1}). It displays a more detailed simulation-based estimate of the objective function. The function is now sampled with a step size of 0.25 (i.e., a smaller step size than in Figure 3-16). Each estimate is obtained as an average over 30 simulation replications. Its corresponding 95% confidence interval is displayed. Within this interval $[-6, -1]$ (in units hr^{-1}), we compute a new green equivalent region. We identify the point with the smallest objective function estimate. This is $\hat{\theta} = -4.75\text{hr}^{-1}$. We conduct for each point, a paired t-test to test whether its performance (i.e., objective function value) is equivalent to that of $\hat{\theta}$. If two adjacent points have statistically equivalent performance to that of $\hat{\theta}$, then both points belong to the new equivalent region. The resulting green region is now defined (in units hr^{-1}) as: $[-5, -4.5]$, $[-3.75, -2.75]$, and $[-2.25, -2]$. This new region is displayed in green in Figure 3-26.

Figure 3-26 indicates along the x-axis the location of: the 9 solutions proposed by Am (all 9 solutions fall in the initial green equivalent region of $[-6, -1]$ (in units hr^{-1}); they are represented by black crosses), 2 solutions proposed by $A\phi$ (2 solutions fall in $[-6, -1]$ (in units hr^{-1}); they are represented by red squares), the pre-calibrated value $\tilde{\theta}$ (represented by a black triangle), and the solution obtained using only the analytical traffic model (i.e., no simulation-based optimization; it is represented by a

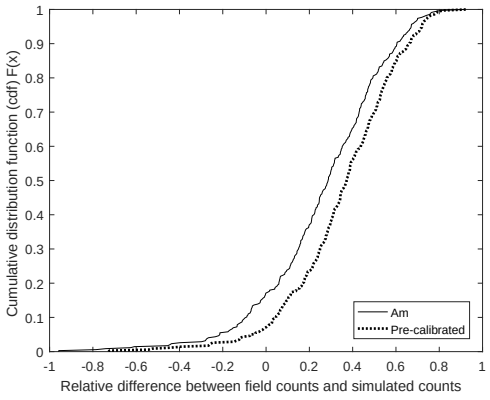


Figure 3-25: Empirical cdf of the relative difference of traffic counts

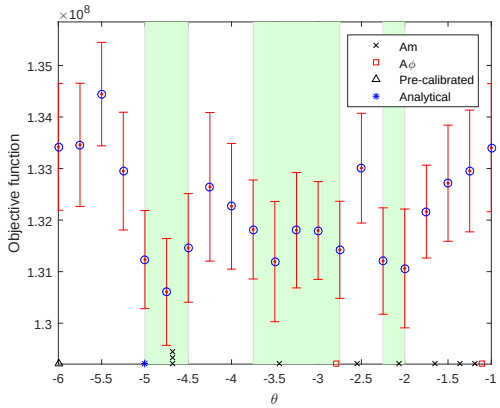


Figure 3-26: More detailed estimation of the simulation-based objective function in the region [-6,-1] (in units hr^{-1})

blue asterisk). Note that 3 of the Am solutions overlap at the value -4.7hr^{-1}, hence they are displayed as stacked upon each other on the x-axis.

Considering this more accurate estimation of the equivalent region, we observe that 5 of the 9 solutions of Am fall within this region, while only 1 solution of $A\phi$ is in the region. The solution using only the analytical traffic model also falls in this region.

The scenarios of Section 3.3 all consider high levels of congestion. For instance, in the toy network the expected demand is 1400 veh/hr, and the expected supply of links 1 and 6 is 1200 veh/hr, giving a ratio of expected demand to expected supply of 1.17. The analytical model describes the within link build-up of congestion but does not capture the occurrence and impact of vehicular spillbacks. Nonetheless, as shown in Figures 3-2, 3-6 and 3-10, it provides a good approximation of the simulation-based objective function. The Berlin network consists of a set of links with varying levels of congestion: ranging from uncongested to highly congested. For instance, the city center contains approximately 7% of links with a ratio of expected demand to expected supply that is greater than 1. For the entire metropolitan network, 2% of the links have a ratio greater than 1.

3.4 Conclusions and Discussion

In this chapter, we propose a computationally efficient calibration algorithm. Efficiency is achieved by providing analytical problem-specific information to the algorithm. We formulate a metamodel that embeds information from an analytical differentiable and tractable network model. The analytical network model provides an analytical description of how the calibration parameter is related to the objective function. The performance of the method is evaluated with case studies for both a hypothetical toy network and for the Berlin metropolitan area network. The performance of the proposed method is compared to that of an algorithm that differs only in that the information from the analytical network model is not provided to it. For both networks, the proposed approach significantly improves the computational

efficiency of the calibration algorithm with an average reduction in the simulation runtime until convergence of more than 80%. The simulator used in both case studies of this chapter, MATSim, is computationally efficient compared to other high-resolution simulators. Inefficient simulators are typically calibrated under tight computational budgets. By accelerating the convergence of traditional black-box calibration algorithms, the proposed approach is of particular interest for the calibration of inefficient simulators.

The analytical route choice model (Equation (3.10b)) is a multinomial logit (MNL) model. This calibration algorithm can be used for a simulator that embeds a different route choice model specification by replacing Equation (3.10b) with an analytical and differentiable approximation of the simulator's route choice model. If such an approximation is not available or is not computationally efficient, then one can continue to use the MNL formulation of this chapter. In this case, the polynomial error term of the metamodel will capture the effects of this specification difference between the analytical and the simulation-based models.

In this chapter, the simulation-based optimization algorithm used is derivative-free, i.e., it does not require the evaluation of derivatives of the simulation-based objective function. Derivative-based algorithms have been proposed for calibration problems (e.g., Yang, 1995; Antoniou *et al.*, 2011; Balakrishna, 2006). Such algorithms have been traditionally designed to achieve asymptotic properties. There is extensive ongoing and recent work that formulates their extensions for use under tight computational budgets (Lu *et al.*, 2015; Tympakianaki *et al.*, 2015). Another area of ongoing research aims to achieve efficiency through the combined use of multiple models with varying efficiency-accuracy trade-offs (e.g., Corthout *et al.*, 2014; Osorio and Selvam, 2017).

The analytical model of the proposed approach is based on the use of an exogenous route choice set. This set is computed once prior to calibration. This contributes to the computational efficiency of the calibration algorithm. For cases where the use of an endogenous route choice set is desirable, one efficient approach would be to use at each iteration of the calibration algorithm, the route choice set used by the

simulator for the current iterate (i.e., the calibration vector value considered as best so far). This would require, at every iteration of the calibration algorithm, to compute a smaller set of routes to be used by the analytical model (this can be done as described in Section 3.2.2), and then to compute the exogenous parameters of the analytical model that depend on the route choice set. This can be done efficiently. This would allow the route choice set used to solve Problem (3.6)-(3.8) to vary across iterations of the calibration algorithm. However, for a given iteration, Problem (3.6)-(3.8) would still be solved assuming a fixed route choice set.

Chapter 4

Supply Calibration of Stochastic Traffic Simulators

The contents of this chapter are available as: Zhang, C., Osorio, C., and Flötteröd, G. (2016). Capacity calibration for large-scale traffic simulators: an efficient and scalable approach. *Technical report, Massachusetts Institute of Technology.*

4.1 Introduction

The focus of this chapter is on the calibration of supply parameters. Traditional gradient-based nonlinear optimization algorithms (e.g., trust-region) have been extensively used for the calibration of speed-density or flow-density functions (e.g., Leclercq, 2005; Hou *et al.*, 2013). The high computational cost required for the estimation of the gradients of the simulation-based objective function has led to the use of two types of algorithms, namely derivative-free algorithms (e.g., GA) and derivative-based algorithms combined with efficient derivative estimation techniques (e.g., SPSA), as discussed in Section 2.2. There is a lack of, and a need for, efficient calibration algorithms for supply models.

Compared to the demand calibration problem addressed in Chapter 3, the analytical calibration metamodel proposed in this chapter is simpler. This is because it assumes traffic assignment to be exogenous, while the model of Chapter 3 has endoge-

nous traffic assignment. Hence, the proposed model scales linearly with the network size, and independently of the dimension of the route choice set. In Section 4.3.3, we consider a Berlin metropolitan area network model, combine the supply calibration results of this chapter with the demand calibration results of Chapter 3, and show that together they lead to significant improvements in the objective function.

The remainder of this chapter is organized as follows. Section 4.2 discusses the proposed methodology. Case studies of both a synthetic toy network and the Berlin metropolitan network are presented in Section 4.3. Section 4.4 concludes this chapter. Appendix A details the simulation-based optimization (SO) calibration algorithm.

4.2 Methodology

We first formulate the supply calibration problem mathematically (Section 4.2.1). In Section 4.2.2, we formulate a metamodel for the supply calibration problem and use the general metamodel SO framework described in Section 2.3 to address the supply calibration problem. The metamodel is both computationally efficient and scalable making it suitable for the efficient supply calibration of large-scale network models.

4.2.1 Problem Formulation

The supply calibration problem is formulated the same as Problem (3.4)-(3.5), but with the behavioral parameter θ replaced by the supply parameter vector α and the lower (resp. upper) bound θ_L (resp. θ_U) replaced by α_L (resp. α_U). Similarly, y_i denotes the field measurement of a given performance metric of link i, $E[F_i(\alpha; z)]$ denotes the simulation-based expectation of this same performance metric, $\mathcal{I}$ denotes the set of links with measurements, z denotes a set of endogenous simulation variables (e.g., network states such as link speeds, travel times, queue-lengths) and exogenous simulation parameters (e.g., network topology, expected demand), and f denotes the simulation-based objective function.

The above problem is a traditional calibration problem, where the objective is to fit the calibration parameters such as to minimize the distance between observed traffic

conditions and simulated conditions. Here, we use a least-squares distance function, which is a commonly used formulation. The proposed methodology is suitable for use with a variety of link performance metrics (e.g., speeds, travel times). For the Berlin metropolitan area case study in this chapter, we have access to traffic count data. Hence, y_i represents the average flow on link i for the considered time of day, and $E[F_i(\alpha; z)]$ represents the expected (simulation-based) flow of link i.

The feasible region (Constraint (3.5)) is defined by upper and lower bounds. The proposed methodology can be directly applied to any general (e.g., non-convex) feasible region as long as it has an analytical and differentiable formulation. Simulation-based constraints (i.e., constraints that depend on the outputs of the stochastic simulator) are not accounted for in this methodology.

The above problem is time-independent. The Berlin case study focuses on calibration for the morning peak hour 8-9 AM. The extension of the proposed methodology to time-dependent supply calibration is straightforward and is part of ongoing work.

4.2.2 Metamodel Formulation

As described in Section 2.3, the approximation f_A is problem-specific. In order to formulate f_A, let us first focus on a specific supply calibration problem. The Berlin metropolitan area case study of Section 4.3 uses the multi-agent transport simulator, MATSim. We calibrate two supply parameters: α and κ. These parameters denote "flow capacity factor" and "storage capacity factor", respectively, in MATSim (page 37-38, Chapter 4.3.1, Horni and Nagel, 2016).

The parameter α (resp. κ) is a scaling factor for adjusting flow (resp. space) capacities of the links. We assume all links have the same scaling parameters (i.e., α and κ are scalars). This formulation can be readily generalized to account for one scaling parameter for each category of links, or even for each individual link. The link model of MATSim is based on a particle-discretized instance of the kinematic wave model and assumes a triangular fundamental diagram, where the flow capacity and the jam density are linearly dependent (see Equation 50.1 of page 348, Flötteröd, 2016). Thus, the flow capacity and the space capacity are also linearly dependent. This

allows us to formulate a one-dimensional calibration problem with $\alpha = \kappa$. Hereafter, the supply parameter is a single scalar denoted by α.

For each link i for which there are field measurements ($i \in \mathcal{I}$ of Equation (3.4)), we proceed as in Chapter 3 and use a link-specific metamodel formulation. For a given link i and a given iteration k of the SO algorithm, the metamodel approximates the (simulation-based) expected flow of link i. Hence, the problem solved at every iteration (i.e., the subproblem) is given similarly by Equations (3.6)-(3.8) with the variable changed from θ to α.

For each link-specific metamodel, we assume the same functional form as in Chapter 3. In other words, at iteration k of the SO algorithm and for link i, the metamodel $m_{i,k}$ is defined by:

$$m_{i,k}(\alpha; \beta_{i,k}) = \beta_{i,k,0} f_{A,i}(\alpha) + \phi(\alpha, \beta_{i,k}), \tag{4.1}$$

and the general-purpose metamodel (ϕ of Equation (2.3) and (4.1)) is a linear function of the calibration parameter α:

$$\phi(\alpha, \beta_{i,k}) = \beta_{i,k,1} + \beta_{i,k,2}\alpha, \tag{4.2}$$

where $\beta_{i,k,1}$ and $\beta_{i,k,2}$ are scalar coefficients. The focus of this chapter is on the formulation of a suitable problem-specific approximation (term $f_{A,i}$).

Let us now formulate the macroscopic analytical network model used to derive the problem-specific approximation of the expected link flow of link i, which is denoted by $f_{A,i}$, in Equation (4.1). The network model is based on the probabilistic queueing theory. We proceed as in Osorio and Chong (2015), and represent each link of the road network as a queue with finite space capacity. Hereafter, the terms "link" and "queue" are used interchangeably. We introduce the following notation. The index i refers to a given queue.

ℓ_i space capacity;

μ_i service rate (i.e., flow capacity);

γ_i external arrival rate;

λ_i effective arrival rate;

p_{ij} turning probability from queue i to queue j;

ρ_i traffic intensity;

$\tilde{p}_i$ probability of queue i being full (i.e., spillback probability);

α supply parameter;

$\mathcal{Q}$ set of queues.

$$\begin{cases} \lambda_i = \gamma_i(1-\tilde{p}_i) + \sum_{j\in\mathcal{Q}} p_{ji}\lambda_j & \forall i \in \mathcal{Q} \quad (4.3a) \\ \rho_i = \dfrac{\lambda_i}{\alpha\mu_i} & \forall i \in \mathcal{Q} \quad (4.3b) \\ \tilde{p}_i = \dfrac{(1-\rho_i)\rho_i^{\alpha\ell_i}}{1-\rho_i^{\alpha\ell_i+1}} & \forall i \in \mathcal{Q} \quad (4.3c) \end{cases}$$

Equation (4.3a) corresponds to Equation (6a) of Osorio and Chong (2015) (in that, the effective arrival rate is denoted by λ_i^{eff}). It is a flow conservation equation that defines the effective arrival rate at queue i as the sum of two terms. The first, corresponds to the arrivals that arise outside the network (i.e., trips that start at queue i). This term states that trips can enter the network through queue i at a rate of γ_i as long as queue i is not full (i.e., with probability $1-\tilde{p}_i$). The traffic intensity, ρ_i of a queue is defined as the ratio of expected demand to expected supply. Equation (4.3b) approximates it as the ratio of the effective arrival rate to the scaled service rate. This equation can be derived by taking Equation (6b) of Osorio and Chong (2015) and both scaling the service rate by the factor α and discarding the impact of downstream spillbacks on the traffic intensity of link i. Equation (4.3c) approximates the probability that a link is full. This represents the probability of spillbacks on the link. In queueing theory, this probability is known as the blocking probability. This expres-

sion is obtained by approximating queue i as an $\mathrm{M/M/1}/\alpha\ell_i$ queue, where $\alpha\ell_i$ is the scaled space capacity of the queue. The derivation of the analytical expression for the blocking probability of such a queue can be obtained in, for instance, Bocharov *et al.* (2004). This equation differs from Equation (6c) of Osorio and Chong (2015) in that it uses a scaled version of the space capacity $\alpha\ell_i$ instead of the space capacity ℓ_i. The expression for the blocking probability assumes an integral value for the space capacity. Here we do not constrain $\alpha\ell_i$ to be integral.

The network model approximates the expected flow on link i ($f_{A,i}$ of Equation (4.1)) with λ_i. For a given α and a given network with n links, the network model (Equation (4.3)) is formulated as a system of n linear equations and $2n$ nonlinear equations. The dimension of the system scales linearly with the number of links in the network and is independent of link-specific attributes, such as link length. This makes it a formulation that is suitable for large-scale networks and is computationally efficient. For a given α, the exogenous parameters of the network model are: $p_{ij}, \gamma_i, \mu_i, \ell_i, \mathcal{Q}$, and the endogenous variables are $\lambda_i, \tilde{p}_i, \rho_i$.

Let us now detail how the proposed analytical network model (Equation (4.3)) differs from that used for demand calibration in Chapter 3. The main distinction is that Chapter 3 assumes endogenous traffic assignment. In other words, the network model analytically approximates the route choice model, while the above model assumes that the turning probabilities (p_{ij}) and the external link arrival rates (γ_i) are exogenous. This significantly simplifies the model, leading to enhanced computational efficiency, and, as will be illustrated in the case studies of Section 4.3, the proposed calibration algorithm remains capable of quickly finding solutions with good performance, i.e., it remains computationally efficient. A secondary distinction between the two models, is that the model of Chapter 3 approximates the traffic intensity of a queue, ρ_i, as the ratio between the total arrival rate (which is equal to $\lambda_i/(1-\tilde{p}_i)$) and the service rate, while here we consider the ratio between the effective arrival rate (λ_i) and the service rate. The implication of this is that the proposed model may underestimate the congestion levels of the links compared to the model of Chapter 3. The case studies of this chapter consider scenarios with a variety of levels of congestion. The

proposed model is shown to perform well for all experiments. This shows the minor effect of this simplification.

Note that other network models can be used to construct the metamodel functions. In other words, the framework is not limited to the use of the above system of equations (Equation (4.3)). Nonetheless, for the SO algorithm to remain computationally efficient for large-scale networks, the subproblem needs to be solved efficiently (since it is solved at every iteration of the SO algorithm). Hence, the analytical network model needs to be highly computationally efficient.

4.3 Case Studies

This section evaluates the performance of the proposed algorithm. The experimental design is presented in Section 4.3.1. Section 4.3.2 validates the performance of the algorithm with a small synthetic case study. Section 4.3.3 then applies the algorithm in the calibration of a large-scale model of a metropolitan area (Berlin, Germany) based on the use of real traffic count data. This Berlin network was also used for demand calibration in Chapter 3, hence, Section 4.3.3 also evaluates the added value of using the proposed supply calibration algorithm along with the demand calibration algorithm of Chapter 3.

4.3.1 Experimental Design

Recall that the main premise underlying the proposed algorithm is that computationally efficient calibration algorithms can be designed by formulating and providing the algorithm with analytical problem-specific structural information. To validate this premise, we benchmark the proposed algorithm with an algorithm that differs only in the analytical information provided to the algorithm. More specifically, the same SO algorithm is used along with a different metamodel. The benchmark approach uses a general-purpose metamodel which does not include any information from an

analytical network model. For link i and iteration k, the metamodel is formulated as:

$$\phi_{i,k}(\alpha; \beta_{i,k}) = \beta_{i,k,1} + \beta_{i,k,2}\alpha. \tag{4.4}$$

Unlike the proposed metamodel (Equation (4.1)), this metamodel does not contain the problem-specific component $\beta_{i,k,0} f_{A,i}(\alpha)$. In this benchmark method, all other algorithmic details are identical to that of the proposed method. Hereafter, we use Am to denote the proposed algorithm and $A\phi$ to denote the benchmark, or general-purpose, algorithm.

The toy network is a hypothetical network adapted from Figure 3-1 in Chapter 3. The main link properties for this chapter are given in Table C.2 of Appendix C. The period considered is one hour of the morning peak period, during which the expected OD demand rate is 1,800 vehicles per hour. There are two possible routes: the north route that travels through node 3 and the south route that travels through node 4. The free-flow travel time on the north (resp. south) route is approximately 5.6 (resp. 7.9) minutes. Under free-flow conditions, the north route has smaller travel time. As congestion increases, so does the probability of choosing the south route.

The topology of the Berlin metropolitan simulation model and the road map of this area are as presented in Figures 3-14 and 3-15 of Chapter 3.

The experimental design information for both networks is the same as that displayed in Table 3.2 except for the bounds, underlying true values, and initial values, which are displayed in Table 4.1. The first row of Table 4.1 specifies the lower bound (α_L) and the upper bound (α_U) for the supply parameter α. For instance, the upper bound for both networks corresponds to scaling the supply by a factor of 10.

The toy network is a synthetic case study. The set of true supply parameter values

	Toy network	Berlin network
Bounds for α values, $[\alpha_L, \alpha_U]$	$[0.01, 10]$	$[0.1, 10]$
True α values, α^*	$\{0.1, 0.4, 1.1\}$	N.A.
Initial α values, α_0	$\{0.01, 0.3, 0.5, 1.3, 10\}$	$\{0.1, 0.3, 0.5, 1.3, 10\}$

Table 4.1: Experimental setup

is given in row 2 of Table 4.1 and denoted by α^*. The three chosen values (i.e., 1.1, 0.4 and 0.1) lead to significantly different, and increasing, levels of congestion. For a given α^*, synthetic "true" traffic counts are obtained from the simulator. For the Berlin network, field traffic count data are used and the true (or optimal) value, α^*, is unknown.

For a given algorithm (Am or $A\phi$) and a given α^*, the algorithm is initialized with 5 different initial values α_0 (row 3 of Table 4.1). For each algorithm, this leads to a total of 15 different experiments (i.e., 5 different initial values for each of the 3 true values) for the toy network and 5 different experiments for the Berlin network. For a given experiment, we run the algorithm 3 times. This is to account for the stochasticity of the simulator.

4.3.2 Synthetic Case Study Results

For the toy network, we run experiments considering three different true values (recall row 2 of Table 4.1), $\alpha^* \in \{1.1, 0.4, 0.1\}$. Let us start by analyzing the performance of the algorithms for $\alpha^* = 1.1$. Of all three values, the value of 1.1 leads to the least congested conditions, yet the levels of congestion are still relatively high. Specifically, the ratio of expected demand to expected supply on links 1 or 6 is 0.91, that on links 2 and 3 is 1.23, and that on links 4 and 5 is 0.55.

Figure 4-1 displays the estimated simulation-based objective function evaluated at points that are evenly distributed in the feasible region. Recall that the simulation-based objective function is defined as:

$$\sum_{i \in \mathcal{I}} \left(y_i - E[F_i(\alpha; z)]\right)^2 . \tag{4.5}$$

The blue circles correspond to the simulated estimates with 95% confidence intervals (in red), which are computed based on 5 replications. The green region in the figure identifies the points that have statistically equivalent performance to that of the true value α^*. The statistically equivalent region corresponds to the range $[1.050, 1.113]$. Statistical equivalence is tested with a paired t-test where the null hy-

pothesis assumes equal expectations of the objective function value, while the alternative hypothesis assumes unequal expectations. We use this statistically equivalent region as a metric to determine the convergence of an algorithm. If an algorithm yields a solution within the statistically equivalent region, we consider it to have converged. The black curve in Figure 4-1 is the approximation of the objective function provided by the analytical network model. This approximation is defined as:

$$\sum_{i \in \mathcal{I}} \left(y_i - \lambda_i(\alpha)\right)^2 . \tag{4.6}$$

In other words, the black curve replaces the simulation-based expected link flow with the expected link flow of the analytical network model. Figure 4-1 indicates that the analytical model closely approximates the simulation-based objective function when $\alpha < 1$. For $\alpha > 1$, the analytical approximation is a flat line, which is inaccurate.

Note that the simulation-based objective function is well approximated by a quadratic function. Hence, for such an experiment method $A\phi$ is expected to perform well.

Figures 4-2 to 4-6 each considers a given initial point, α_0. In each figure, the x-axis displays the total number of points simulated (i.e., the computational budget consumed so far) and the y-axis displays the current iterate (i.e., the best solution

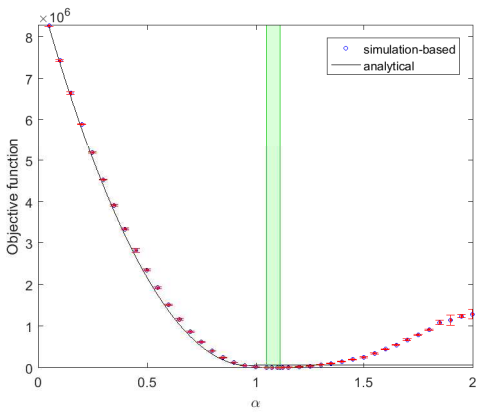


Figure 4-1: Simulation-based and analytical objective functions for $\alpha^* = 1.1$

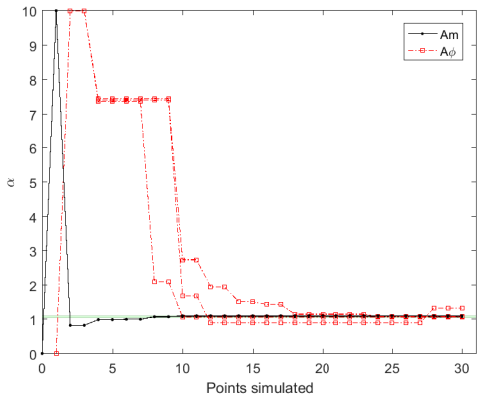


Figure 4-2: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 1.1$ and $\alpha_0 = 0.01$

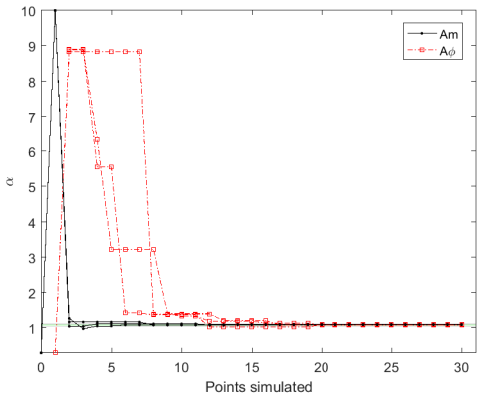


Figure 4-3: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 1.1$ and $\alpha_0 = 0.3$

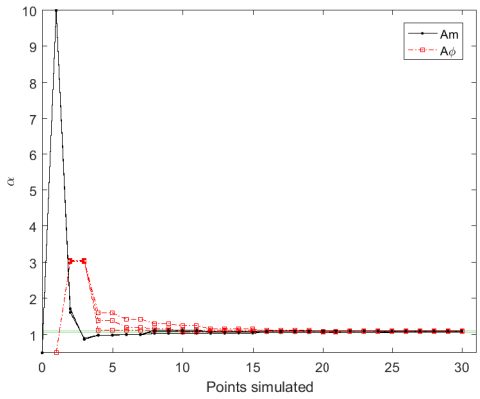


Figure 4-4: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 1.1$ and $\alpha_0 = 0.5$

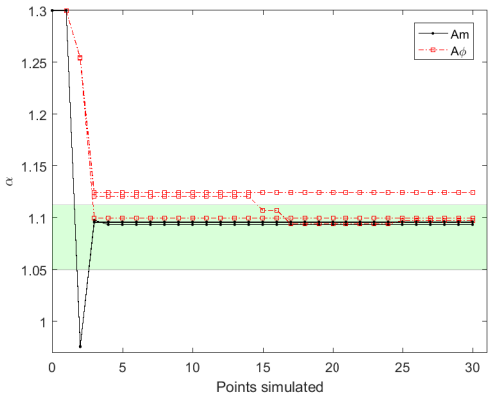


Figure 4-5: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 1.1$ and $\alpha_0 = 1.3$

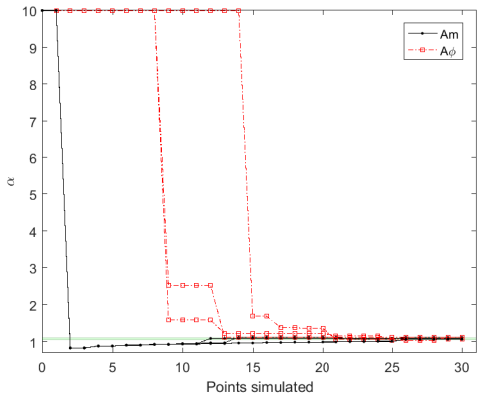


Figure 4-6: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 1.1$ and $\alpha_0 = 10$

α found so far). In each figure, there are three solid black (resp. dashed red) lines representing the 3 runs of the proposed method Am (resp. the benchmark method $A\phi$). The green horizontal region corresponds to the statistically equivalent region derived based on Figure 4-1.

Figure 4-2 considers initial point $\alpha_0 = 0.01$. The three black lines mostly overlap. All black lines (i.e., Am runs) approach the green region faster than the red lines (i.e., $A\phi$ runs). This is also the case for $\alpha_0 = 0.3$ (Figure 4-3) and $\alpha_0 = 10$ (Figure 4-6), and to a lesser extent for $\alpha_0 = 0.5$ (Figure 4-4) and $\alpha_0 = 1.3$ (Figure 4-5).

Upon depletion of the computational budget, we evaluate for each run, whether

the final solution is within the green region. If so, we consider the algorithm to have converged. For Am, 14 of the 15 runs converge. For $A\phi$, 11 of the 15 runs converge. The average number of points simulated until convergence for all converged runs is 9 for Am and 16 for $A\phi$. Hence, Am appears to converge both more often and faster than $A\phi$.

Let us now consider the scenario with true value $\alpha^* = 0.4$. This leads to higher levels of congestion than in the previous scenario. In this scenario, the ratio of expected demand to expected supply exceeds 1 for all links, i.e., all links are significantly congested. The simulation-based and analytical objective functions are displayed in Figure 4-7. As before, the simulation-based objective function could be closely approximated by a quadratic function. Hence, we expect $A\phi$ to perform well for this scenario. Just as in the previous scenario, the analytical objective function (black curve) closely approximates the simulation-based objective function for $\alpha < 1$, and provides an inaccurate approximation for $\alpha > 1$. We proceed as in the previous scenario to compute a statistically equivalent region for α^*, it is estimated as $[0.397, 0.413]$ and is displayed as the green interval in Figure 4-7.

Figures 4-8 to 4-12 display, respectively, the results for initial points, α_0, equal to 0.01, 0.3, 0.5, 1.3, and 10. For all five initial points, all runs of Am converge. They do so very fast. On the other hand, the convergence of $A\phi$ is sensitive to the initial

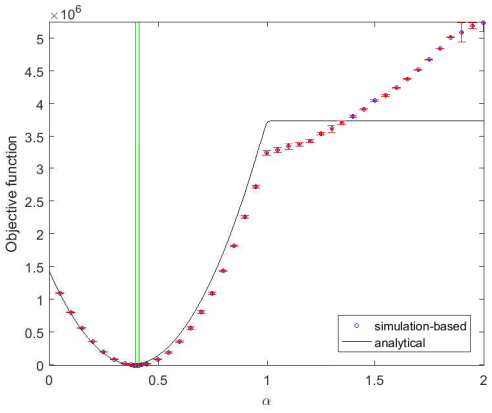


Figure 4-7: Simulation-based and analytical objective functions for $\alpha^* = 0.4$

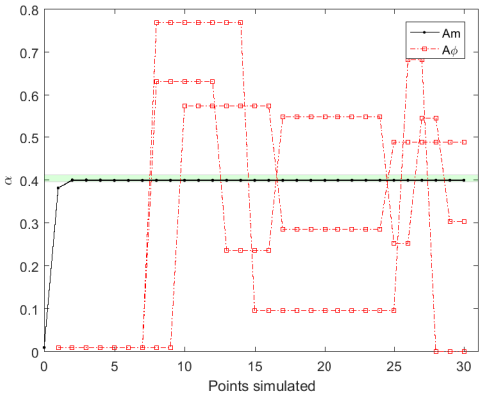


Figure 4-8: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.4$ and $\alpha_0 = 0.01$

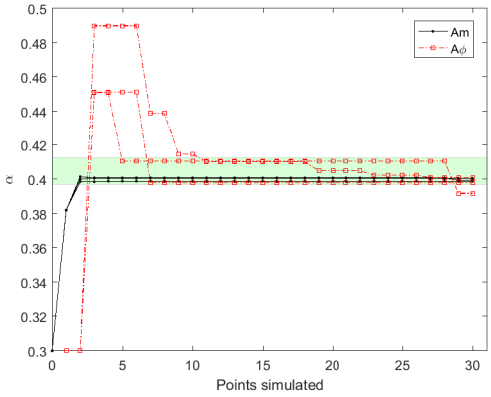


Figure 4-9: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.4$ and $\alpha_0 = 0.3$

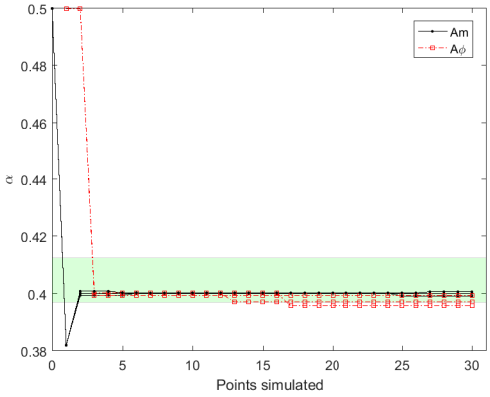


Figure 4-10: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.4$ and $\alpha_0 = 0.5$

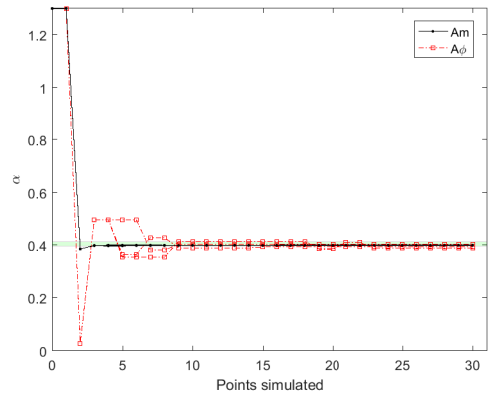


Figure 4-11: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.4$ and $\alpha_0 = 1.3$

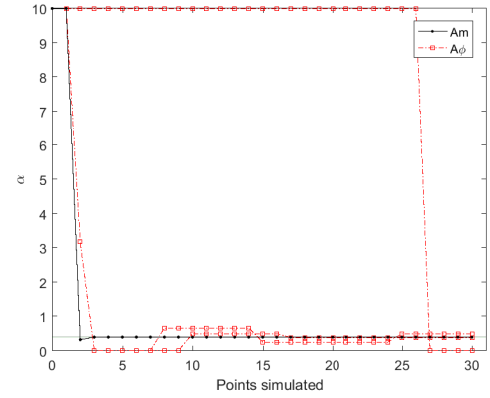


Figure 4-12: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.4$ and $\alpha_0 = 10$

point: none of the runs converge for Figures 4-8 and 4-12, while some, but not all, do for Figures 4-9, 4-10, and 4-11. In total 5 of the 15 runs of $A\phi$ converge. Of these 5 converged runs, the average number of simulated points until convergence is 7. For Am, all runs converge, and the average number of simulated points until convergence is 2.

An additional interesting difference between Am and $A\phi$ is that once a run of Am enters the green region (i.e., has converged) it no longer exits the green region, while for $A\phi$ there are cases where points within the green region are sampled yet are considered to have worse performance than points outside the green region. This

indicates that the analytical structural information provided by the analytical network model can contribute not only to converge faster and more frequently, but also to differentiate between the performance of optimal and near optimal solutions.

Let us now consider the third, and final, scenario with true value $\alpha^* = 0.1$. This scenario is the most congested one among the three. Figure 4-13 displays the simulation-based objective function, the analytical approximation of the objective functions, and the statistically equivalent region. The estimate of the statistically equivalent region is $[0.099, 0.103]$. As for the previous experiments, the analytical approximation is accurate for $\alpha < 1$, yet inaccurate for $\alpha > 1$.

Figures 4-14 to 4-18 show the results of this scenario. The trends are similar to those of the past 2 scenarios. All Am runs converge, while only 8 of 15 $A\phi$ runs converge. Among the converged runs, the average number of simulated points until convergence is 3 for Am and 5 for $A\phi$.

Let us now evaluate the computational efficiency of each method in terms of two performance metrics: (i) the number of simulated points until convergence, and (ii) the simulation runtime until convergence. The first metric is of particular interest since it is simulator-independent, i.e., it does not depend on the efficiency of the underlying simulator. The first (resp. second) metric is displayed in Table 4.2 (resp. 4.3). In Table 4.2, the runs that have not converged are indicated by the symbol •. In Ta-

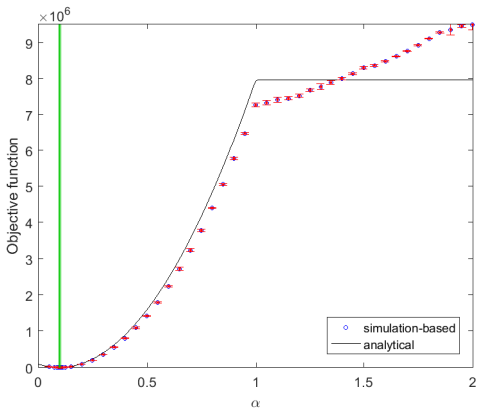


Figure 4-13: Simulation-based and analytical objective functions for $\alpha^* = 0.1$

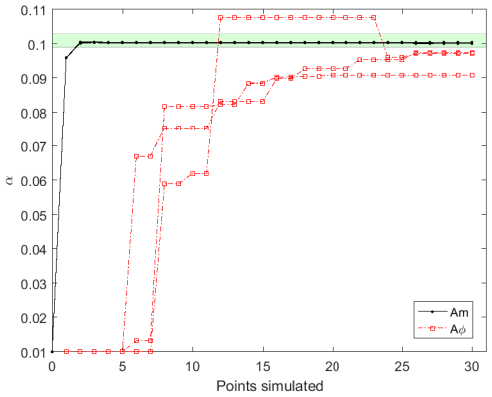


Figure 4-14: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.1$ and $\alpha_0 = 0.01$

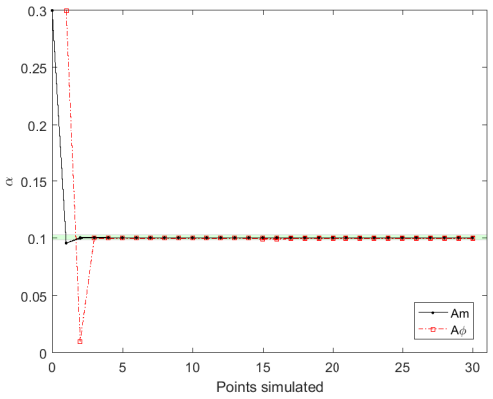


Figure 4-15: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.1$ and $\alpha_0 = 0.3$

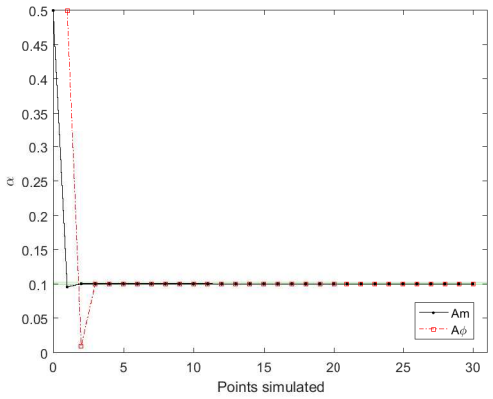


Figure 4-16: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.1$ and $\alpha_0 = 0.5$

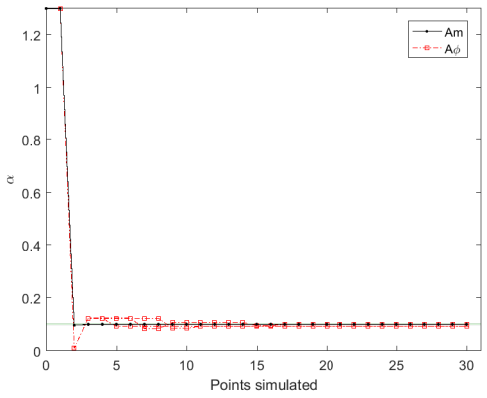


Figure 4-17: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.1$ and $\alpha_0 = 1.3$

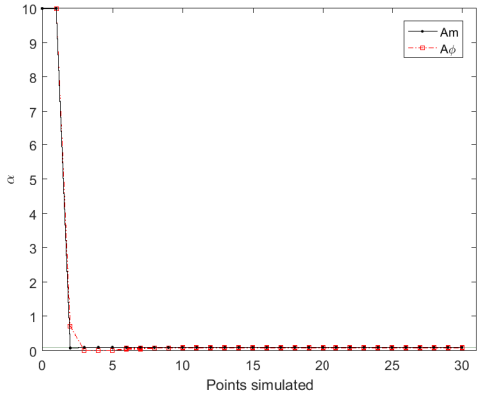


Figure 4-18: Algorithmic solutions as a function of the total number of simulated points, for $\alpha^* = 0.1$ and $\alpha_0 = 10$

ble 4.3, the runs that have not converged are underlined and the reported runtime is the runtime until the computational budget is depleted.

Note that different α values, lead to different levels of congestion, and hence different simulation runtimes. In other words, the computational runtime of a single simulation run depends on the specific α value and varies significantly across the feasible region of α.

Table 4.2 indicates that Am converges for 44 of the 45 toy network experiments, i.e., it converges for 98% of the experiments. On the other hand, $A\phi$, converges for 24 of the 45 experiments, i.e., it converges for 53% of the experiments. Among the

		Am			$A\phi$		
$\alpha^* = 0.1$	$\alpha_0 = 0.01$	2	2	2	•	•	•
	$\alpha_0 = 0.3$	2	2	2	3	3	3
	$\alpha_0 = 0.5$	2	2	2	3	3	3
	$\alpha_0 = 1.3$	3	3	3	•	17	•
	$\alpha_0 = 10$	4	4	11	•	10	•
$\alpha^* = 0.4$	$\alpha_0 = 0.01$	2	2	2	•	•	•
	$\alpha_0 = 0.3$	2	2	2	•	11	7
	$\alpha_0 = 0.5$	2	2	2	3	3	•
	$\alpha_0 = 1.3$	3	3	3	9	•	•
	$\alpha_0 = 10$	3	3	3	•	•	•
$\alpha^* = 1.1$	$\alpha_0 = 0.01$	8	8	8	•	24	10
	$\alpha_0 = 0.3$	12	8	4	20	17	20
	$\alpha_0 = 0.5$	12	8	16	20	16	10
	$\alpha_0 = 1.3$	3	3	3	•	3	15
	$\alpha_0 = 10$	•	12	26	•	•	21

Table 4.2: Number of points simulated until convergence for the toy network, the symbol • represents non-converged cases.

		Am			$A\phi$		
$\alpha^* = 0.1$	$\alpha_0 = 0.01$	63	65	64	1216	1031	1074
	$\alpha_0 = 0.3$	36	35	38	68	67	65
	$\alpha_0 = 0.5$	18	19	18	49	54	55
	$\alpha_0 = 1.3$	14	24	15	207	168	201
	$\alpha_0 = 10$	36	23	75	739	220	1263
$\alpha^* = 0.4$	$\alpha_0 = 0.01$	23	37	30	707	658	1894
	$\alpha_0 = 0.3$	18	30	12	591	262	295
	$\alpha_0 = 0.5$	23	11	15	28	58	846
	$\alpha_0 = 1.3$	14	14	29	210	475	531
	$\alpha_0 = 10$	24	22	18	1666	1427	136
$\alpha^* = 1.1$	$\alpha_0 = 0.01$	76	92	93	248	622	87
	$\alpha_0 = 0.3$	44	27	16	58	77	59
	$\alpha_0 = 0.5$	46	27	24	71	89	31
	$\alpha_0 = 1.3$	12	23	11	245	9	87
	$\alpha_0 = 10$	79	19	79	129	145	68

Table 4.3: Simulation runtimes until convergence for the toy network (minutes). For the non-converged cases, the statistic reported is underlined and indicates the runtime until the computational budget is depleted

converged runs, convergence is achieved on average after 5 points are simulated for

Am and after 11 points are simulated for $A\phi$. Considering only the converged runs, the average runtime until convergence is 33 minutes for *Am* and 119 minutes for $A\phi$. The average runtime, considering both converged and non-converged cases (i.e., considering all entries of Table 4.2) is 34 minutes for *Am* and 406 minutes for $A\phi$. This corresponds to an improvement of 92% in runtime.

In summary, the toy network experiments show that *Am* leads to significantly improved performance: the frequency of convergence is improved by 83%, and for the converged runs it improves the average convergence speed (measured in number of simulation iterations) by 55%. The method *Am* has increased robustness to both the initial points and the simulator stochasticity. The scenarios account for various levels of congestion, including highly congested scenarios. The performance of *Am* is consistent across these congestion levels. Recall that for all experiments, the simulation-based objective functions could be well approximated by a quadratic function (Figures 4-1, 4-7, and 4-13), hence the benchmark method $A\phi$ is expected to perform well. Even under such scenarios, the proposed method *Am* outperforms the benchmark method $A\phi$. We expect the added value of the proposed method to be even greater for more intricate experiments (e.g., non-convex objective functions with several local minima).

4.3.3 Berlin Case Study Results

We now evaluate the performance of *Am* and of $A\phi$ for the calibration of the Berlin metropolitan area network. Unlike the synthetic case study of the toy network, for Berlin network the true (or best) parameter value, α^*, is unknown. We run the algorithms considering a set of five different initial values, α_0.

Let us first discuss the congestion levels of the Berlin metropolitan network case study. The congestion level depends on the supply parameter, α. Increasing values of α lead to decreasing levels of congestion. Table 4.4 summarizes the congestion levels for three different values of α. The first row of the table considers the lower bound of the feasible region (denoted by α_L and equal to 0.1), the last row considers the upper bound of the feasible region (denoted by α_U and equal to 10), and the second

	City center	Metropolitan area
$\alpha = \alpha_L$	81%	11%
$\alpha = \alpha_{Pre\text{-}calib}$	7%	2%
$\alpha = \alpha_U$	0%	0%

Table 4.4: Congestion levels of the Berlin network with various α

row considers the value currently used in the model, we refer to this value as the pre-calibrated value (denoted by $\alpha_{Pre\text{-}calib}$ and equal to 2). The column denoted by "City center" (resp. "Metropolitan area") displays the percentage of links in the city center (resp. metropolitan area or full network) with a ratio of expected demand to expected supply that is greater than 1. For instance, this table indicates that for $\alpha = \alpha_L$, 81% of the city center links have a ratio that is greater than 1, i.e., they are highly congested. For $\alpha = \alpha_U$, none of the city center links are highly congested. This table shows that the considered feasible region ($\alpha \in [\alpha_L, \alpha_U]$) leads to various levels of congestion ranging from free-flow conditions to highly congested conditions.

Let us proceed as for the toy network and estimate the simulation-based objective function. Figure 4-19 displays the simulation-based objective function estimates (blue circles) with 95% confidence intervals (in red) and the analytical approximation obtained from the network model (black curve). The simulated points are equally spaced. Additional points in the vicinity of the minimum of the simulation-based objective function are simulated in order to estimate the function more accurately near its minimum. The true (or best) supply parameter value α^* is observed to be around 1.1. The green region of Figure 4-19 represents the set of statistically equivalent points: $[1.04, 1.10]$. Solutions within this green region are considered to have converged. The simulation-based and analytical objective functions share similar functional form.

Figures 4-20 to 4-24 each considers a different initial value, α_0. We proceed as for the toy network, and analyze the current iterate (i.e., point with best simulated performance) as a function of the total number of simulated points. Figure 4-20 considers the initial point $\alpha_0 = 0.1$, which is the lower bound of the feasible region (displayed in row 1 of Table 4.1). This initialization leads to highly congested traffic

conditions, and hence to high simulation runtimes. All three solid black lines representing the method Am enter, and stay within, the green region (i.e., they converge) after 3 points are simulated, while all three dashed red lines representing the method $A\phi$ fail to enter the green region before the computational budget is depleted (some cross the region but do not evaluate points within the region). For all other initial points, we observe similar trends: Am converges faster and more often than $A\phi$. The difference in performance is particularly marked in Figure 4-24, where all Am runs converge within 3 simulation points, while none of the $A\phi$ runs converge.

Table 4.5 indicates for each run and each method, the number of points simulated

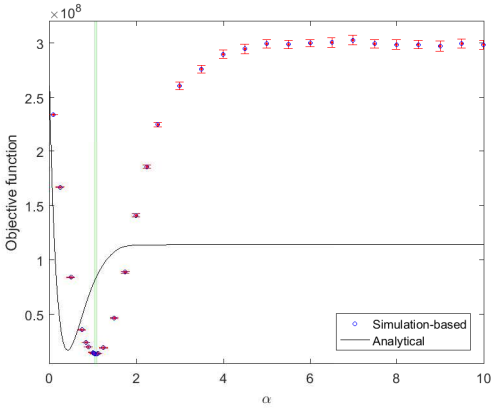


Figure 4-19: Simulation-based and analytical objective functions for the Berlin network

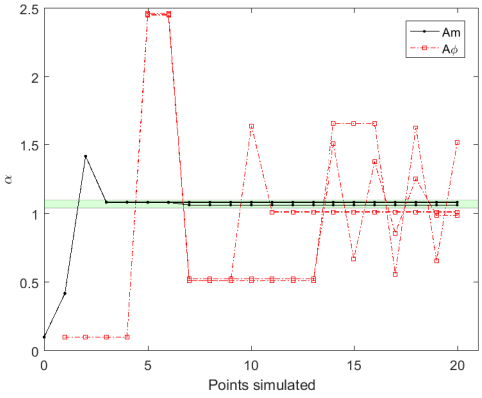


Figure 4-20: Algorithmic solutions as a function of the total number of simulated points, for $\alpha_0 = 0.1$

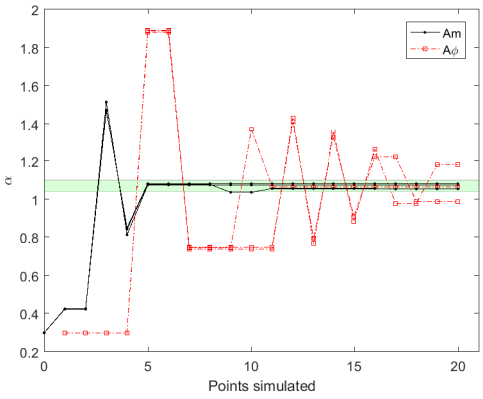


Figure 4-21: Algorithmic solutions as a function of the total number of simulated points, for $\alpha_0 = 0.3$

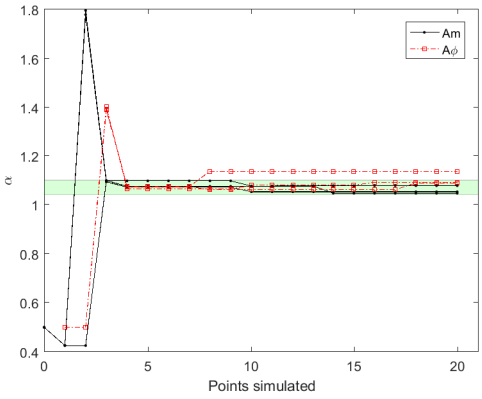


Figure 4-22: Algorithmic solutions as a function of the total number of simulated points, for $\alpha_0 = 0.5$

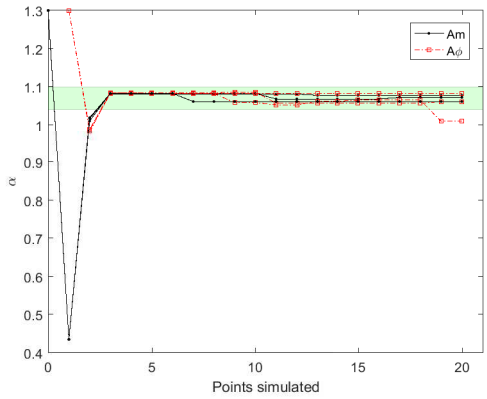


Figure 4-23: Algorithmic solutions as a function of the total number of simulated points, for $\alpha_0 = 1.3$

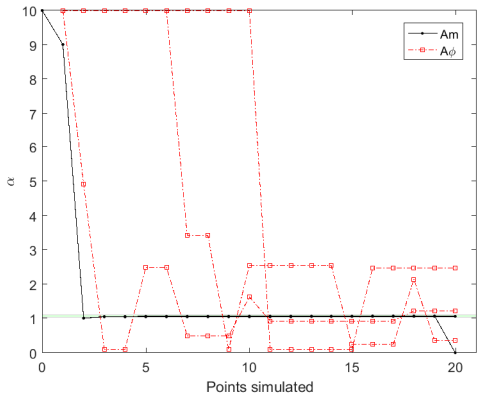


Figure 4-24: Algorithmic solutions as a function of the total number of simulated points, for $\alpha_0 = 10$

until convergence. This table indicates that all Am runs converge, while only 5 of the 15 runs of $A\phi$ converge. When considering only the converged runs, the average number of points simulated until convergence is 4 (resp. 5) for Am (resp. $A\phi$). In other words, the method Am increases by a factor of 3 the frequency of convergence and, when considering only the converged cases Am increases the speed of convergence by 20%. This table also highlights the stark difference in performance when initializing the algorithms with either the lower bound (first row of Table 4.5) or the upper bound (last row). In these two cases, all Am runs converge after 3 points have been simulated, while none of $A\phi$ runs converge.

Table 4.6 presents the simulation runtimes, in minutes, until convergence for the converged cases, and for the non-converged cases, the table indicates the runtime until the simulation budget is depleted. The non-converged runs are underlined. When considering only the converged runs, the method Am reduces the average simulation runtime until convergence by 28% (669 minutes for $A\phi$ and 479 minutes for Am, i.e., a reduction of 3.2 hours). The average runtime, accounting for both converged and non-converged cases, is 479 minutes for Am and 1727 minutes for $A\phi$. The proposed method reduces the runtime by 72%.

In summary, compared to $A\phi$, Am leads to a significantly higher proportion of converged runs, and when considering only the converged runs, Am leads to faster

	Am			$A\phi$		
$\alpha_0 = 0.1$	3	3	3	•	•	•
$\alpha_0 = 0.3$	5	5	11	•	11	•
$\alpha_0 = 0.5$	3	3	3	•	4	4
$\alpha_0 = 1.3$	3	3	3	3	3	•
$\alpha_0 = 10$	3	3	3	•	•	•

Table 4.5: Number of points simulated until convergence for the Berlin network, the symbol • represents non-converged cases

	Am			$A\phi$		
$\alpha_0 = 0.1$	394	424	427	2359	2230	2231
$\alpha_0 = 0.3$	541	539	1683	1800	1312	1667
$\alpha_0 = 0.5$	302	422	684	1905	629	538
$\alpha_0 = 1.3$	413	328	416	448	419	1371
$\alpha_0 = 10$	174	221	223	3140	2912	2948

Table 4.6: Simulation runtimes until convergence for the Berlin network (minutes). For the non-converged cases, the statistic reported is underlined and indicates the runtime until the computational budget is depleted

convergence.

Let us now compare in more detail the performance of the solutions derived by each method. Recall that each method yields a total of 15 solutions (i.e., 3 SO runs for each of the 5 initial points). For each solution, we evaluate its performance based on estimates of the objective function as follows. We run 50 simulation replications to yield 50 observations of the objective function. For each simulation replication, 100 sequential assignment iterations are carried out to obtain one objective function realization. We then use the 50 simulation replications to construct a cumulative distribution function (cdf) of the objective function. We compare the performance of the various solutions by comparing their cdf's.

Figure 4-25 displays the cdf's for all 15 solutions of both Am (solid black curves) and $A\phi$ (dash-dotted red curves). The x-axis of the figure represents objective function values. For a given x value, the corresponding y value of a curve gives the proportion of simulation replications (out of the 50 replications) that have objective function values smaller than x. Hence, the more a cdf curve is shifted to the left, the

higher the proportion of low objective function values, i.e., the better the performance of the corresponding solution. This figure shows that 3 $A\phi$ solutions have significantly worse performance than all other solutions. These 3 solutions correspond to the 3 rightmost curves.

Figure 4-26 excludes these 3 cdf's and plots the remaining cdf's. Here, the black curves are almost all to the left of the red curves. In other words, most Am solutions outperform most $A\phi$ solutions. Additionally, most black curves are very similar. This means that the solutions proposed by Am have similar performance across most of the 15 experiments. Actually, the 14 best Am solutions have statistically equivalent performance. Thus, the performance of Am is similar for experiments initialized with different initial points. Additionally, for a given initial point, the performance is similar across SO runs. Therefore, the performance is robust to both the quality of the initial points and the simulator stochasticity. On the other hand, the performance of $A\phi$ solutions varies significantly across experiments, meaning that $A\phi$ is less robust to the quality of the initial points and to the stochasticity of the simulator.

For each method, we select the solution with the best performance. This is defined as the solution with the smallest objective function estimate. The best Am (resp. $A\phi$) solution corresponds to a supply parameter value α of 1.075 (resp. 1.061). Hereafter, these values are denoted by *Am-best* and *Aϕ-best*, respectively. Let us now

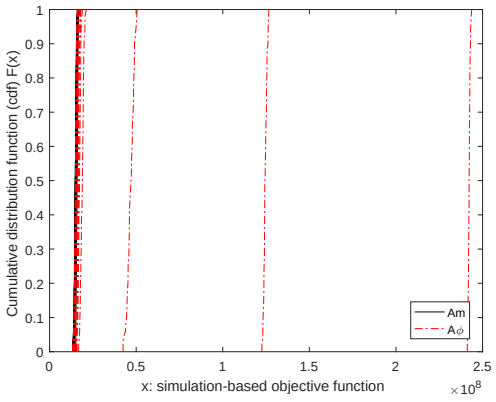


Figure 4-25: Objective function distributions for all Am and all $A\phi$ solutions

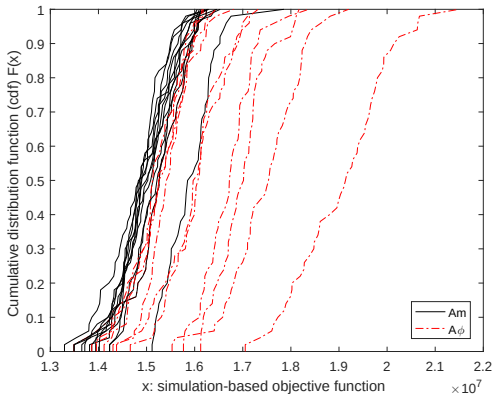


Figure 4-26: Objective function distributions for all Am solutions and 12 of the 15 $A\phi$ solutions

compare the performance of three points: *Am-best*, *Aϕ-best*, and the pre-calibrated value (denoted by *Pre-calib*) which is currently used for the Berlin network model. Figure 4-27 displays the cdf of the objective function for each of these three points. The solid black curve corresponds to *Am-best*, the dash-dotted red curve corresponds to *Aϕ-best*, and the dotted black curve corresponds to *Pre-calib*. This plot indicates that *Am-best* and *Aϕ-best* outperform *Pre-calib*. Figure 4-28 displays the cdf's of only *Am-best* and *Aϕ-best*. It indicates that *Am-best* outperforms *Aϕ-best*. This difference is statistically significant. The average objective function is improved by 6%.

Let us now analyze how these supply calibration results complement the demand calibration results derived in Chapter 3. The above analysis compares the performance of three α values: the pre-calibrated value ($\alpha = 2$), the best value proposed by *Am* ($\alpha = 1.075$), and the best value proposed by *Aϕ* ($\alpha = 1.061$). The algorithm proposed in Chapter 3 is a demand calibration algorithm. Its performance was illustrated on the same Berlin network with the same objective function, based on the calibration of the travel time coefficient of the route choice model. This demand parameter is denoted by θ. Chapter 3 calibrates θ assuming a fixed supply parameter α set to the pre-calibrated value ($\alpha = 2$). In the present chapter, we have calibrated α assuming the demand parameter is fixed and set to its pre-calibrated

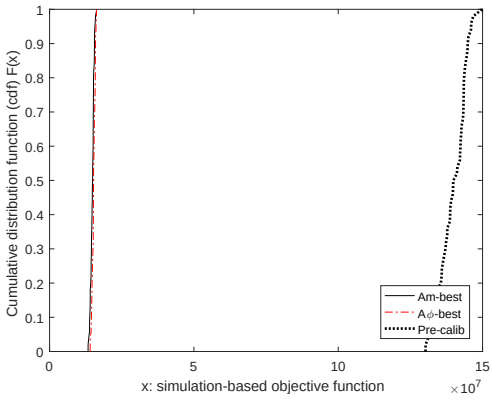


Figure 4-27: Objective function distributions for the best solutions of each method and for the pre-calibrated value

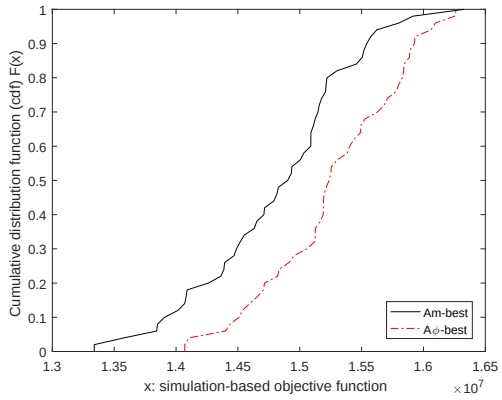


Figure 4-28: Objective function distributions for the best solutions of each method

value ($\theta = -6$). In Chapter 3, the performance of three θ values were also compared: the pre-calibrated value ($\theta = -6$), the best value obtained by the proposed demand calibration algorithm (which was denoted by Am) ($\theta = -4.7$), and the best value proposed by the benchmark demand calibration algorithm (which was denoted by $A\phi$) ($\theta = -2.8$).

Figure 4-29 displays 9 cdf curves of the objective function. They correspond to the 9 combinations of the above 3 supply solutions ($\alpha \in \{2, 1.075, 1.061\}$) and the above 3 demand solutions ($\theta \in \{-6, -4.7, -2.8\}$). For a specific combination, we construct the cdf of the simulation-based objective function as described before. The 3 blue curves correspond to cases where the supply parameter takes its pre-calibrated value ($\alpha = 2$) and the demand parameter takes one of the 3 values: its pre-calibrated value (dotted blue curve), the best value as proposed by the demand calibration algorithm of Chapter 3 (solid blue curve), and the best value as proposed by the benchmark method of Chapter 3 (dash-dotted blue curve). The three black (resp. red) cdf curves of Figure 4-29 correspond to the three cases with α set to the best value obtained from the proposed (resp. benchmark) supply calibration algorithm. The black and red curves of this figure significantly outperform the blue curves. This means that regardless of the value of the demand parameter, the biggest improvement in performance can be obtained by improving the fit of the supply parameter. The 3 blue cdf's correspond to the 3 cdf's displayed in Figure 3-24 of Chapter 3.

Figure 4-30 displays only the black and red cdf's of Figure 4-29. The black (resp. red) curves correspond to cases where α takes the best value obtained by the proposed (resp. benchmark) supply calibration algorithm. The dotted curves correspond to θ set to its pre-calibrated value. The solid (resp. dash-dotted) curve corresponds to θ set to the best solution obtained by the proposed (resp. benchmark) demand calibration algorithm. Note that the black and the red dotted curves correspond to the black and red curves of Figure 4-28. Figure 4-30 indicates that the best combination corresponds to the case where α takes the value obtained by the proposed supply calibration algorithm and θ takes the value obtained by the proposed demand calibration algorithm of Chapter 3. This difference in performance compared to the

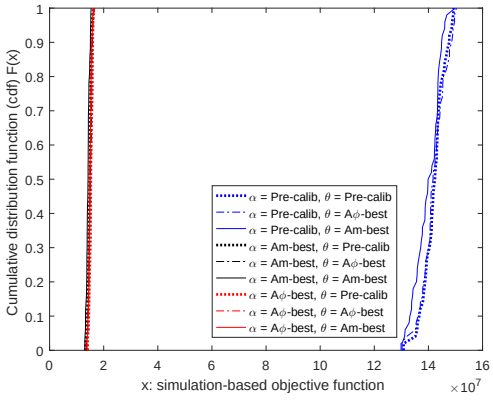


Figure 4-29: Objective function distributions for combinations of the best values (of α and θ) and the pre-calibrated values (of α and θ)

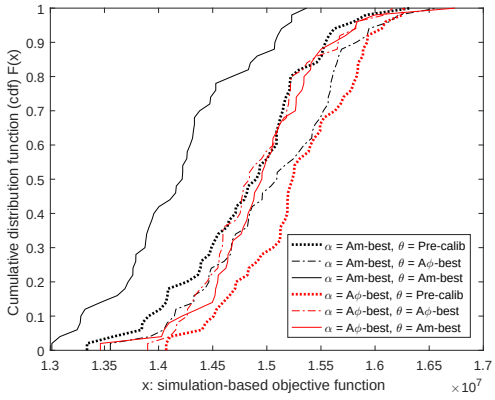


Figure 4-30: Objective function distributions for combinations of the best values (of α and θ) and the pre-calibrated value of θ

other solutions is statistically significant.

In summary, for all runs, the proposed supply calibration algorithm *Am* converges, while the benchmark method converges only one third of the time. When considering the converged runs, the proposed method converges faster than the benchmark method: it requires, on average, 20% less points to converge with a 28% reduction in simulation runtime. The performance of the proposed method is robust to the quality of the initial points and to the stochasticity of the simulator.

We now analyze the computational runtime of the algorithm *Am*. At every iteration of the algorithm *Am*, the two steps with highest computational cost are the simulation evaluation (step 4 of Figure 2-2) and solving the (metamodel) subproblem optimization (step 3a of Figure 2-2). For all experiments of this Berlin case study, we consider the computational runtimes of each of these two steps.

Figure 4-31 displays two cdf curves, one for each step. Each curve is constructed based on 15 observations, one for each SO run. The x-axis considers the runtime in minutes. For a given x value, the y-axis considers the proportion of observations (out of the 15) that take x minutes or less to be completed. The red dash-dotted curve represents the total simulation runtime per SO run (i.e., total simulation runtime for all iterations of a given SO run). The solid black cdf represents the total subproblem optimization runtime per SO run (i.e., total time spent solving the subproblem for

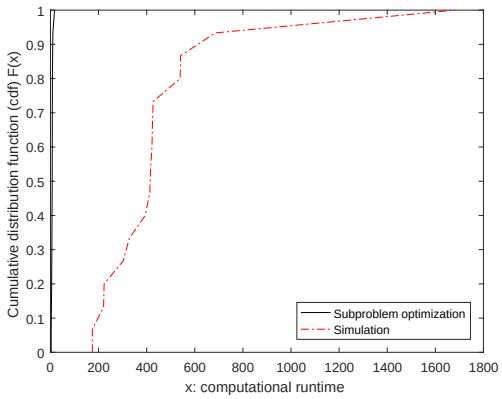


Figure 4-31: Cumulative distribution of the computational runtimes of the simulator and of the metamodel subproblem optimization

all iterations of a given SO run). The black cdf is to the left of the plot, it is almost parallel to, and barely distinguishable from, the y-axis. The total simulation runtimes are within the range: $[174, 1683]$ minutes, while the total subproblem optimization runtimes are within the range: $[3, 17]$ minutes. Over 90% of the total subproblem optimization runtimes are smaller than 10 minutes. This figure highlights that the computational cost of solving the metamodel subproblem is negligible compared to that of evaluating the simulation model. Recall that the metamodel subproblem contains the proposed analytical network model as a set of constraints. Hence, this figure illustrates the computational efficiency of the proposed analytical network model.

4.4 Conclusions and Discussion

This chapter proposes a supply calibration algorithm that is suitable for stochastic and large-scale network simulators. The proposed approach is based on a metamodel simulation-based optimization algorithm. The metamodel combines information from the simulator with information from an analytical network model. This chapter formulates a highly computationally efficient and scalable network model that is used as part of the metamodel. The network model is analytical and differentiable. It is formulated as a system of nonlinear equations, the dimension of which scales linearly with the number of links in the network and scales independently of the link

attributes (e.g., link length). It is this network model formulation that leads to an SO algorithm that is both highly efficient and robust to both the stochasticity of the simulator and to the quality of the initial points. The performance of the proposed approach is evaluated with a small synthetic case study and a case study of the Berlin metropolitan area. The performance is compared to that of an SO approach that differs only in that the metamodel does not use information from the analytical network model. The proposed approach outperforms the benchmark approach: it converges both more frequently and faster.

Chapter 5

Origin-Destination (OD) Demand Calibration of Stochastic Traffic Simulators

The contents of this chapter are available as: Zhang, C. and Osorio, C. (2018). Efficient offline calibration of origin-destination (OD) demand for large-scale stochastic traffic models. *Technical report, Massachusetts Institute of Technology.*

5.1 Introduction

We focus here on the offline calibration of demand as defined by origin-destination (OD) matrices. This is known as OD calibration and is the most widely studied and traditionally challenging calibration problem.

The transportation demand, represented by time-dependent OD demand matrices, is an essential input for traffic simulators to evaluate, for example, traffic management and planning strategies. Each entry of an OD demand matrix contains the number of trips made during a certain time interval between an origin zone and a destination zone. OD calibration has been one of the main problems attracting attention from both transportation researchers and practitioners.

Compared to the calibration problems addressed in Chapters 3 and 4, the problem

of OD calibration is particularly challenging for the following reasons. First, it is a high-dimensional problem: the number of non-zero entries of an OD demand matrix is typically in the order of 1,000. Second, there is most often only a small set of links that have sensors (i.e., for which we have field measurements). This makes the optimization problem under-determined. This has led the calibration community to include in the formulation of the calibration problem information from a prior (also known as a seed) OD matrix. Specifically, the objective function includes a term that aims to reduce the distance between a solution and the prior OD matrix. This prior matrix is typically estimated based on census data or past static traffic assignment studies, possibly outdated. Third, when using a detailed simulation-based traffic model (rather than an analytical model), the problem is often non-convex and contains many local minima. Fourth, when using a stochastic simulator, the problem becomes a simulation-based optimization (SO) problem. In the field of SO, problems with dimensions in the order of 200 are considered high-dimensional. Hence, there is currently a lack of suitable algorithms designed for high-dimensional SO problems. As discussed in Section 2.3, the objective function has no closed-form expression and can only be estimated via computationally inefficient simulation evaluations. Fifth, the use of a computationally costly stochastic simulator makes the use of derivative-based SO algorithms prohibitive. Instead, derivative-free algorithms which do not require estimation of the derivatives of the simulation-based function are used. However, these are less efficient, and hence less appropriate for high-dimensional problems than their derivative-based counterparts.

In this chapter, we perform OD calibration for a Berlin metropolitan network with over 11,000 nodes and 24,000 links. In Chapter 3, we proposed a metamodel formulation for the calibration of a one-dimensional demand parameter (i.e., the travel time coefficient of a route choice model). From a methodological perspective, the main distinction between this chapter and our previous work lies in the key component of metamodel SO which is the formulation of the analytical network model. More specifically, the formulation of Chapter 3 is more intricate in that it accounts for endogenous traffic assignment. Hence, it is formulated as a system of nonlinear (rather

than linear) equations. It can provide a more accurate approximation of the SO objective function, yet at a higher computational cost. In Chapter 3, we used it for a one-dimensional problem. Its use for high-dimensional problems, such as OD calibration problems, has yet to be explored. In addition, the objective function for the OD calibration problem in this chapter is different from that used in the previous chapters. In this chapter, the objective function is defined as the sum of: (i) the distance between field observations and simulated observations, and (ii) the distance between the prior OD matrix and the solution. The latter term is used to improve the under-determination of the calibration problem.

The remainder of this chapter is organized as follows. Section 5.2 formulates the OD calibration problem and presents the proposed methodology. Section 5.3 compares the performance of the proposed approach to a benchmark approach, which differs only in that it does not use information from the analytical network model. We carry out a case study for a synthetic toy network and one for the large-scale Berlin (Germany) metropolitan network. The main conclusions are presented in Section 5.4. Appendix A presents the SO algorithm. Appendix C gives the implementation details.

5.2 Methodology

5.2.1 Problem Formulation

Consider an urban network divided into traffic analysis zones (i.e., TAZ), which are spatial units commonly used in transportation planning models (page 57, Miller and Shaw, 2001). Travel demand in the network is represented by an origin-destination (OD) demand matrix that states the expected number of trips between all pairs of TAZ. A given pair is referred to as an OD pair. An OD matrix is defined for a given time interval. Basically, the OD calibration problem is to determine an OD matrix that results in simulated traffic performance metrics that are similar to those estimated with field data. We focus on a formulation that considers the most widely available type of traffic data: link traffic counts. We consider a single OD matrix for the time

period of interest. The goal is to identify the OD matrix that yields simulated link flows similar to field link counts.

To formulate the OD calibration problem, we introduce the following notation.

d_z	expected travel demand for OD pair z;
f	simulation-based objective function;
F_i	flow on link i as defined by the simulator;
y_i	average flow on link i estimated from field data;
$\tilde{d}_z$	prior value for the expected demand for OD pair z;
δ	weight parameter for prior information;
$\mathcal{I}$	set of links with sensors;
$\mathcal{Z}$	set of OD pairs.

The offline OD calibration problem is formulated as follows.

$$\min_{d} \qquad f(d) = \sum_{i \in \mathcal{I}} \left(y_i - E[F_i(d, u_1; u_2)]\right)^2 + \delta \sum_{z \in \mathcal{Z}} \left(\tilde{d}_z - d_z\right)^2 \tag{5.1}$$

$$d \geq 0. \tag{5.2}$$

The first term of the objective function, f, represents the distance between the observed field traffic counts, y_i, and the simulation-based expected flow, $E[F_i(d, u_1; u_2)]$. The latter is a function of the decision vector, d, a vector of endogenous simulation variables, u_1 (e.g., link speeds, travel times, queue-lengths) and a vector of exogenous simulation parameters, u_2 (e.g., network topology, traffic management strategies). Hereafter, we use simplified notation and denote $E[F_i(d, u_1; u_2)]$ by $E[F_i(d)]$. The second term represents the distance between the proposed OD matrix, d, and a prior OD matrix, $\tilde{d}$. The latter is referred to in some papers as the seed or the initial OD matrix. It is typically estimated from historical data (e.g., census data) and from a static traffic assignment analysis. Case studies that provide more details on this prior OD matrix component can be found in Balakrishna (pages 158-159, 2006). The factor, δ, is an exogenous and fixed weight scalar that represents the relative weight

of the prior OD demand information versus the traffic count data. Constraint (5.2) is an analytical (i.e., not simulation-based) and differentiable lower bound constraint that ensures positive OD demand. Hence, the problem consists of a simulation-based objective function with analytical bound constraints.

Problem (5.1)-(5.2) is challenging to address for the following reasons. The decision vector, d, is typically high-dimensional, with a dimension in the order of several thousands. The number of links with sensors, card($\mathcal{I}$), is typically small compared to the total number of links in the network. The problem is under-determined and contains numerous local minima. Among the many solutions to this problem some will be physically plausible (i.e., they will be consistent with the land-use and activity patterns of the city), while others may merely represent mathematically valid, yet physically implausible, solutions. The goal of the second term of the objective function is to enable the algorithm to identify physically plausible solutions. The expected flow function, $E[F_i(d)]$, is a nonlinear function that lacks sound mathematical properties, such as convexity, and has no closed-form expression available. It can be estimated by running multiple simulation replications, each of which is computationally costly to evaluate.

Let us illustrate the intricate spatial-temporal traffic phenomena represented by the stochastic traffic simulator that leads to expected link flows. The expected flow function can be defined as:

$$E[F_i(d, u_1; u_2)] = \sum_{z \in \mathcal{Z}} d_z \sum_{r \in R_z} \Delta_{ri} P_z(r|d, u_1; u_2), \tag{5.3}$$

where $P_z(r|d, u_1; u_2)$ is the probability that a trip-maker traveling on OD pair z selects a route r among the set of feasible routes for OD pair z, R_z, and Δ_{ri} equals 1 if route r traverses link i and 0 otherwise. This probability depends on the link travel times, which in turn depend on link flows. Hence, the expected link flow function is typically estimated iteratively. In a traffic simulator, these iterations can be interpreted as a learning process over subsequent days, where each day all trip-makers make route choices according to the most recent network conditions u_1, followed by a simulation

of the corresponding vehicle flows through the network, which in turn updates the network conditions.

5.2.2 Metamodel Formulation

To address the OD calibration problem, we use a metamodel SO approach as described in Section 2.3. This section formulates a novel metamodel specifically for OD demand calibration.

Most often, the metamodel is defined as an analytical approximation of the objective function (f of Equation (5.1)). In this work, we propose to use one metamodel for each of the simulation-based terms in the objective function. In other words, for each link i with sensor, we define a metamodel to approximate the expected link flow function, $E[F_i(d)]$. To formulate the metamodel, we introduce the following notation. The index k refers to a given SO iteration, and the index i refers to a given link.

m_k^{obj}	objective function of the analytical optimization problem;
$m_{i,k}$	analytical approximation of the expected flow for link i;
$\beta_{i,k}$	vector of parameters for metamodel $m_{i,k}$;
$\beta_{i,k,j}$	element j of vector $\beta_{i,k}$;
λ_i	expected demand of link i as approximated by the analytical network model;
p_{ij}	turning probability from link i to link j;
$\tilde{p}_{zi}$	proportion of demand from OD pair z that takes a route that starts with link i;
q	vector of exogenous parameters of the analytical network model;
$\mathcal{L}$	set of links in the network.

At iteration k of the SO algorithm, the analytical optimization problem solved

(Step 2a of Figure 2-2) is defined as follows.

$$\min_{d} \qquad m_k^{obj}(d) = \sum_{i \in \mathcal{I}} \left(y_i - m_{i,k}(d; \beta_{i,k}, q)\right)^2 + \delta \sum_{z \in \mathcal{Z}} \left(\tilde{d}_z - d_z\right)^2 \tag{5.4}$$

$$d \geq 0 \tag{5.5}$$

$$m_{i,k}(d; \beta_{i,k}, q) = \beta_{i,k,0} \lambda_i(d) + \left(\beta_{i,k,1} + \sum_{z=1}^{\text{card}(\mathcal{Z})} \beta_{i,k,z+1} d_z\right) \tag{5.6}$$

$$\lambda_i(d) = \sum_{z \in \mathcal{Z}} \tilde{p}_{zi} d_z + \sum_{j \in \mathcal{L}} p_{ji} \lambda_j(d). \tag{5.7}$$

Hereafter, we refer to the above optimization problem as the *metamodel optimization problem*. This problem differs from the initial SO problem (i.e., Problem (5.1)-(5.2)) in three ways. First, the simulation-based expected link flow function for link i, $E[F_i(d)]$, is replaced by the analytical metamodel, $m_{i,k}$. The latter depends on the decision vector d, on a vector of parameters $\beta_{i,k}$ that are both link- and iteration-specific, and on a vector of exogenous parameters q (e.g., network topology, link attributes). Second, it is an analytical and differentiable optimization problem. Hence, we can use traditional algorithms to address it. Third, it has a set of 2 additional constraints. Equation (5.6) defines the metamodel as the sum of two terms. These terms are known in the SO literature as the physical component (denoted by $\lambda_i(d)$) and the functional or general-purpose component (which is the term in parentheses), respectively. The goal of the physical component is to provide a problem-specific approximation of the simulation-based function, $E[F_i(d)]$, while that of the functional component is to provide a general-purpose (i.e., valid for all types of problems) approximation of the simulation-based function, $E[F_i(d)]$. The functional component is typically chosen based on its mathematical properties (e.g., convexity) such as to guarantee asymptotic convergence properties for the SO algorithm. Functional components often used include low-order polynomials, radial-basis functions, and Kriging functions. For a more detailed classification and description of metamodels, see Søndergaard (pages 12-33, 2003).

Equation (5.6) defines the functional component as a linear (polynomial) function.

The physical component is defined by Equation (5.7). We refer to the latter as the analytical network model. It is a linear system of equations, which represents demand conservation. It states that the (analytical) expected demand on link i is defined as the sum of the expected demand that arises from trips that start on link i (first summation term in Equation (5.7)) and the expected demand that arises from trips that arise from upstream links (second summation term in Equation (5.7)). Note that the expected link demand, λ_i, is used as an approximation for the expected link flow. The term p_{ij} represents an exogenous probability of turning from link i to link j. The term $\tilde{p}_{zi}$ represents an exogenous fixed proportion of demand of OD zone z which starts trips on link i (i.e., it enters the network through link i). Both p_{ij} and $\tilde{p}_{zi}$ are estimated, prior to optimization through simulation by using the prior OD matrix ($\tilde{d}$ of Equation (5.1)).

The main challenge in metamodel SO is to formulate a metamodel that both: (i) leads to solutions to the metamodel optimization problem (i.e., Problem (5.4)-(5.7)) that are good solutions to the original SO Problem (5.1)-(5.2) (i.e., they yield small SO objective function estimates), and (ii) is computationally tractable. The latter is essential because the metamodel optimization problem is solved at *every* iteration of the SO algorithm. Hence, it needs to be solved in a computationally efficient way. Otherwise, one is better off allocating the computational resources to running additional simulations rather than to solving this analytical (and approximate) problem. For the metamodel to achieve both of these goals, it should be able to: (i) describe well the network-wide interactions between OD demand and link flows; (ii) be sufficiently scalable that it can be used for both large-scale networks and for high-dimensional decision vectors; and (iii) be sufficiently efficient such that the metamodel optimization problem can be solved quickly.

The formulation proposed above achieves these goals. It is a scalable formulation: the analytical network model (Equation (5.7)) is defined as a system of equations with a dimension that scales linearly with the number of links in the network and that does not depend on other link attributes (e.g., link lengths). It is a computationally efficient formulation: the metamodel optimization problem is an analytical

problem with lower bound constraints and linear equality constraints. In particular, the analytical network model is formulated as a system of linear equations. Hence, the metamodel optimization problem can be addressed by a variety of commercial solvers.

Nonetheless, the scalability and computational tractability come at the cost of using a simple low-resolution analytical network model. More specifically, the analytical network model is a stationary model, which does not describe the temporal propagation and dissipation of congestion. As described in Section 5.2.1, the simulator has endogenous traffic assignment. In other words, it accounts for how the spatial distribution of demand, as described by the OD demand matrix, impacts the link costs (e.g., travel times, speeds) and thus the route choices. The analytical network model assumes exogenous traffic assignment. In other words, it does not capture the interactions between the spatial distribution of demand, d, and the traffic assignment, p_{ij} and $\tilde{p}_{zi}$. Nonetheless, the experiments of Section 5.3 indicate that despite these simplifications, the metamodel leads to good analytical approximations for all levels of congestion.

5.3 Case Studies

We use the proposed methodology to address calibration problems for a synthetic toy network (Section 5.3.1) and for a real-world network of the Berlin metropolitan area (Section 5.3.2). For both networks, we compare the performance of our proposed method to that of an SO method that differs only in the metamodel. More specifically, the benchmark method does not include the analytical network model. In other words, the metamodel does not contain a physical or problem-specific component. This is obtained by setting the term $\beta_{i,k,0}$ of Equation (5.6) to zero. The comparison serves to evaluate the added value of the problem-specific structural information provided by the analytical network model. The proposed (resp. benchmark) method is denoted by Am (resp. $A\phi$). For both case studies, the MATSim simulator is used (Horni *et al.*, 2016).

5.3.1 Synthetic Toy Network

Experimental Design

The synthetic toy network is shown in Figure 5-1. Its topology is adapted from the network in Shao *et al.* (2015). The network contains a total of 11 links, 10 nodes, and 2 OD pairs (from node 1 to node 9, and from node 2 to node 10). For each OD pair, there are two alternative routes: a straight route and a route that passes through nodes 4 and 7. All links are uni-directional single-lane roads. The full specification of the link attributes is given in Table C.3 of Appendix C. We focus on a 1 hour time period.

The experimental setup is the same as that shown in Table 3.2 except for the first three rows, which are detailed in Table 5.1. Table 5.1 indicates that the bounds for the decision vector are $[0, +\infty)$ (first row). The second row presents the two synthetic "true" OD demand cases that are assumed. The first case considers an asymmetrical OD demand of 800 veh/hr for OD pair 1-9 and 1,400 veh/hr for OD pair 2-10. The second case considers a symmetrical OD demand of 1,400 veh/hr for both OD pairs. These two cases lead to different levels of congestion throughout the network. More specifically, the ratio of expected demand to expected supply varies across the links in case 1 (resp. case 2) from 0.44 to 0.80 (resp. 0.60 to 0.81). For each true OD demand case, synthetic traffic counts are obtained, via simulation, on links 5, 6, and 7. Note that this leads to an OD calibration problem that is not under-determined. The weight factor δ (of Equation (5.1)) is set to 0.01 (row 3).

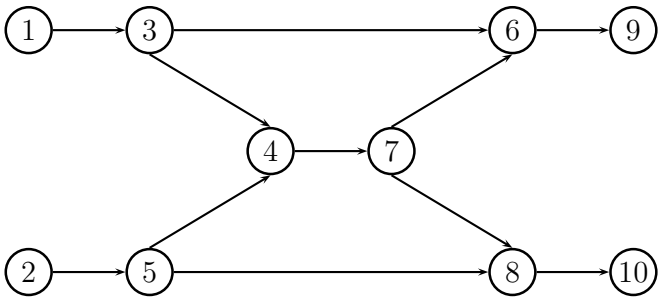


Figure 5-1: Synthetic toy network

Table 5.1: Experimental design

	Synthetic toy network	Berlin network
Bounds for d values (veh/hr)	$[0, +\infty)$	$[0, +\infty)$
True d values (veh/hr), d^*	$\{[800, 1400]; [1400, 1400]\}$	N.A.
Weight factor of the prior OD matrix, δ	0.01	0.01

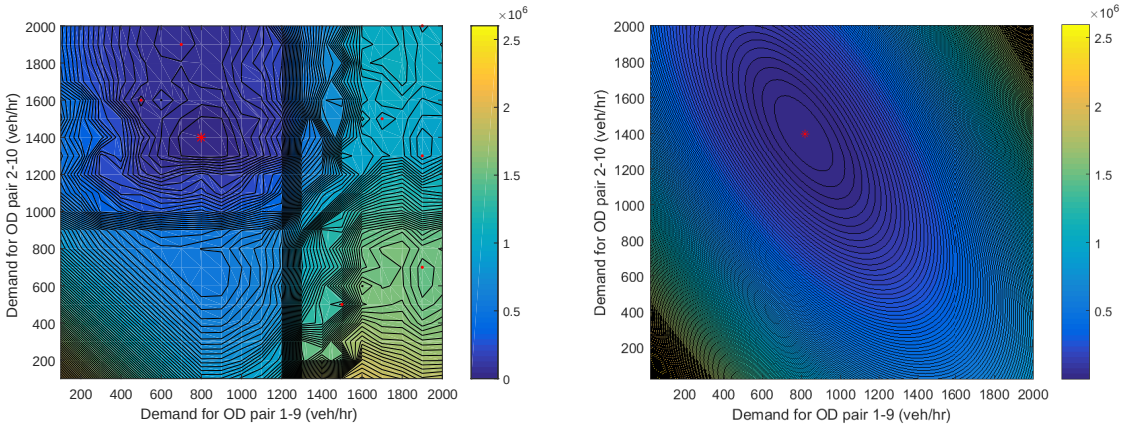
The prior OD matrix ($\tilde{d}$ of Equation (5.1)) is obtained by perturbing the true OD demand, d^*, as follows. For each OD pair z, its demand is defined as the sum of its true demand, d^*_z, and of a random perturbation term which is uniformly sampled from the interval $[-0.3s, 0.3s]$. The scalar s denotes the maximum entry of the true OD matrix, d^*. The factor 0.3 can be interpreted as 30% variations of this maximum value s. After this perturbation, any negative OD demand terms are set to zero.

For each true OD matrix, we consider six scenarios with different initial points (i.e., different starting values for the optimization algorithms). Three initial points are obtained by perturbing the prior OD matrix in the same way we perturb the true OD matrix described above. We refer to these points as perturbed initial points. The remaining three initial points are obtained by sampling each OD entry independently and uniformly in the interval $[0, 2000]$. We refer to these points as random initial points.

We define a scenario as a combination of a true OD demand and an initial point. There are 2 true OD demands and 6 initial points, which gives a total of 12 scenarios. For each scenario, we run each SO algorithm three times. This serves to account for the impact of the simulators' stochasticity on the algorithm's performance. The experiments are carried out on a standard laptop with a 4-core Intel i7-3740QM processor and 8GB RAM.

Numerical Results

We first consider the asymmetrical true OD demand (i.e., $[800, 1400]$ veh/hr). Figure 5-2a illustrates how even such a simplistic toy experiment can lead to intricate objective functions. It displays a contour plot of the SO objective function (i.e., f of Equation (5.1)). This function is estimated by sampling equally spaced and uni-



(a) Simulation-based objective function

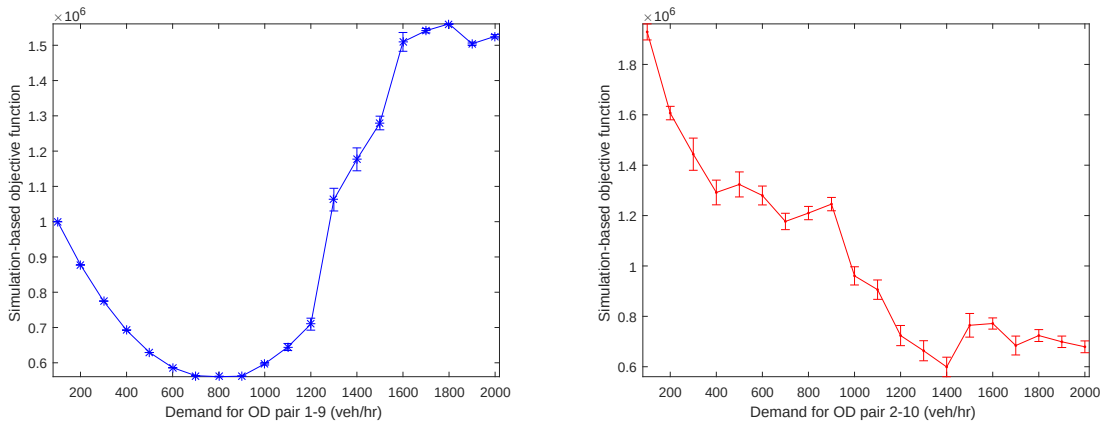
(b) Objective function approximation provided by the analytical network model

Figure 5-2: Objective function for true demand $d^* = [800, 1400]$ (veh/hr)

formly distributed points in the range $[0, 2000]$ with increments of 100 along each of the two demand dimensions. Each estimate is obtained from 10 simulation replications. Based on this estimation, there are a total of 8 local minima (marked with red dots), which include one global minimum (marked with a red asterisk).

The approximation of the SO objective function as derived by the analytical network model is displayed as a contour plot in Figure 5-2b. To compute this function, for a given OD matrix, d, we solve the system of linear equations defined by Equation (5.7). We then consider Equation (5.1) and replace $E[F_i(d)]$ with $\lambda_i(d)$. There is a single local and global minimum (marked with a red asterisk). This analytical function, unlike its simulation-based counterpart, is convex. The comparison of Figures 5-2a and 5-2b indicates that the analytical network model closely approximates the simulation-based objective function. Importantly, the location of the global minimum is accurate. Additionally, this figure indicates that the use of the analytical network model allows us to address an intricate SO problem with non-convex objective function by solving a series of convex and analytical optimization problems.

The two plots of Figure 5-3 each displays a one-dimensional cut of the simulation-based objective function along with corresponding 95% confidence intervals. The left plot varies the demand for the first OD pair (node 1 to node 9), d_1, and leaves the demand for the second OD pair, d_2, fixed to its underlying true value (i.e., 1,400



(a) Cut for OD pair 1

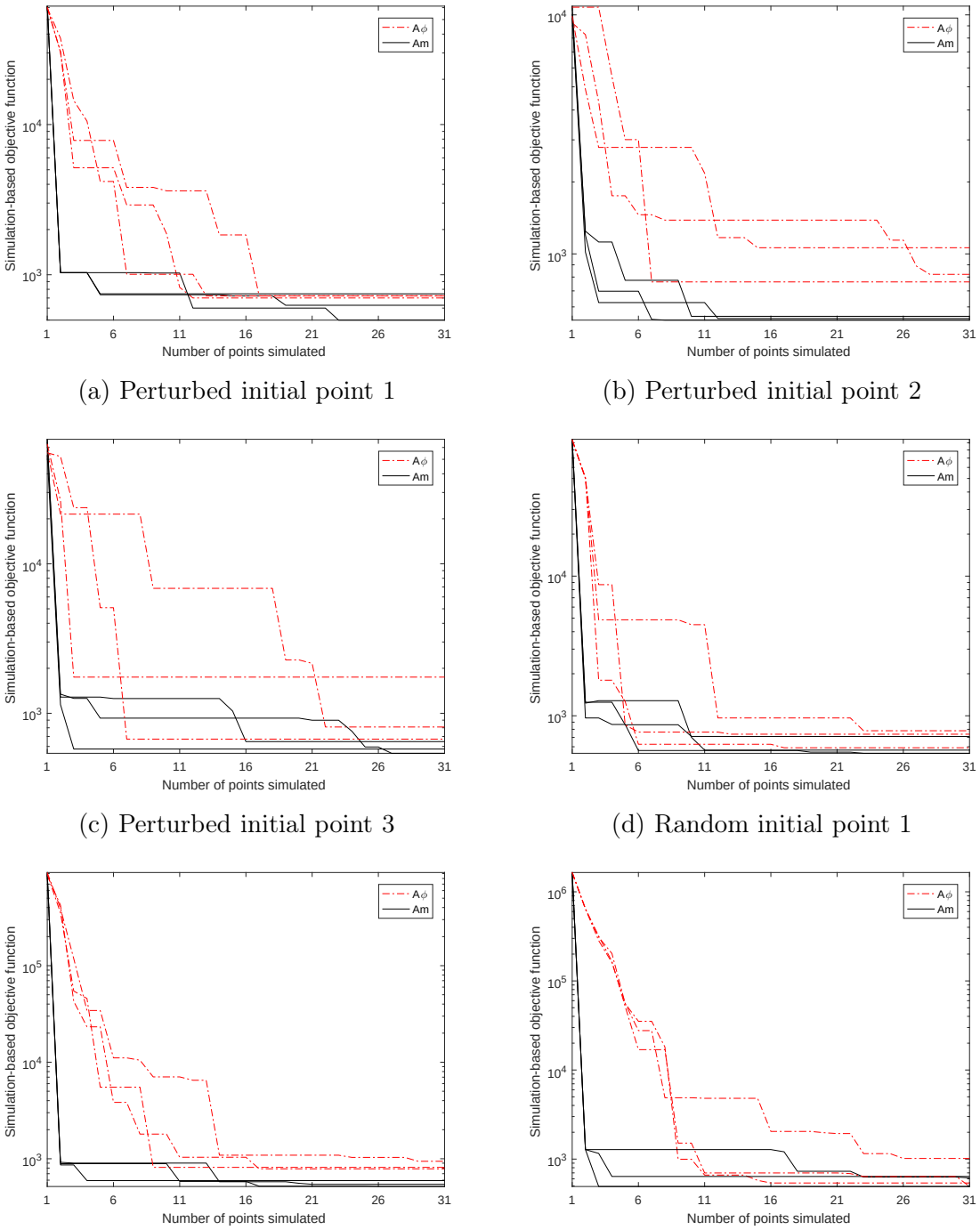
(b) Cut for OD pair 2

Figure 5-3: One-dimensional cut of the simulation-based objective function for true demand $d^* = [800, 1400]$ (veh/hr)

veh/hr). Similarly, the right plot varies the demand for the second OD pair (node 2 to node 10), d_2, and sets d_1 to its underlying true value (i.e., 800 veh/hr). Both plots indicate a non-convex function with multiple local minima. Figures 5-2a and 5-3 illustrate that even for such a simplistic toy network, the SO objective function is an intricate (e.g., non-convex) function with numerous local minima.

Figure 5-4 displays six plots. Each plot considers a different scenario. Plots 5-4a to 5-4c consider the three scenarios with perturbed initial points. Plots 5-4d to 5-4f consider the three scenarios with random initial points. Each plot displays three solid black (resp. dash-dotted red) lines, which correspond to each of the 3 runs of the algorithm Am (resp. $A\phi$). The x-axis displays the computational budget consumed so far (i.e., number of points simulated). The y-axis represents the corresponding simulation-based objective function of the current iterate (i.e., point with the best simulated performance). Note that the y-axis is displayed with a logarithmic scale.

For all Am runs of all plots, the lines of Am achieve a significant reduction of the objective function once the second point is simulated (i.e., at $x = 2$). For Am, the second point to be simulated is that obtained by solving the calibration problem using only the analytical network model. In other words, we minimize the function displayed in Figure 5-2b. Recall that the y-axis uses a logarithmic scale. Hence, the use of the analytical network model improves the objective function by several



(a) Perturbed initial point 1

(b) Perturbed initial point 2

(c) Perturbed initial point 3

(d) Random initial point 1

(e) Random initial point 2

(f) Random initial point 3

Figure 5-4: Objective function estimate of the current iterate as a function of the total number of simulated points, for true demand $d^* = [800, 1400]$ (veh/hr)

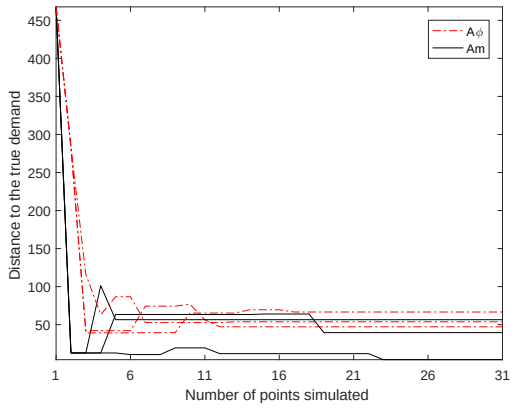
orders of magnitude, and this holds for all initial points. Thus, the use of an analytical network model makes the SO algorithm robust to the quality of the initial

points. For all plots, as the iterations advance, Am identifies points with improved performance. More specifically, it identifies points that outperform that obtained by optimizing the analytical network model alone (i.e., the point obtained at $x = 2$). Overall, the benchmark algorithm $A\phi$ performs well: it reduces the objective function significantly from the initial point. Nevertheless, Am is able to achieve good performance faster (i.e., with fewer simulated points) compared to $A\phi$. In addition, once the computational budget is depleted, the solutions of Am lead systematically to small objective function estimates, while this is not always the case for the solutions derived by $A\phi$. As an example, for Figure 5-4b, the final solutions proposed by Am have a performance that is on average 39% better than those proposed by $A\phi$. More specifically, the average (over SO runs) of the objective function estimates of the final solutions is 537.6 for Am and 881.8 for $A\phi$.

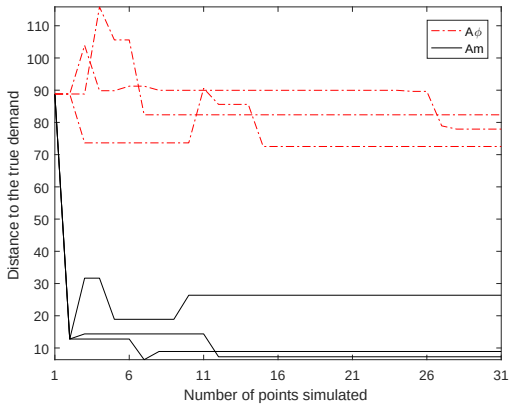
Since we know the true OD demand for these synthetic experiments, we can evaluate the distance of a given demand from the true demand (i.e., the distance to the optimal solution). Figure 5-5 displays six plots. As before, each plot considers a given scenario. The difference from the previous plots is that Figure 5-5 displays along the y-axis the Euclidean distance between the current iterate and the true demand. All Am runs outperform all $A\phi$ runs for Plots 5-5b, 5-5c and 5-5e. Two of the three Am runs outperform all other runs for plot 5-5a. For Plots 5-5d and 5-5f, the best solution of Am outperforms all other runs.

Figure 5-6a illustrates the trajectory of a given Am run and a given $A\phi$ run considering random initial point 3 (i.e., one of the runs in Figures 5-4f and 5-5f). The contour lines display the SO objective function. The green (resp. red) line displays the trajectory of Am (resp. $A\phi$). This plot shows how Am immediately identifies a current iterate in the neighborhood of the true demand, while $A\phi$ requires more iterations to reach this neighborhood. Given the intricacy of the SO objective function, this immediate convergence is remarkable.

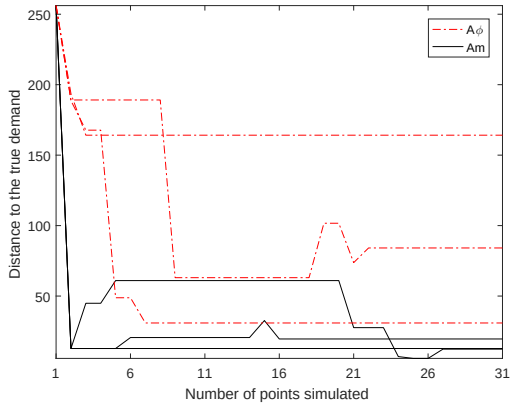
We now evaluate the quality of the solutions proposed by each algorithm for each of the six scenarios. In order to select a "best" solution for a given method and a given scenario, we consider the solutions proposed once the computational budget is



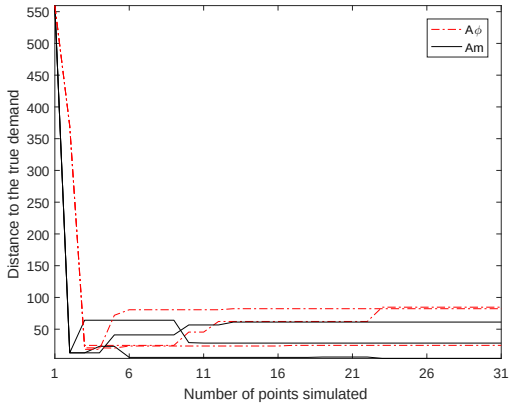
(a) Perturbed initial point 1



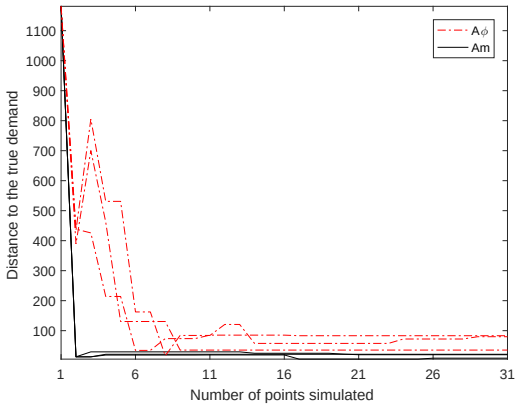
(b) Perturbed initial point 2



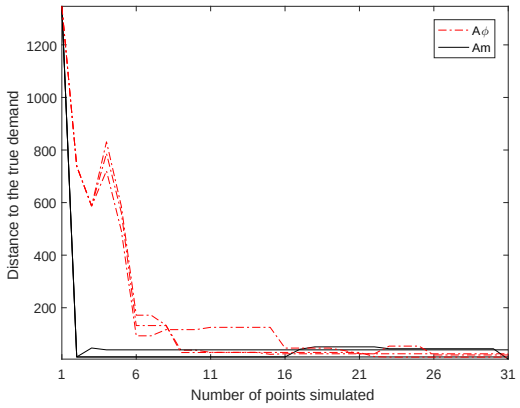
(c) Perturbed initial point 3



(d) Random initial point 1



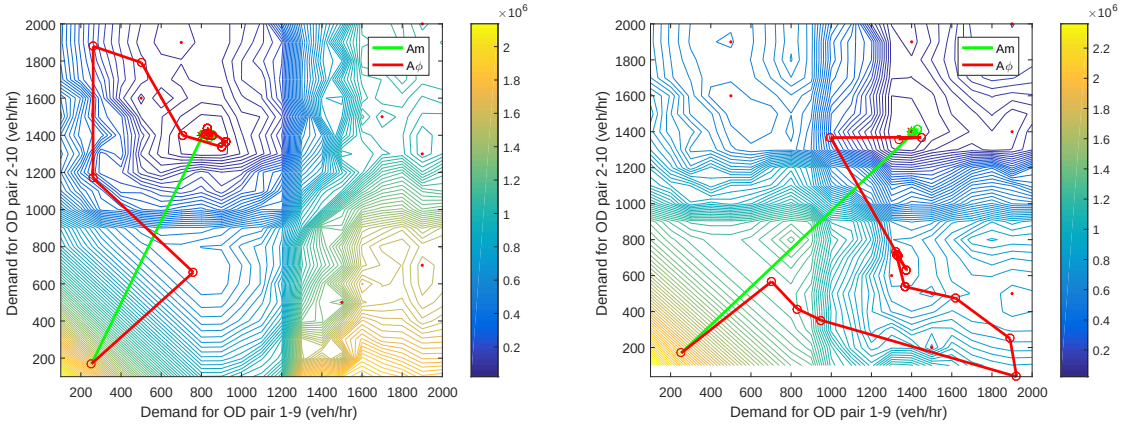
(e) Random initial point 2



(f) Random initial point 3

Figure 5-5: Distance to the true demand as a function of the total number of simulated points, for true demand $d^* = [800, 1400]$ (veh/hr)

depleted (there are 3 such solutions, one for each algorithm run). We then select as the "best" solution, among the 3 proposed solutions, that with the smallest objective



(a) $d^* = [800, 1400]$ (veh/hr) (b) $d^* = [1400, 1400]$ (veh/hr)

Figure 5-6: Trajectory of the solutions proposed by each algorithm when initialized with random initial point 3

function estimate. Figure 5-7 displays six black solid (resp. red dash-dotted) curves, which correspond to each of the six "best" solutions proposed by Am (resp. $A\phi$). For a given "best" point, we estimate its performance by running 10 simulation replications. Each curve is the cumulative distribution function (cdf) of the 10 estimates of the objective function. The x-axis represents objective function estimates. For a given x value, the corresponding y value of a curve gives the proportion of simulation replications (out of the 10) that have objective function values smaller than x. Hence, the more a cdf curve is shifted to the left, the higher the proportion of low objective function values, i.e., the better the performance of the corresponding solution. Three Am solutions outperform all $A\phi$ solutions. The other three Am solutions outperform all but one $A\phi$ solution. All Am curves are close to each other, while there is a higher variance in performance across the $A\phi$ solutions. This indicates a higher robustness of Am to both the quality of the initial points and to the stochasticity of the simulator.

We now analyze the second case with a symmetrical true OD demand (i.e., $[1400, 1400]$ veh/hr). The total demand is higher than that for the first case, leading to more links with high levels of congestion. Contour plots of the simulation-based objective function and its analytical approximation are presented in Figure 5-8. Both functions, simulation-based and analytical, are symmetrical. As before, the simulation-based function contains multiple (ten) local minima (marked with red dots), one of

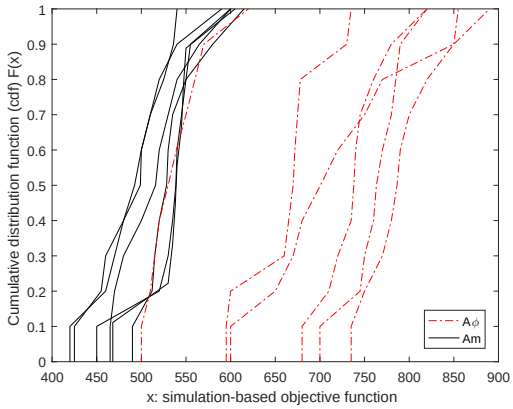
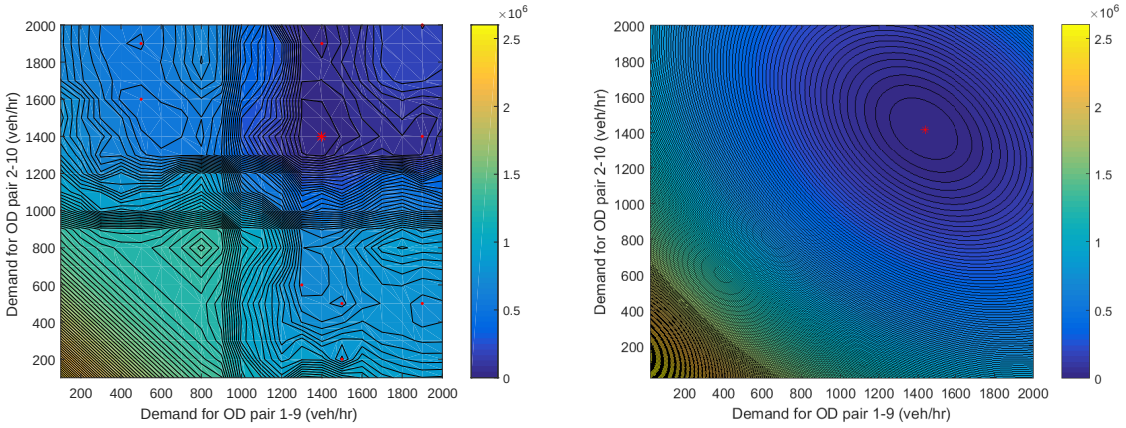


Figure 5-7: Distribution of the objective function for each of the best solutions proposed by each algorithm, for true demand $d^* = [800, 1400]$ (veh/hr)

which is a global minimum (marked with a red asterisk), while the analytical network model has a single global minimum, which closely approximates that of the simulator.

We proceed as before to analyze the performance of the algorithms. Each plot of Figure 5-9 considers one initial point. Plots 5-9a to 5-9c (resp. Plots 5-9d to 5-9f) are associated with scenarios initialized by perturbed (resp. uniformly distributed random) points. Similar conclusions as before hold: (i) at the second simulated point, *Am* yields a significant improvement in performance, which is due to the analytical network model; (ii) as iterations advance, *Am* continues to identify solutions with



(a) Simulation-based objective function

(b) Objective function approximation provided by the analytical network model

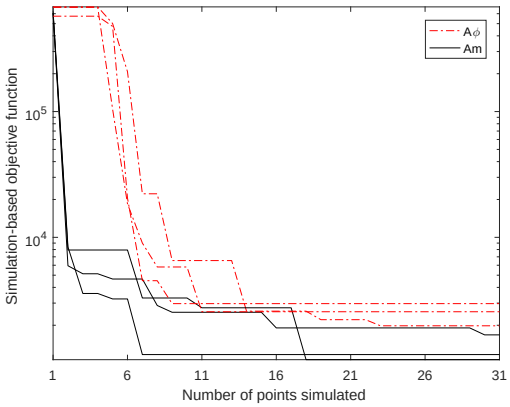
Figure 5-8: Objective function for true demand $d^* = [1400, 1400]$ (veh/hr)

good performance; and (iii) for most runs, Am outperforms $A\phi$ both across iterations and at the final iteration. Compared to the previous experiment, the performance of $A\phi$ appears here to be more sensitive to the quality of the initial points. More specifically, some runs of $A\phi$ (e.g., Plots 5-9e and 5-9f) have current iterate points with performance similar to that of the initial points until the last 5-10 points are simulated.

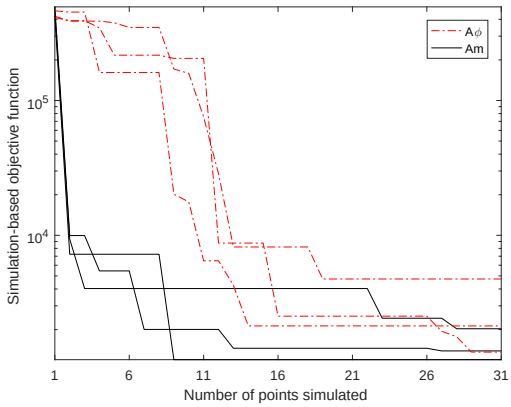
Figure 5-10 displays one plot per initial point. For each plot, it displays the distance to the true demand as a function of the number of simulated points. For all plots, the performance of Am is very similar across SO runs (i.e., for a given plot, the three Am curves are similar). For $A\phi$, there is higher variability both for SO runs of a given initial point (i.e., a given plot) and across initial points (i.e., across plots). These plots indicate that right after the first simulated point, Am identifies current iterates that are close to the true demand. This is consistent across initial points. On the other hand, the performance of $A\phi$ is more sensitive to the quality of the initial points. For 3 of the 6 scenarios (i.e., Plots 5-10b, 5-10e, and 5-10f), there are $A\phi$ runs that require a larger number of simulated points ($\sim$10-20) to identify points with significantly improved performance. Especially, one $A\phi$ run (Plot 5-10f), which upon depletion of the computational budget, yields a proposed solution that is still far from the true demand (with a distance of $\sim$800).

Figure 5-6b illustrates the trajectory of a given Am run and a given $A\phi$ run considering random initial point 3 (i.e., one of the runs in Figure 5-9f and 5-10f). The conclusions are the same as for the previous demand scenario: Am immediately identifies a current iterate in the neighborhood of the true demand, while $A\phi$ requires more iterations to do so.

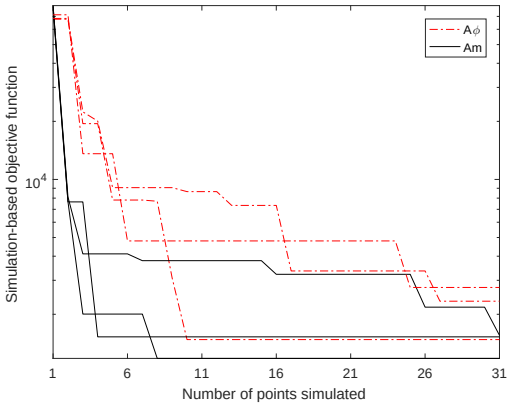
We proceed as before and select, for each scenario and each method, the "best" solution. Figure 5-11 compares the cdf's of the best solutions across both methods and scenarios. The conclusions are very similar to those of the asymmetrical true OD demand experiments: (i) three of the Am solutions outperform all $A\phi$ solutions; (ii) the remaining three Am solutions have performance similar to that of one $A\phi$ solution and they outperform the remaining 5 $A\phi$ solutions; and (iii) the Am solutions have



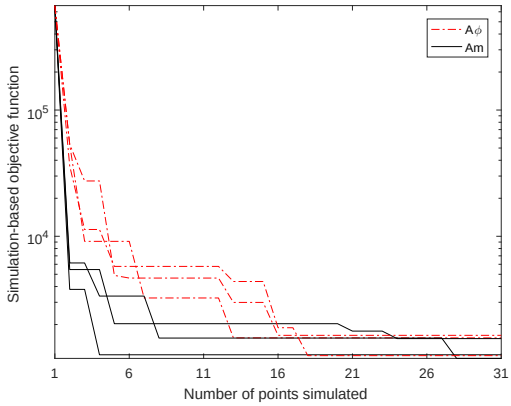
(a) Perturbed initial point 1



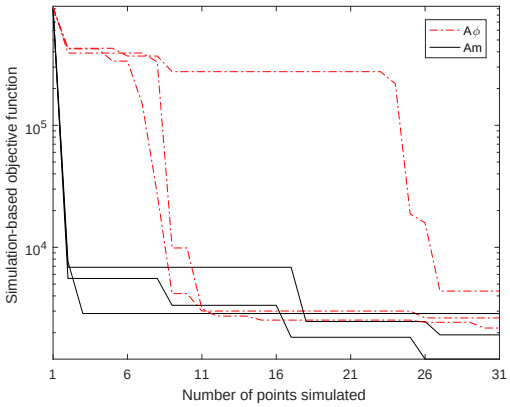
(b) Perturbed initial point 2



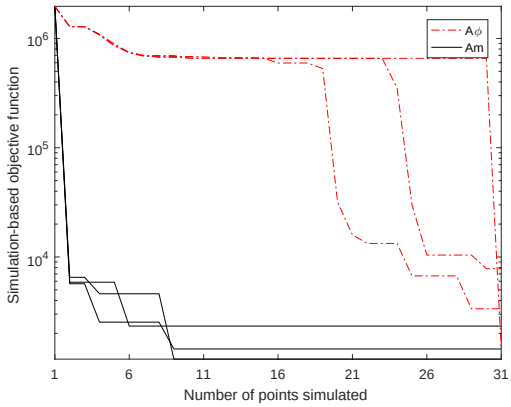
(c) Perturbed initial point 3



(d) Random initial point 1



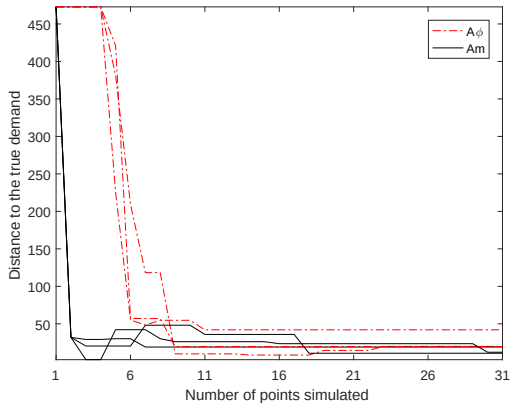
(e) Random initial point 2



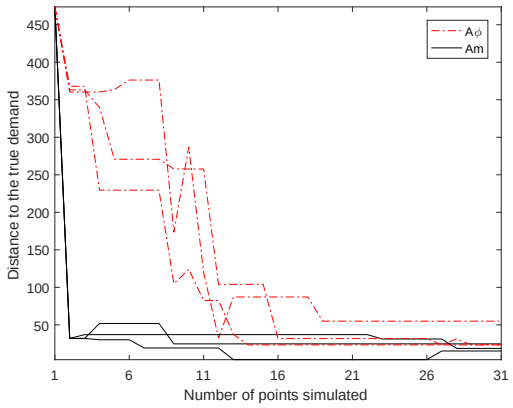
(f) Random initial point 3

Figure 5-9: Objective function estimate of the current iterate as a function of the total number of simulated points, for true demand $d^* = [1400, 1400]$ (veh/hr)

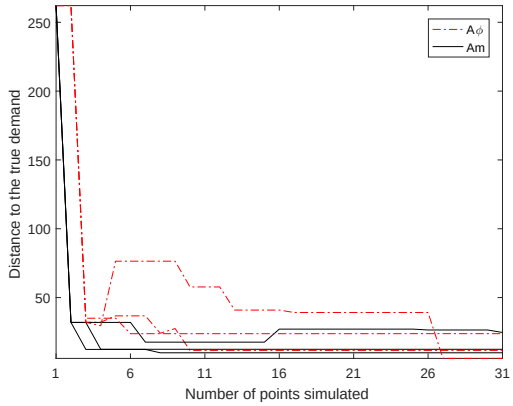
similar performance, while there is a high variance in performance across solutions for $A\phi$. The last point illustrates the robustness, provided by the analytical network



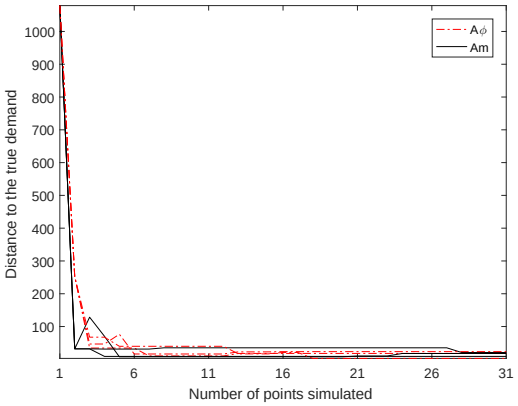
(a) Perturbed initial point 1



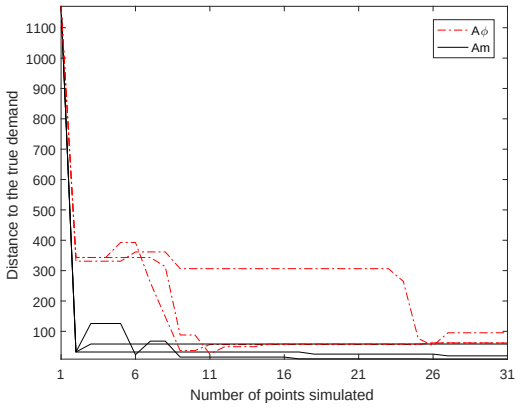
(b) Perturbed initial point 2



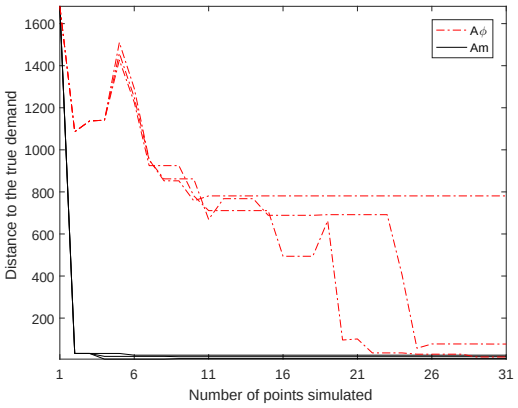
(c) Perturbed initial point 3



(d) Random initial point 1



(e) Random initial point 2



(f) Random initial point 3

Figure 5-10: Distance to the true demand as a function of the total number of simulated points, for true demand $d^* = [1400, 1400]$ (veh/hr)

model, of *Am* with regard to both the quality of the initial points and the simulators' stochasticity.

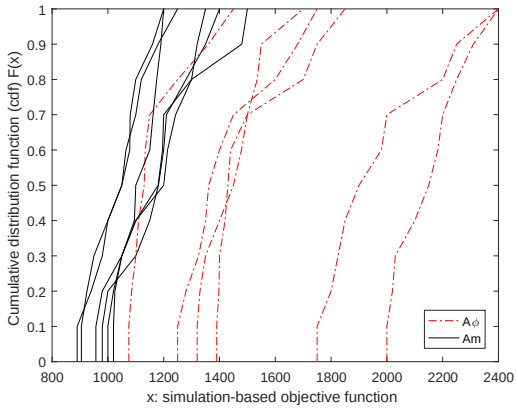


Figure 5-11: Distribution of the objective function for each of the best solutions proposed by each algorithm, for true demand $d^* = [1400, 1400]$ (veh/hr)

5.3.2 Berlin Metropolitan Network

We now study a high-dimensional OD demand calibration problem. We perform OD calibration for the Berlin (Germany) metropolitan area, which consists of two federal states: Berlin and Brandenburg. A map of the road network of this area is shown in Figure 3-15 of Chapter 3. The road network simulation model is displayed in Figure 3-14 of Chapter 3. The network is decomposed into a set of 557 traffic analysis zones (TAZ), according to the senate Department for the Environment, Transport and Climate Protection in Berlin, Germany, which is presented in Figure 5-12. They include 138 LORs (i.e., statistical planning region; in German, *Lebensweltlich Orientierte Räume*) within the city of Berlin and 419 municipalities in Brandenburg. The network consists of a total of 24,335 links, 11,345 nodes, and 2,585 OD pairs. Unlike the synthetic toy network, the link traffic counts are now field data obtained from a set of 346 links; the true OD demand is unknown; and the problem is underdetermined. Additionally, the network model is computationally costly to evaluate with the simulator. Hence, there is a pressing need to design calibration algorithms that can identify points with good performance within small computational budgets.

Figure 5-12: Traffic analysis zones

Experimental Design

The prior OD demand vector, $\tilde{d}$, is an existing matrix derived from census data. The experimental design is detailed in the second column of Table 5.1. The rest of the experiment design refers to Table 3.2. The demand bounds are the same as for the synthetic case study (row 1 of Table 5.1). The true OD demand is unknown (row 2). The weight factor for the prior OD matrix is set to $\delta = 0.01$ (row 3).

We consider 3 scenarios, each with different initial points, two of which are perturbed initial points. We use the same perturbation approach as for the toy network. The perturbation term is sampled in the interval $[-500, 500]$. The third point is randomly drawn from a uniform distribution in $[0, 2000]$. Experiments are carried out on a workstation with a 32-core Dual Intel Xeon Processor E5-2630 v3 and 512GB RAM.

Numerical Results

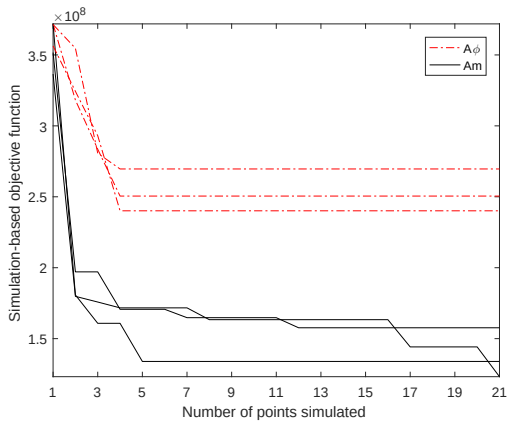
Figure 5-13 displays one plot for each scenario. Note that the y-axis is no longer logarithmic. For all runs of all three scenarios, Am outperforms $A\phi$ and does so across all iterations (i.e., all x-axis values). Additionally, for all runs and all initial

points, upon depletion of the computational budget, the solutions proposed by Am have significantly lower objective function estimates than those of $A\phi$. On average, the final solutions proposed by Am improve the objective function by 70% compared to those of $A\phi$. The performance of Am is similar both across SO runs for a given initial point (i.e., across curves of a given plot) and across initial points (i.e., across plots). This indicates the robustness of Am to both the quality of the initial points and to the simulator's stochasticity. In contrast, the runs of $A\phi$ vary in performance both across SO runs for a given initial point, and, more markedly, across initial points. In all plots, the significant improvement of Am at the second simulated point is remarkable. As discussed before, this is due entirely to the analytical network model. As the iterations advance, Am continues to identify points with further improved performance. For Figures 5-13b and 5-13c, $A\phi$ finds points with improved performance as the iterations advance. Nonetheless, for Figure 5-13a, for all three $A\phi$ runs, no further progress is made after 4 points are simulated.

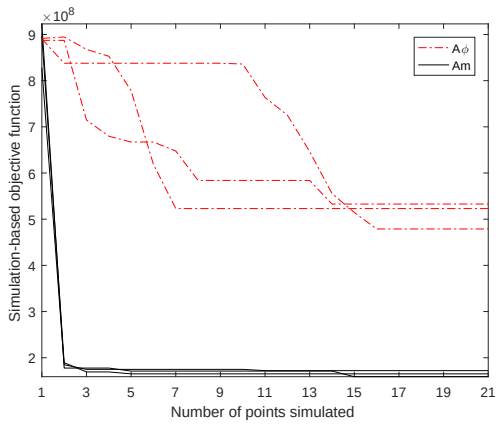
For each scenario, each SO run and each method, we consider the final proposed solution. This corresponds to the current iterate once the computational budget is depleted. For each proposed solution, we run 10 simulation replications. Figure 5-14 compares the cumulative distribution functions (cdf's) of each proposed solution. The solutions proposed by Am (resp. $A\phi$) are displayed as black solid (resp. red dash-dotted) curves. The figure also displays the cdf of the solution obtained by using only the analytical network model (i.e., no simulation-based optimization is performed). This cdf is displayed as a black solid line with crosses, and is denoted by *Analytical* in the plot legend. This solution corresponds to the point evaluated by Am at $x = 2$ for each plot of Figure 5-13. Figure 5-14 also displays the cdf of the initial point (blue dashed), as well as the cdf of the prior OD matrix (black dotted).

For all 3 plots of Figure 5-14, the following observations hold.

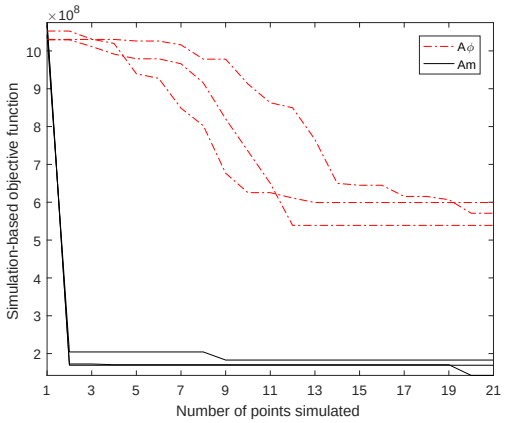
- All solutions of Am outperform all other solutions. They outperform the analytical solution, which indicates the added value of complementing the analytical information with the simulation-based information. They outperform $A\phi$, which indicates the added value of complementing the simulation-based information



(a) Perturbed initial point 1



(b) Perturbed initial point 2



(c) Random initial point 1

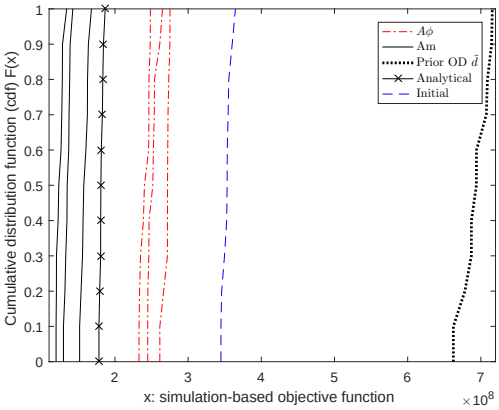
Figure 5-13: Objective function estimate of the current iterate as a function of the total number of simulated points.

with the analytical information.

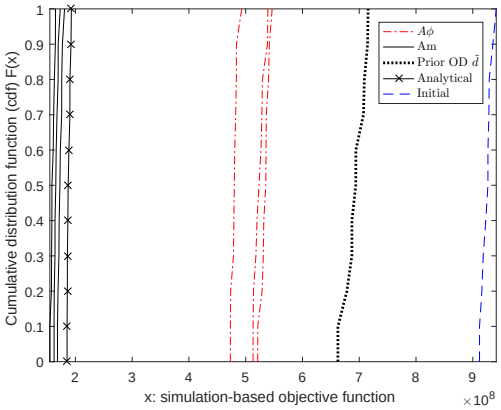
- The analytical solution outperforms all 3 solutions proposed by $A\phi$. This indicates the added value of the analytical information. Even though it is a very simple model, it contains essential structural information that leads to the identification of points with good performance.

- All three solutions of $A\phi$ outperform the initial point and the prior OD matrix.

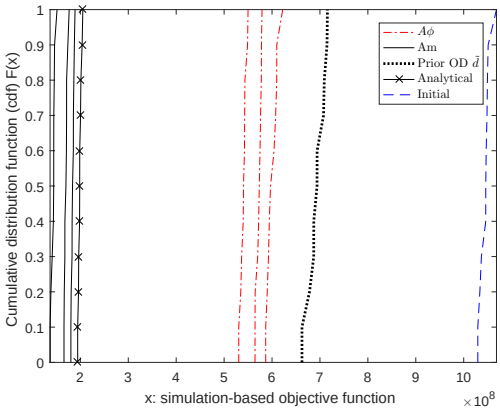
Regardless of the initial points (i.e., across all 3 plots), Am yields solutions with similarly good performance. On the other hand, $A\phi$ has similar performance for a given initial point, yet its performance varies significantly across initial points. This



(a) Perturbed initial point 1



(b) Perturbed initial point 2



(c) Random initial point 1

Figure 5-14: Distribution of the objective function for each of the best solutions proposed by each method

indicates the robustness of Am, provided by the analytical network model, to the quality of the initial points.

Table 5.2 considers for each initial point, the "best" solution by each method (i.e., smallest average objective function value). It displays the percentage improvement in the average objective function value compared to the value of the prior OD matrix, $\tilde{d}$. It emphasizes that the performance of $A\phi$ is sensitive to the initial points, with an improvement that varies from 22% to 65%. The analytical network model yields similar improvements in the order of 73%. Additionally, the 3 solutions proposed by the analytical network model are identical for all three initial points. The proposed method Am yields improvements that are similar across all initial points, which range

Table 5.2: Percentage improvement of the objective function of the "best" solution compared to the prior OD $\tilde{d}$ matrix

Initial point	$A\phi$	Analytical	Am
Perturbed point 1	65.4	73.1	82.3
Perturbed point 2	31.0	72.7	77.1
Random point 1	22.4	72.8	79.5

from 77% to 82%.

Let us now compare the performance of the methods in terms of their ability to replicate the traffic counts from field data. For each method, we consider all 9 runs displayed in the plots of Figure 5-14 and select the solution with best performance (i.e., for each cdf curve we compute its average value and select the cdf with the lowest value). Each plot of Figure 5-15 contains 346 points that represent each of the 346 sensors in the network. The plots compare the field traffic counts (x-axis) to those of: (i) the prior OD matrix (y-axis of the left-most plot); (ii) the best solution of $A\phi$ (y-axis of the middle plot); and (iii) the best solution of Am (y-axis of the right-most plot). In each plot, the diagonal line is included as a reference. The closer the points are to the diagonal, the better the fit to the field data. These plots indicate that the best solution of Am outperforms both the best solution of $A\phi$ and the prior OD matrix.

A commonly used metric to evaluate the calibration performance is the root mean

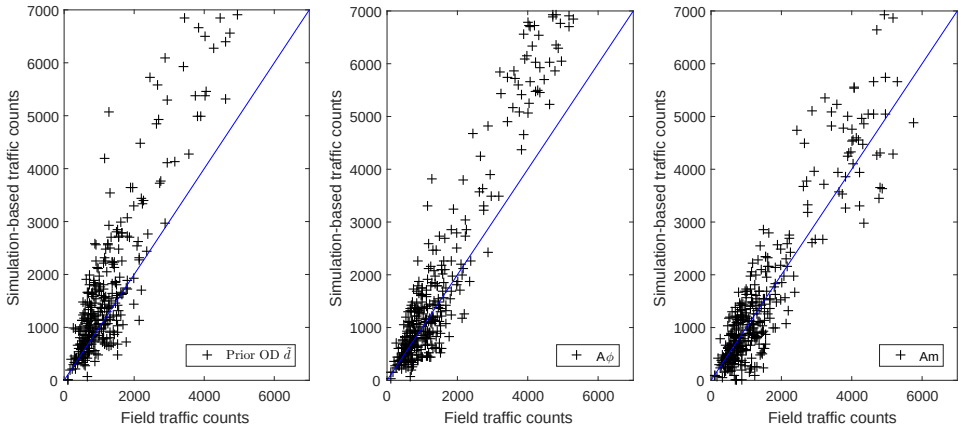


Figure 5-15: Comparison of field traffic counts and simulated traffic counts

square normalized error (RMSN) function, which is defined by:

$$RMSN = \frac{\sqrt{\frac{1}{N}\sum_{i\in\mathcal{I}}(y_i - \hat{E}[F_i])^2}}{\frac{1}{N}\sum_{i\in\mathcal{I}} y_i}, \tag{5.8}$$

where $N = \text{card}(\mathcal{I})$ and $\hat{E}[F_i]$ denotes the simulated estimate of the expected flow of link i for a given method. Small RMSN values are indicative of good fit of the simulated estimates to the field data. Table 5.3 lists RMSN statistics for the prior OD matrix (column 1), for the best solution of $A\phi$ (column 2), and for the best solution of Am (column 3). The best solution of Am improves the fit compared to the prior OD matrix by 58%, and improves the fit compared to the best solution of $A\phi$ by 29%.

Table 5.4 compares the computational runtimes of Am and $A\phi$. Note that these Berlin network runs are carried out on a server for which the CPU resources allocated to a given job vary over time and depend on what other jobs are running simultaneously. We provide these computation statistics to give a general idea of runtimes, nonetheless the differences in runtimes cannot be attributed solely to the algorithmic performance of Am and $A\phi$. The table indicates, for each of the 9 runs of each method, the total computational runtime per experiment (i.e., time to run one SO run until depletion of the computational budget). This total runtime includes the runtime for each metamodel optimization (i.e., solution of Problem (5.4)-(5.7)) and the total simulation runtime. Each row corresponds to the runs of a given initial point. Overall, Am reduces total runtime on average by 30% (64 hours versus 45 hours). For the first perturbed initial point (row 1), the computing times are similar: Am has an average computing time of 62.8 hours versus 59.9 hours for $A\phi$. For the other two initial points (rows 2 and 3), the runtimes for Am are significantly smaller than those for $A\phi$. For the second perturbed initial point (row 2), the average computing time for Am is less than half that of $A\phi$ (27.2 hours versus 58.1 hours), i.e.,

Table 5.3: Root mean square normalized error (RMSN) statistics

	Prior OD $\tilde{d}$	$A\phi$	Am
$RMSN$	0.952	0.561	0.397

Table 5.4: Total computational runtime (hr) per algorithmic run

Initial point	Am			$A\phi$		
Perturbed point 1	71.1	63.0	54.2	63.5	55.6	60.6
Perturbed point 2	31.9	25.0	24.6	44.2	70.0	60.2
Random point 1	43.8	40.5	50.6	72.2	77.0	73.8

there is a 53% reduction in average computing time. For the random initial point (row 3), the reduction is 40% (45 hours versus 74.3 hours).

The table indicates that even though each run consists of the same computational budget (which is defined as a maximum number of simulated points), the total runtime varies significantly across runs. This variation is due to both varying simulation runtimes and varying metamodel optimization runtimes. In particular, the simulation runtimes are a function of total demand and the level of congestion on individual links.

We can combine the results of Tables 5.2 and 5.4 to summarize the added value of using the information derived from the analytical network model. It allows the proposed method to improve performance by 49% over the benchmark method (based on Table 5.2 there is an average of 39.6% improvement for $A\phi$ compared to an average of 79.6% for Am) while also achieving an average reduction in computing time of 30%. The use of analytical structural information to address intricate SO problems, such as demand calibration, allows identifying solutions with significantly improved performance and does so significantly faster.

5.4 Conclusions and Discussion

This chapter presents a computationally efficient algorithm for OD demand calibration for stochastic, high-resolution, and large-scale network simulators. The proposed approach is based on a metamodel simulation-based optimization algorithm. The computational efficiency is achieved by using a metamodel that combines information from the simulator with that from an analytical network model. The analytical network model is differentiable, computationally efficient, scalable, and convex. It is formulated as a system of linear equations, the dimension of which scales linearly

with the number of links in the network and does not depend on other link attributes, such as link lengths. We formulate an analytical network model that is simple yet captures problem-specific structure. We show with a synthetic toy network its ability to approximate the intricate (e.g., non-convex) simulation-based objective function and to accurately locate the global optimal solutions.

The performance of the proposed calibration algorithm is evaluated with both a synthetic toy network and a large-scale metropolitan area network of Berlin, Germany. Its performance is benchmarked against a general-purpose algorithm that differs only in that the metamodel does not use information from the analytical network model. For both networks, the experiments indicate the significant added value of using such analytical structural information for the SO algorithm. For the Berlin network, compared to the benchmark method, the proposed method yields an average of 70% improvement in the quality of the solution, as measured by its objective function estimates, while simultaneously reducing the computational runtimes by an average of 30%. The analytical structural information is also shown to yield an algorithm that is robust to both the quality of the initial points and the stochasticity of the simulator. For all experiments, the proposed method was able to identify a solution with significantly improved performance at the first iteration of the algorithm.

Given the intricacy of simulators, as well as their high computational runtime costs, there is a pressing need to enable their use to efficiently address the types of intricate optimization problems faced by transportation stakeholders around the globe. A promising approach is to use the family of metamodel ideas, combining ideas from analytical traffic modeling and simulation-based traffic modeling. Ongoing work explores the use of metamodel ideas for online calibration. Of ongoing interest is also the combination of these metamodel ideas with disaggregate data, which is increasingly available, such as partial vehicle trajectories, to improve the calibration and validation of large-scale network simulation models.

Chapter 6

Enhanced Origin-Destination (OD) Demand Calibration Incorporating Emerging Data Sources

The contents of this chapter are available as: Zhang, C. and Osorio, C. (2018). Enhanced offline origin-destination (OD) demand calibration incorporating emerging data sources. *Technical report, Massachusetts Institute of Technology.*

6.1 Introduction

This chapter formulates origin-destination (OD) demand calibration combining several heterogeneous data sources as a simulation-based optimization (SO) problem of high dimensionality whose objective function depends on the output of a large-scale, stochastic, noisy traffic simulator that does not have an analytical closed-form expression; and it also introduces a computationally efficient metamodel SO solution algorithm that is able to take advantage of higher resolution emerging data sources.

This chapter considers a specific type of high-resolution data: intersection turning flows. They are the proportions of traffic flows turning from one link to downstream links at intersections. This type of traffic data can be obtained by various methods. For instance, they can be obtained from the existing detector infras-

tructure. Coifman and Krishnamurthy (2007) propose a methodology for deriving path travel times and turning fractions from dual loop detectors by re-identifying vehicles at downstream detector locations. However, the success of this method is built upon the premise that vehicles have distinct lengths. More reliable ways of collecting turning flow data require the use of advanced transportation technologies, e.g., AVI and AVL systems, for probe vehicle re-identification and tracking. Specifically, they can be obtained from roadside cameras, video-based sensors, RFID plate or tag readers, and GPS that capture the chained locations of probe vehicles. Intersection turning flows or other similar data types, such as turning fractions and route choice probabilities, have already been used to improve demand calibration in literature (e.g., Van Der Zijpp and De Romph, 1997; Mishalani *et al.*, 2002; Alibabai and Mahmassani, 2008). Mishalani *et al.* (2002) use link travel times and intersection turning flows processed from video cameras to estimate dynamic OD matrices for a 3-intersection arterial and show the intersection turning flow data significantly improves the quality of the estimated OD matrices. Similarly, Alibabai and Mahmassani (2008) show that the addition of intersection turning movements improves the reliability of the estimated OD demand. Intersection turning flows can be a promising data source for demand calibration to complement the conventional traffic count data. This chapter incorporates the high-resolution intersection turning flow data to formulate an OD demand calibration problem and uses metamodel SO techniques to address it in a computationally efficient way.

In this chapter, we assume that intersection turning flows, going from the subset of links with loop detectors to each of their downstream links, are available. Specifically, we assume that the conventional traffic data collected by loop detectors is complemented by the richer and more detailed intersection turning flow data. The problem formulation and solution procedure proposed in this chapter are flexible with respect to the other types of traffic data used. This sheds light on the development of enhanced calibration framework for large-scale stochastic traffic simulation models that leverages big data in the future.

This chapter enhances the metamodel SO demand calibration algorithm proposed

in Chapter 5, which is extended based on Chapters 3 and 4. Compared to Chapter 5, this chapter is enhanced in the following ways: (1) using a link-independent metamodel to approximate the objective function directly, which avoids the problem of overfitting; (2) using a fundamental diagram-based analytical network model with endogenous assignment, which is able to better capture traffic congestion in the network and provides the mapping of OD demand to traffic counts, intersection turning flows and (possibly) other traffic measurements; (3) leveraging an additional data source, i.e., the intersection turning flows, for a refined solution to the SO calibration problem. Results on a large-scale real-world metropolitan Berlin network show enhanced performance of the calibration algorithm.

This chapter is organized as follows. Section 6.2 mathematically formulates the offline OD demand calibration problem combining several heterogeneous data sources, explains the metamodel SO solution procedure, and presents a novel link-independent metamodel. Section 6.3 presents the experimental design and reports the numerical results. Specifically, we first validate the proposed algorithm on a synthetic toy network and then demonstrate its performance on a large-scale network in the metropolitan area of Berlin, Germany. Section 6.4 summarizes the main conclusions and implications. Appendix A details the enhanced calibration algorithm and Appendix C provides details regarding the implementation.

6.2 Methodology

6.2.1 Problem Formulation

The offline OD demand calibration using several heterogeneous data sources is formulated as an SO problem. We model the transportation network in terms of traffic analysis zones (TAZ). Each component of the vector of decision variables d is the expected number of trips made between a given origin zone and a given destination zone. We aim to obtain an OD demand matrix that yields simulation-based traffic measurements as close as possible to their observed values.

The following notation is used to formulate the SO demand calibration problem:

f	simulation-based objective function;
F_i	flow on link i as defined by the simulator;
G_{ij}	flow turning from link i to downstream link j at an intersection;
y_i	average flow on link i estimated from the field data;
g_{ij}	average flow turning from link i to downstream link j at an intersection;
d_z	expected travel demand for OD pair z;
$\tilde{d}_z$	prior travel demand for OD pair z;
$\mathcal{L}_1$	subset of links with field traffic counts;
$\mathcal{L}_2$	subset of link pairs with field intersection turning flows;
$\mathcal{Z}$	set of OD pairs;
δ_1	weight for prior OD demand information;
δ_2	weight for intersection turning flows;
u_1	endogenous simulation parameters;
u_2	exogenous simulation parameters.

With the notation, the SO problem can be expressed as:

$$\min_{d} \quad f(d) = \frac{1}{|\mathcal{L}_1|}\sum_{i\in\mathcal{L}_1}(y_i - E[F_i(d,u_1;u_2)])^2 + \delta_1\frac{1}{|\mathcal{Z}|}\sum_{z\in\mathcal{Z}}(\tilde{d}_z - d_z)^2 + \delta_2\frac{1}{|\mathcal{L}_2|}\sum_{(i,j)\in\mathcal{L}_2}(g_{ij} - E[G_{ij}(d,u_1;u_2)])^2 \tag{6.1}$$

$$d \geq 0, \tag{6.2}$$

where $|\cdot|$ represents the cardinality of a set. Endogenous (resp. exogenous) simulation parameters u_1 (resp. u_2) are such as link speeds, travel times, and queue lengths (resp. network topology, traffic management strategies) in the simulator.

The objective function (6.1) minimizes the goodness-of-fit measure, which is modeled as a weighted combination of the distances between: (1) the observed and the simulated traffic counts; (2) the observed and the simulated intersection turning flows; and (3) the decision variables and the prior estimate of the OD demand matrix (also

called seed matrix or target matrix). Additional weights are pre-determined and assigned to the second and the third terms based on the decision maker's relative preferences for, or confidence in, the specific data source. For example, the weight factor δ_1 is usually very small to avoid confining the search to the vicinity of the prior OD demand matrix, which is usually less reliable. Specifically, we set δ_1 to 0.01 and set δ_2 to 1 in this chapter.

The objective function (6.1) depends on the output of a large-scale, stochastic, noisy traffic simulator. Specifically, it depends on two simulation-based components $E[F_i]$ and $E[G_{ij}]$, which are derived from the intricate spatial-temporal traffic dynamics modeled by traffic simulators. They can be defined as follows:

$$E[F_i(d, u_1; u_2)] = \sum_{z \in \mathcal{Z}} d_z \sum_{r \in R_z} \Delta_{ri} P_z(r|d, u_1; u_2) \tag{6.3}$$

$$E[G_{ij}(d, u_1; u_2)] = \sum_{z \in \mathcal{Z}} d_z \sum_{r \in R_z} \Omega_{rij} P_z(r|d, u_1; u_2), \tag{6.4}$$

where R_z is the set of feasible routes connecting OD pair z. $P_z(r|d, u_1; u_2)$ represents the probability of choosing a route $r \in R_z$ for a trip-maker traveling between OD pair z. Binary variable Δ_{ri} (resp. Ω_{rij}) equals 1 if link i (resp. consecutive link pair (i, j)) is on route r and 0 otherwise.

Equations (6.3)-(6.4) map OD demand d to available traffic measurements, namely, link traffic counts and intersection turning flows. The route choice probability $P_z(r|d, u_1; u_2)$ is a function of link travel times, which in turn depend on link flows $E[F_i]$ and intersection turning flows $E[G_{ij}]$. In traffic simulators, Equations (6.3)-(6.4) are based on simulations of replanning (e.g., route choice, mode choice, and departure time choice) and network loading, which are recursively evaluated. Therefore, they are nonlinear functions that lack sound mathematical properties such as convexity, have no closed-form expression, and can only be evaluated by running simulations. Due to the stochasticity of simulation models, each simulation evaluation requires multiple simulation replications. For high-resolution traffic simulators, each simulation replication is computationally inefficient. This poses challenges primarily to the computational

efficiency of calibration, especially for simulation-intensive traditional "black-box" algorithms that require a large number of simulation evaluations.

In addition, Problem (6.1)-(6.2) is a high-dimensional optimization problem since the decision variable, OD demand vector d, is typically in the order of several thousands. Due to the scarcity of traffic data y_i and g_{ij}, the problem is under-determined and contains numerous local minima. The inclusion of the prior OD demand $\tilde{d}$ helps the algorithm to identify solutions that are consistent with the land-use and activity patterns of the area. Compared to the formulation in Chapter 5, the enhanced formulation includes the term (i.e., the third component in (6.1)) addressing the distance between the simulation-based and the observed turning flows, which complements the problem formulation with additional high-resolution information.

6.2.2 Metamodel Formulation

We use a metamodel SO algorithm for solving the OD demand calibration problem. Adapted from the methodology in Section 2.3, the metamodel SO algorithm is customized to solve demand calibration problems.

Next, we present the novel metamodel for Problem (6.1)-(6.2). The proposed metamodel is an analytical approximation of the simulation-based objective function (6.1). We approximate the objective function by using a link-independent metamodel. Compared to a link-dependent metamodel (e.g., Zhang *et al.*, 2017, 2016a; Zhang and Osorio, 2018) that approximates each link separately, the link-independent metamodel requires fewer model parameters to be fitted and thus avoids overfitting. In order to formulate the metamodel, we introduce additional notation as follows.

f_A analytical approximation of SO objective;

k iteration index;

m_k metamodel function at iteration k;

β_k parameter vector of metamodel m_k;

ψ_{ij} expected turning flow from link i to downstream link j;

$\tilde{p}_{zs}$ proportion of demand from OD pair z that takes a route between link pair s;

$\hat{d}_s$ expected travel demand between link pair s;

λ_i expected demand of link i;

k_i expected density of link i;

v_i expected speed of link i;

$\tilde{t}_i$ expected travel time of link i;

t_r expected travel time of route r;

$P(r)$ route choice probability of route r;

k_i^{jam} jam density of link i;

v_i^{max} maximum speed of link i;

q^{cap} link flow capacity;

ℓ_i length of link i;

θ travel time coefficient of the route choice model;

c scaling parameter common to all links;

$O(r)$ OD pair of route r;

$\tilde{v}$ endogenous analytical model variables;

$\tilde{q}$ exogenous analytical model parameters;

h macroscopic analytical network model;

$\mathcal{L}(r)$ set of links of route r;

$\mathcal{R}_1(i)$ set of routes that include link i;

$\mathcal{R}_2(ij)$ set of routes that include consecutive link pair (i, j);

$\mathcal{R}_3(s)$ set of routes for link pair s.

As defined earlier, d_z denotes the demand for OD zone pair z, which is the aggre-

gation of individual trips at the link level. Regarding one zone z, trips may come from (resp. go to) different trip generation (resp. attraction) links within the zone. Each pair of these links (i.e., trip generation or attraction links) is called an OD link pair and is denoted by s. The aggregate zonal demand d_z and the disaggregate link level demand $\hat{d}_s$ is related through an incidence matrix $\tilde{p}$, which is exogenously defined.

The subproblem optimization at iteration k of the SO algorithm is formulated as:

$$\min_{d} \quad m_k(d;\beta_k) = \beta_{k,0} f_A(d) + \phi(d;\beta_k) + \delta_1 \frac{1}{|\mathcal{Z}|} \sum_{z\in\mathcal{Z}} (\tilde{d}_z - d_z)^2 \tag{6.5}$$

$$f_A(d) = \frac{1}{|\mathcal{L}_1|} \sum_{i\in\mathcal{L}_1} (y_i - \lambda_i)^2 + \delta_2 \frac{1}{|\mathcal{L}_2|} \sum_{(i,j)\in\mathcal{L}_2} (g_{ij} - \psi_{ij})^2 \tag{6.6}$$

$$\phi(d;\beta_k) = \beta_{k,1} + \sum_{z\in\mathcal{Z}} \beta_{k,z+1} d_z \tag{6.7}$$

$$h(d,\tilde{v};\tilde{q}) = 0 \tag{6.8}$$

$$d \geq 0. \tag{6.9}$$

The difference between Problem (6.1)-(6.2) and the Subproblem (6.5)-(6.9) is that the latter replaces all simulation-based components with counterparts derived from the analytical network model (Constraint (6.8)), which are firstly scaled by a factor $\beta_{k,0}$ and are then complemented by a linear polynomial of the decision variables, d, modeled by Equation (6.7). The subproblem optimization is the problem solved in Step 2a in Figure 2-2.

The analytical network model (6.8), which is formulated as a system of nonlinear equations, is presented as follows:

$$\hat{d}_s = \tilde{p}_{zs} d_z \tag{6.10}$$

$$\lambda_i = \sum_{r \in \mathcal{R}_1(i)} P(r) \hat{d}_{O(r)} \tag{6.11}$$

$$\psi_{ij} = \sum_{r \in \mathcal{R}_2(ij)} P(r) \hat{d}_{O(r)} \tag{6.12}$$

$$P(r) = \frac{e^{\theta t_r}}{\sum_{j \in \mathcal{R}_3(O(r))} e^{\theta t_j}} \tag{6.13}$$

$$t_r = \sum_{i \in \mathcal{L}(r)} \tilde{t}_i \tag{6.14}$$

$$\tilde{t}_i = \frac{\ell_i}{v_i} \tag{6.15}$$

$$v_i = v_i^{max} \left(1 - \frac{k_i}{k_i^{jam}}\right) \tag{6.16}$$

$$k_i = c \frac{k_i^{jam}}{q^{cap}} \lambda_i. \tag{6.17}$$

The analytical network model is based on the model of Osorio (2017). Firstly, Equation (6.10) disaggregates demand between traffic analysis zones to that between specific link pairs. Equation (6.11) defines the expected demand through link i, λ_i, as the sum of the expected route demand over the set of routes that contains link i. Similarly, Equation (6.12) defines the expected demand over consecutive link pair (i, j). The route demand is the summation of demand from all OD pairs that traverse the route. The demand for an OD pair is mapped to a route by multiplying the demand by the corresponding probability of choosing that route, $P(r)$. The route choice model is defined as a multinomial logit (MNL) model in Equation (6.13). The route choice is solely based on the route travel time, t_r, which is modeled in Equation (6.14) as the summation of the travel time of links along the route. Equation (6.15) defines the travel time of link i, $\tilde{t}_i$, as the link length divided by the expected speed, which is approximated by a simple linear fundamental diagram, as shown in Equation (6.16). Equation (6.17) assumes a linear relationship between the expected link density and the expected link demand where parameter c is chosen based on insights from prior

experience. In this chapter, we set c to $\frac{1}{6}$.

Compared to the original model of Osorio (2017), the proposed model differs in the following ways. Equation (6.10) is included for converting demand at the aggregate zonal level to that between a specific pair of links. This enables the model to be applicable to networks whose demand are represented in terms of zones. Equation (6.12) is new. It approximates the traffic measurements corresponding to the new data source: intersection turning flows. This is how we derive the analytical approximation for the more detailed turning flow data. Equation (6.16), modeling the relationship between the expected speed and the expected density, is further simplified from the original nonlinear function to a linear one. The aforementioned changes enable the analytical model to be capable of deriving other types of traffic measurements, to be able to handle aggregate demand at zonal level, and to further improve the computational tractability of the model.

6.3 Case Studies

6.3.1 Experimental Design

In order to comprehensively illustrate the methodology proposed in this chapter, we first validate the algorithms on a synthetic toy network, and then apply them to a large-scale network in the metropolitan area of Berlin, Germany.

Network Topologies

We consider two road transportation networks: a toy network and the real Berlin metropolitan network.

The topology of the synthetic network is shown in Figure 6-1. The network contains a total of 11 links and 11 nodes. All links are single-lane uni-directional roads. Four OD pairs, namely, 1-4, 7-11, 9-4, and 9-11, are considered. For demand between OD pair 1-4, there are two alternative routes: 1-2-3-4 and 1-2-5-6-3-4, respectively. For demand between the rest of OD pairs, there is a single route available. Hypo-

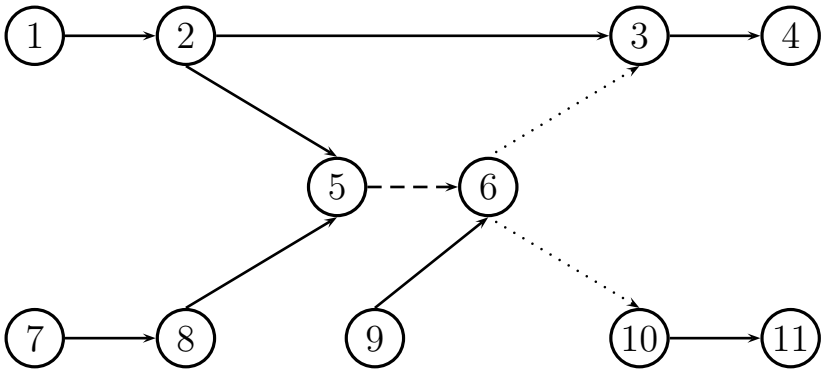


Figure 6-1: Synthetic toy network

thetically, a single loop detector is installed on link 5-6 (dashed) to collect traffic counts and two pairs of AVI sensors are installed on the following links: 5-6 (dashed), 6-3 (dotted), and 6-10 (dotted), to get turning flows at this intersection. In other words, the traffic flows turning from link 5-6 to its downstream links 6-3 and 6-10 are available. Unlike traffic counts, intersection turning flows provide additional information regarding travelers' route choice behavior. For example, turning flows reflect the route choice probabilities at intersections. We estimate the demand for all four OD pairs from traffic counts and intersection turning flows for a one-hour time interval. Additional link attributes are detailed in Table C.4 of Appendix C.

The large-scale transportation network in the metropolitan area of Berlin includes both the city of Berlin and the federal state of Brandenburg, Germany. A map of the road transportation network of this area is shown in Figure 3-15 of Chapter 3. The corresponding network implemented in the traffic simulator is displayed in Figure 3-14 of Chapter 3. The network is decomposed into a set of 557 traffic analysis zones (TAZ) which are presented in Figure 5-12 of Chapter 5. We focus on a single time interval: the morning peak hour, 8-9 AM. In this chapter, we assume, AVI sensors are installed at all count stations and their downstream links so that intersection turning flows from all count stations to their successors are recorded. In order to have controlled experiments, traffic counts and intersection turning flows are all synthesized via simulation runs.

Algorithms

The primary objectives of the case studies are to:

a. Highlight the added value of the analytical network model;

b. Quantify the added value of using intersection turning flows;

c. Demonstrate the flexibility, effectiveness, and efficiency of the metamodel SO calibration algorithm in handling multiple data sources.

In accordance with the objectives, we primarily conduct comparative analyses by considering the proposed algorithm and three other algorithms as listed in Table 6.1. The proposed algorithm is denoted by *Am-new*, which uses the analytical network model and incorporates richer data sources, namely, traffic counts and intersection turning flows. We compare it to other algorithms either do not use the analytical network model or do not incorporate intersection turning flows. Algorithms titled *new* are those that incorporate the more detailed data source while algorithms titled *old* only use the conventional traffic count data (i.e., without the last term in the objective function (6.1)). Algorithms titled *Am* use a metamodel that is dependent on the analytical network model, however, algorithms titled $A\phi$ use only a general-purpose metamodel without exploiting the problem-specific structure (i.e., they are "black-box" algorithms with $\beta_{k,0}$ set to zero in (6.5)). In addition, we also consider algorithms *Analy-new*/*Analy-old*, which use only the analytical network model for calibration with/without using intersection turning flows.

Pair-wise comparative analyses are conducted by comparing two algorithms that differ only in one factor. Through analysis of algorithm pairs {*Am-new*, $A\phi$*-new*}

Table 6.1: Calibration algorithms

	Traffic counts	Intersection turning flows	Analytical network model
Am-new	✓	✓	✓
$A\phi$*-new*	✓	✓	
Am-old	✓		✓
$A\phi$*-old*	✓		

and {*Am-old*, *Aϕ-old*}, the performance regarding Objective **a** is studied. Similarly, Objectives **b** and **c** are assessed by comparing algorithm pairs {*Am-new*, *Am-old*} and {*Aϕ-new*, *Aϕ-old*}.

Settings

The case studies are configured as displayed in Table 6.2. The rest of the experiment design are shown in Table 3.2. The demand scenario for the toy network case study is associated with various congestion levels across links. Specifically, it leads to a ratio of expected demand to expected supply that is greater than 1 on links 5-6, 9-6, 6-3, 3-4, and 6-10. The configuration for both networks as shown in Table 6.2 is consistent except for the computational budget, which is reduced for the Berlin network due to the added computational cost of simulating large-scale networks.

For the toy network, the prior OD demand matrix (i.e., seed matrix) is obtained by perturbing each entry of the true OD demand matrix with an error uniformly distributed in the range of $\{-30\%, 30\%\}$ of its maximal entry of the true OD matrix. For the Berlin network, traffic counts and intersection turning flows are first obtained by running simulations with the original demand specified by the simulator. The seed matrix is then obtained by applying the same procedure as described for the toy network. The seed matrix serves as an initial guess of the OD demand matrix, which is then updated using information from traffic counts and intersection turning flows.

Initial points are the demand vectors that are used to start algorithms. We vary initial points for a given algorithm to test its sensitivity to the initial conditions. For both case studies, two types of initial points are generated beforehand. The first type of initial points, which is referred to as *perturbed* initial points, is obtained through

Table 6.2: Experimental settings

	Synthetic toy network	Berlin network
Bounds for d values (veh/hr)	$[0, +\infty)$	$[0, +\infty)$
True d values (veh/hr), d^*	$[1500, 1000, 1000, 1000]$	N.A.
Weight factor of the prior OD matrix, δ_1	0.01	0.01
Weight factor of turning flows, δ_2	1	1

further perturbing the seed matrix by an additional 30% of the maximal entry of the seed matrix. The second type of initial points are generated randomly by sampling each entry of the OD demand vector independently from a uniform distribution in the interval $[0, 2000]$ (veh/hr). We refer to these points as *random* initial points.

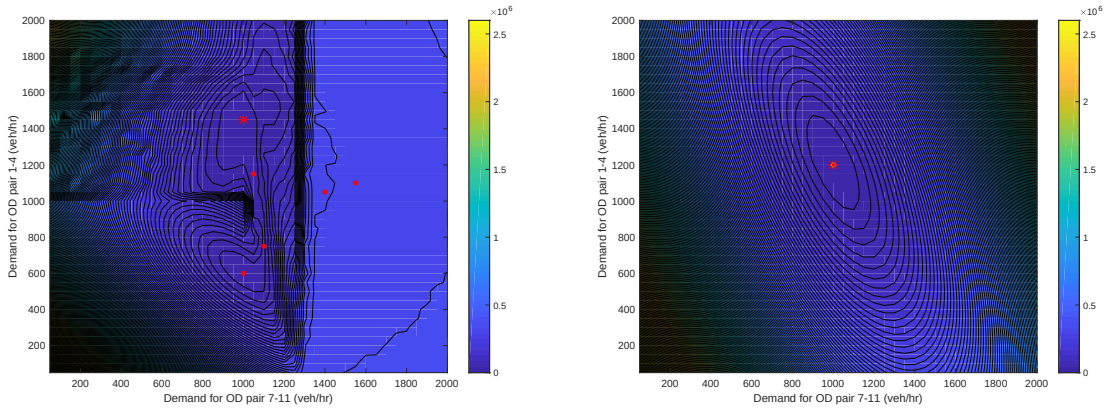
We call an experiment with a given initial point a scenario. For each scenario, we run each algorithm three times. We do this in order to account for the stochasticity of traffic simulators. The traffic simulation model used is MATSim (Multi-Agent Transport Simulation; Horni *et al.*, 2016), which is activity-based and agent-based. The toy experiments are carried out on a standard laptop with a 4-core Intel i7-3740QM processor, and the Berlin experiments are carried out on a workstation with a 32-core Dual Intel Xeon Processor E5-2630 v3 and 512GB RAM.

6.3.2 Results and Discussion

This section presents the numerical results of the case studies and analyzes the algorithmic performance in terms of accuracy, efficiency, and robustness, etc.

Toy Network

One challenge for demand calibration is the intricacy of the objective function which exhibits numerous local minima. In order to illustrate the objective function visually, we fix the demand for OD pairs 9-4 and 9-11 to their true values and only vary the demand for OD pairs 1-4 and 7-11. We first sample equally spaced and uniformly distributed points in the range of $[0, 2000]$ veh/hr with increments of 50 along each of the two demand dimensions. For a given demand, a point estimate of the objective function is obtained as the average of the estimates from 10 simulation replications. A contour plot is then constructed based on a set of point estimates and is displayed in Figure 6-2a. We mark the global minimum with a red asterisk and mark local minima with red dots. Figure 6-2a has a global minimum at $[1450, 1000]$ veh/hr, which is close to the true demand for the two varying OD pairs, $[1500, 1000]$ veh/hr. Compared to the objective function without using the more detailed data (i.e., $\delta_2 = 0$



(a) Simulation-based objective function (b) Analytical approximation to (a)

Figure 6-2: Contour plot of objective functions for true demand $[1500, 1000, 1000, 1000]$ (veh/hr)

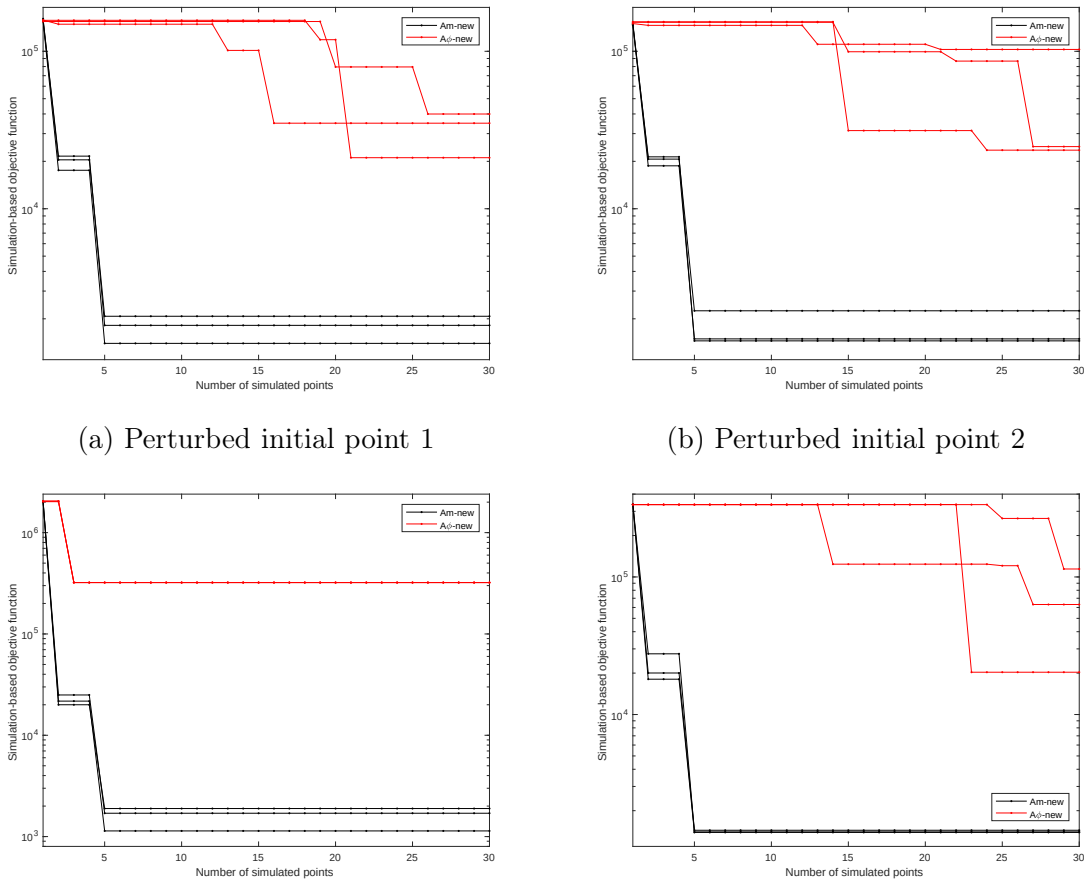
in the objective function (6.1)), the total number of minima is reduced from 25 to 6 by taking advantage of the more detailed data source. Therefore, the inclusion of intersection turning flows adds additional field data that constrains the set of OD demands that are consistent with the field measurements. Given that OD calibration problems are under-determined problems, this is an important advantage of using such new or higher resolution data sources.

The main premise of the proposed metamodel SO methodology is that computational efficiency can be improved by taking advantage of the problem-specific analytical information. We first investigate this by comparing the simulation-based objective function to its analytical counterpart. The latter differs from the former by using traffic measurements derived from the analytical network model instead of evaluating the simulation model. The analytical approximation of Figure 6-2a is displayed in Figure 6-2b. Similarly, we mark the global minimum with a red asterisk. Figure 6-2b has the global minimum at $[1200, 1000]$ veh/hr. Comparing the contour plots, the analytical objective function approximates the simulation-based objective function in terms of the approximate shape and the global minimum. This indicates that the analytical network model is able to capture problem-specific structural information of the simulator. To demonstrate the added value of the structural information provided by the analytical network model, we first compare algorithms *Am-new* to *Aϕ-new*. The

latter differs from the former only in terms of not embedding the analytical network model.

In Figure 6-3, we show the simulation-based objective function estimate of the current iterate (i.e., point with the best simulated performance) as a function of the number of simulated points. Note that the y-axis is displayed on a logarithmic scale. Algorithm *Am-new* (resp. *Aϕ-new*) is denoted by solid black (resp. red) curves. Each subplot represents one scenario with a different initial point. For each algorithm, three experiments are conducted and represented separately in each subplot. The x-axis displays the computational budget consumed so far as the number of simulated points, and the y-axis represents the corresponding simulation-based objective function of the current iterate.

Overall, the black curves achieve much smaller objective function values than the red ones for all scenarios. The curves corresponding to *Am-new* significantly reduce the objective function when only two points are simulated, and continuously achieve reductions in the objective function until the fifth point is simulated. In fact, the second simulated point is obtained by solving the calibration problem using only the analytical network model (i.e., $\beta_{k,0} = 1$ and $\beta_{k,j} = 0\ \forall j > 0$ in Subproblem (6.5)-(6.9)). Specifically, *Am-new* directs any given initial point to the minimum of the analytical objective function in Figure 6-2b, which closely approximates the global minimum of the simulation-based objective function in Figure 6-2a and thus corresponds to a much smaller objective function than the initial point. We then calculate the average objective function value for all final solutions achieved by a given method for all scenarios and all runs, which is 1,620 for *Am-new* and 117,456 for *Aϕ-new*. As a reference, the average objective function value across all initial points is 669,386. In contrast to the initial points, *Am-new* (resp. *Aϕ-new*) achieves an average reduction in the objective function by 99.8% (resp. 82.5%). Compared to *Aϕ-new*, *Am-new* further reduces the objective function by 98.6% on average. In summary, *Am-new* outperforms *Aϕ-new* by consistently yielding high-quality solutions (i.e., corresponding to smaller objective function) across all iterations and obtaining solutions with better performance when the computational budget is depleted, regardless of the



(a) Perturbed initial point 1

(b) Perturbed initial point 2

(c) Random initial point 1

(d) Random initial point 2

Figure 6-3: Simulation-based objective function estimate of the current iterate as a function of the total number of simulated points

quality of the initial points.

An important metric for evaluating the quality of calibration is the ability to replicate the field traffic measurements. Figure 6-4 displays for all scenarios the comparison of the "true" traffic measurements, i.e., the traffic count on link 5-6, and the turning flows from 5-6 to 6-3 and 6-10, and the simulated ones for a given algorithm. Three algorithms, namely, *Aϕ-new*, *Am-new*, and *Am-old*, are compared to the initial solution, denoted by *Init*. Each subplot corresponds to one algorithm. Traffic counts are marked with crosses and turning flows are marked with circles. The traffic measurements corresponding to different scenarios are shown in different colors: red for the perturbed initial point 1, yellow for the perturbed initial point 2,

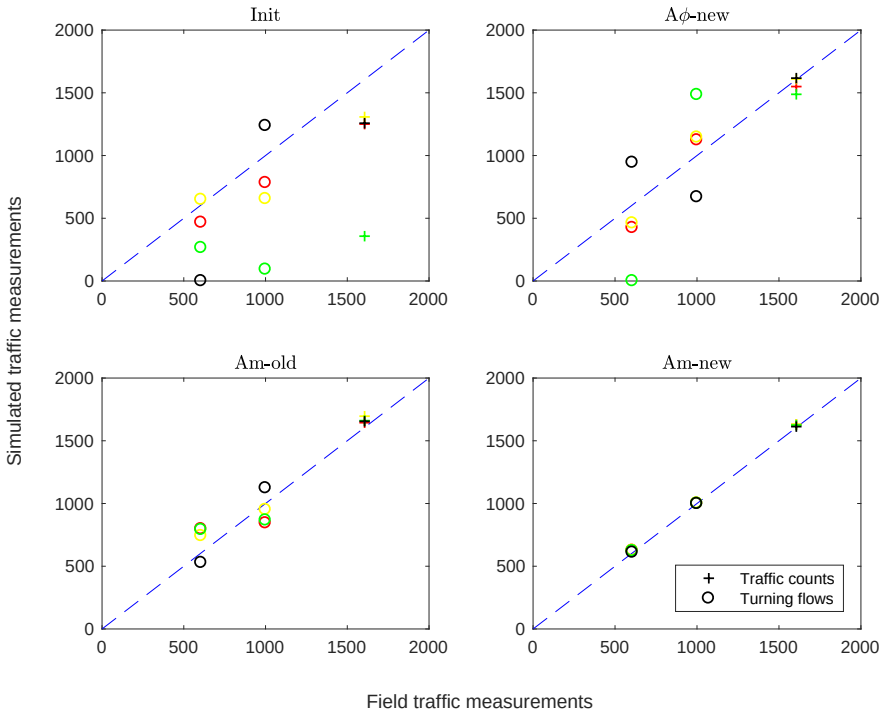


Figure 6-4: Comparison of the "true" traffic measurements and the simulated traffic measurements for the algorithmic solutions

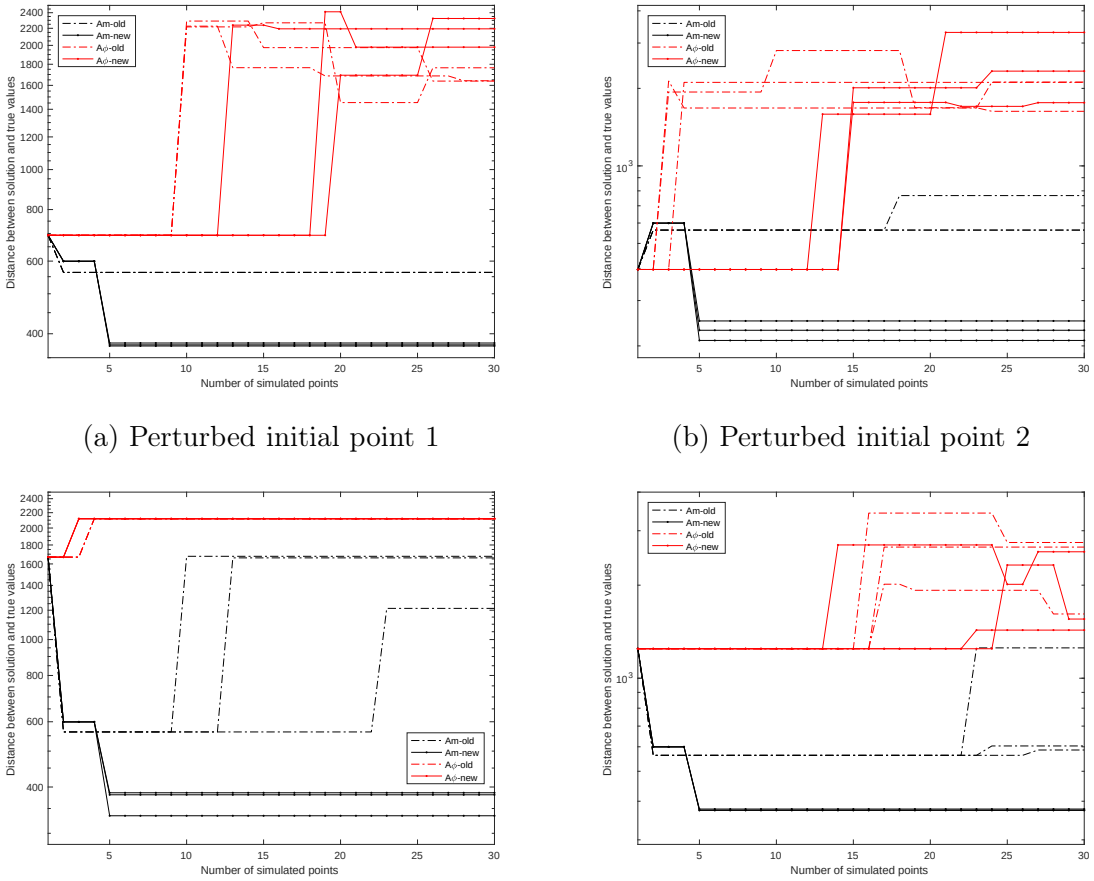
green for the random initial point 1, and black for the random initial point 2. For all subplots, the x-axis represents the "true" traffic measurements, the y-axis represents the simulated traffic measurements for an algorithmic solution, and a dashed blue diagonal line is displayed as a reference. The closer the data points are to the dashed blue line, the better the simulated traffic measurements replicate the "true" ones.

The subplot representing *Aϕ-new* closely approximates the traffic count but do not approximate well the turning flows, especially when the random initial points are supplied. This indicates that *Aϕ-new* is not robust to the initial conditions. Compared to *Aϕ-new*, *Am-old* provides a better approximation to both traffic measurements. *Am-new* further outperforms *Am-old* by providing a good approximation to the traffic count and a much better approximation to the turning flows. Since the only difference between *Am-new* and *Am-old* is the more detailed data, the added value of embedding the more detailed data in the problem formulation is highlighted in the presence of the embedded analytical network model. In conclusion, algorithms

titled *Am* outperform all others by consistently approximating well the "true" traffic measurements for all initial points, and *Am-new* further outperforms *Am-old* in terms of its better approximation to the turning flows by taking advantage of the more detailed data source. A quantitative analysis of the fit to field data in terms of the root mean square normalized (RMSN) errors for the toy network experiments is available in Tables B.3 and B.4.

Although the objective function is an important metric for evaluating the quality of calibration, a small objective function only indicates that the simulated traffic measurements fit the field data well but may not necessarily entail the accuracy of calibration, especially when the problem is under-determined (i.e., there are multiple local minima). The most effective way of evaluation is to compare the calibrated demand to the "true" demand directly, which is inapplicable in practice since the "true" demand is unknown. However, this is possible for controlled experiments such as the toy network experiments, where the "true" demand is known since all traffic measurements are synthesized. We calculate the Euclidean distance between the current iterate and the "true" demand for each simulated point.

Figure 6-5 displays the distance between the current iterate and the "true" demand as a function of the number of simulated points. Each subplot represents the scenario with a different initial point. For each scenario, solutions from all four algorithms are compared. For all scenarios except for Figure 6-5b, *Am-new* decreases the distance monotonically as more points are simulated. In Figure 6-5b, *Am-new* increases the distance a bit from iterations 1 to 4 but still achieves a reduction in the distance eventually. Overall, compared to *Am-new* the performance of *Am-old* deteriorates as more points are simulated in scenarios represented by Figures 6-5b and 6-5d. For Figure 6-5c, the final solutions for algorithm titled $A\phi$ are very different although their corresponding objective functions are very close. We then compare in terms of the average distance from the final solution to the "true" demand across all scenarios and all runs for a given method, which is, respectively, 2,023 for $A\phi$-*old*, 2,150 for $A\phi$-*new*, 882 for *Am-old*, and 337 for *Am-new*. This average distance calculated across all initial points is 1,001. Compared to the initial points, *Am-old* and *Am-new* achieves



(a) Perturbed initial point 1

(b) Perturbed initial point 2

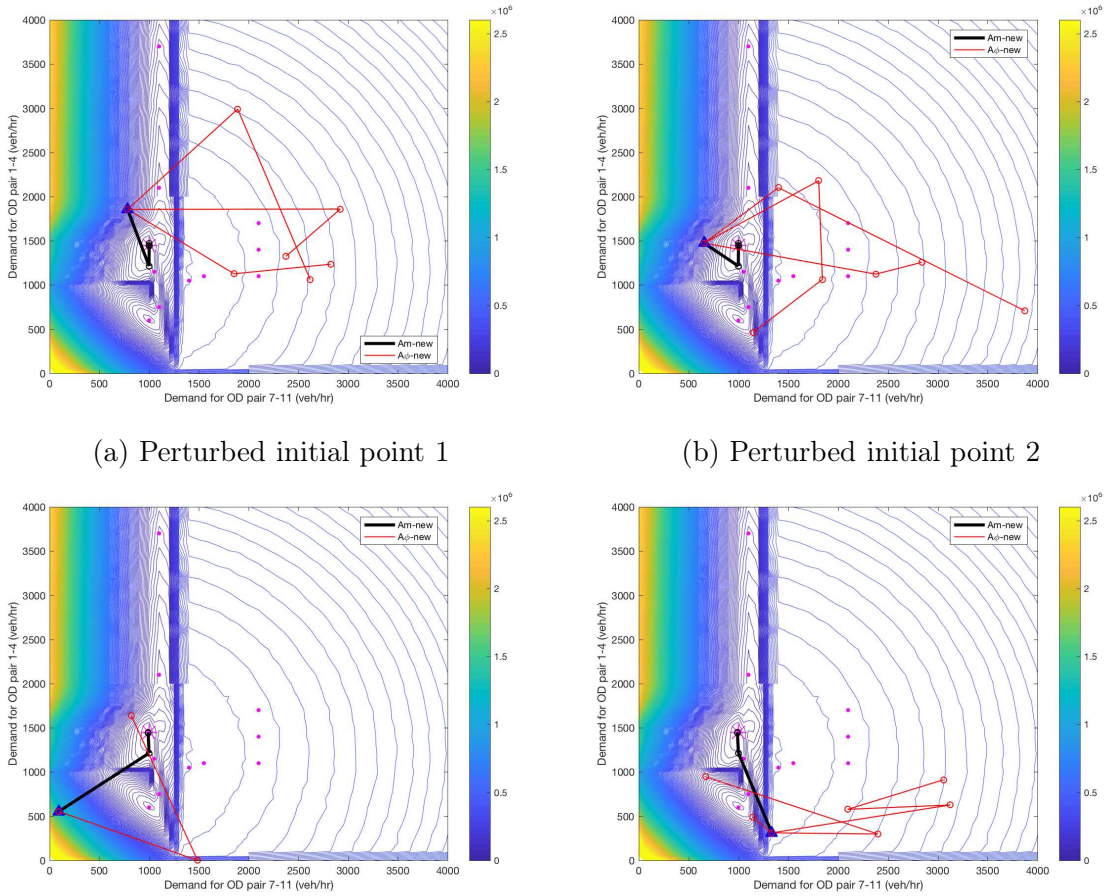
(c) Random initial point 1

(d) Random initial point 2

Figure 6-5: Distance between the current iterate and the "true" demand as a function of the total number of simulated points

an average reduction in distance by 11.9% and 66.3%, respectively. In contrast to *Am-old*, *Am-new* further reduces the distance by 61.8%. However, both *Aϕ-old* and *Aϕ-new* end up with more than doubled distance than the initial points. To sum up, the solutions derived by *Am-new* outperform those of *Am-old*, which reflects the added value of the more detailed data source, and both outperform algorithms titled *Aϕ*, which reveals the added value of the embedded analytical network model.

We illustrate the trajectory of the solution of a given algorithmic run for each initial point over the contour plot of the simulation-based objective function in Figure 6-6. Different from Figure 6-2a, we display an unfilled contour plot of the simulation-based objective function, which is extended to show each axis in the range



(a) Perturbed initial point 1

(b) Perturbed initial point 2

(c) Random initial point 1

(d) Random initial point 2

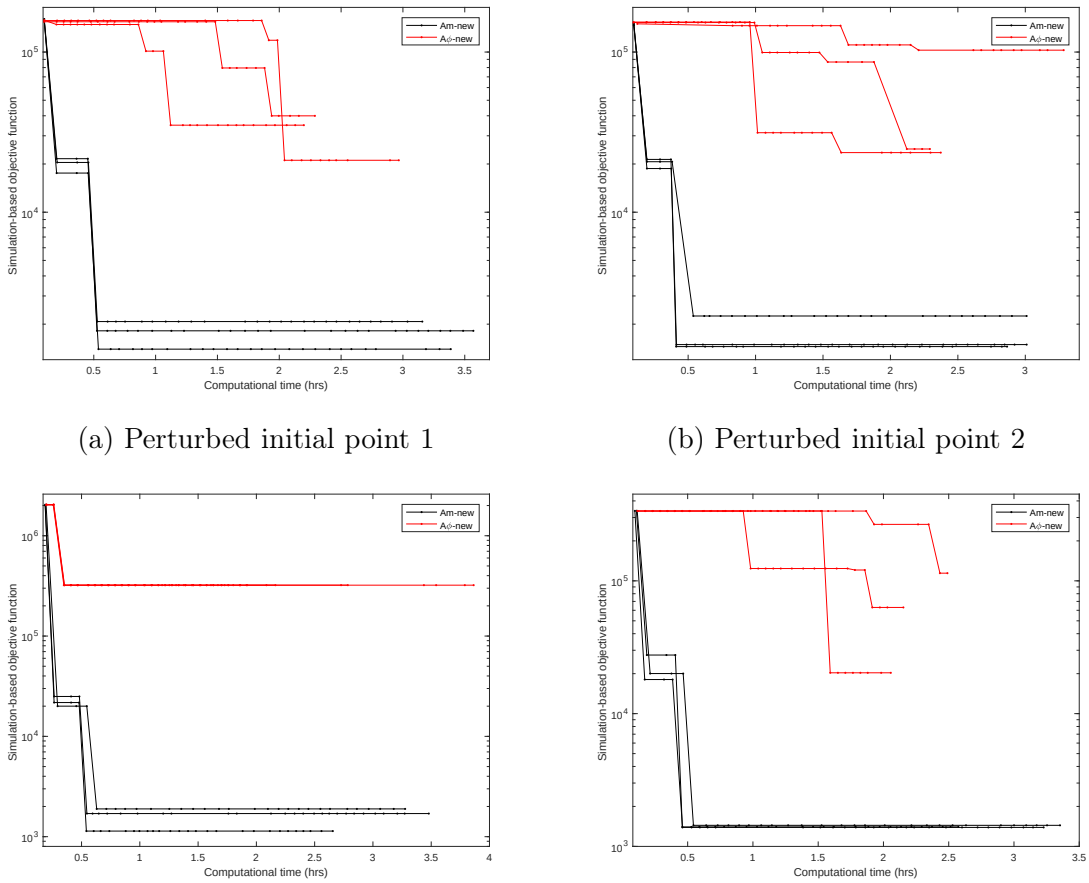
Figure 6-6: Trajectory of the solutions proposed by each algorithm, for different initial points

of $[0, 4000]$ veh/hr, in the background. The left part (i.e., $[0, 1300]$ veh/hr) of all subplots in Figure 6-6 shows gradual changes of color while the remaining part tends to be represented by just a few lines in similar colors. This is due to the fact that the objective function value changes dramatically in the left part and thus contour lines in different colors are clustered. However, the rest part of the objective function is stagnant (i.e., flat). The global minimum of the contour plot is marked with a magenta asterisk and local minima are marked with magenta dots. Over the contour plot, each subplot displays the trajectories of iteration-specific solutions proposed by all 3 runs of *Am-new* and *Aϕ-new* for a given scenario. The solution trajectories for *Am-old* and *Aϕ-old* are not shown here since they use an objective function that is

different from the one shown in the background. Note that the actual objective function for the 4-dimensional problem is more intricate with more local minima than the one displayed in the plot because the plot shows only the first two dimensions of the demand vector by fixing the other two dimensions at their "true" values. The location of initial solutions is marked with a blue triangle. Starting from the initial solution, it is apparent that *Am-new* converges quickly to the vicinity of the global minimum and does so for all 4 scenarios. For all scenarios, *Am-new* first goes to a point near $[1200, 1000]$ veh/hr, which is the minimum identified by using only the analytical network model. After that, *Am-new* identifies points with further improved performance and are close to the global optimal solution. Across all subplots, *Aϕ-new* traverses a larger space by deriving points that are far away from the global minimum. As more points are simulated, *Aϕ-new* slowly makes progress towards the global minimum in some scenarios as seen in Figure 6-6c. For the rest of scenarios, the algorithm *Aϕ-new* is prone to be stuck at local minima that are far away from the "true" solution. Therefore, the added value of the more detailed data source is particularly highlighted when the analytical network model is embedded in the calibration algorithm.

Lastly, we examine the computational efficiency of *Am-new* and *Aϕ-new* by looking at the simulation-based objective function of the current iterate as a function of the computation time, which is displayed for different initial points in Figure 6-7. Figure 6-7 differs from Figure 6-3 in terms of the x-axis, where the former uses the elapsed time in hours instead of the number of simulated points. The overall computation times for both algorithms are comparable. Along the x-axis, *Am-new* achieves a significant reduction in objective function much quicker than *Aϕ-new*. It is worth noting that *Am-new* may take a longer time per simulated point since the total computational time of *Am-new* is greater than *Aϕ-new* in Figures 6-7a, 6-7b, and 6-7d. This is due to the fact that *Am-new* needs to solve a more intricate subproblem optimization problem. Nonetheless, the additional optimization time is still very small compared to the simulation time, and the contribution of the embedded analytical network model clearly justifies the extra computational burden.

In summary, *Am-new* outperforms all other algorithms by achieving good so-



(a) Perturbed initial point 1

(b) Perturbed initial point 2

(c) Random initial point 1

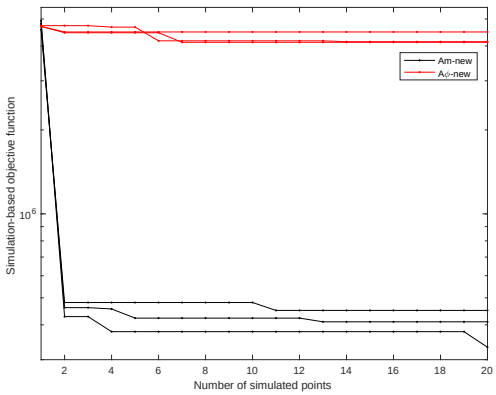
(d) Random initial point 2

Figure 6-7: Simulation-based objective function estimate of the current iterate as a function of the computation time (hrs)

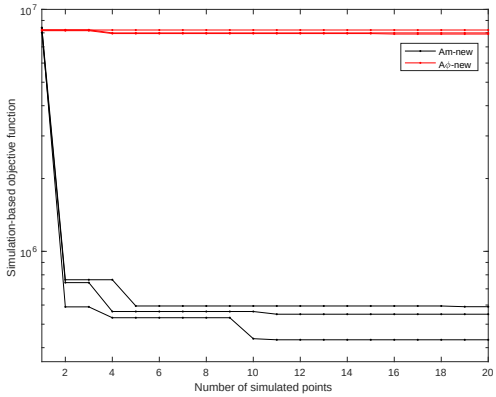
lutions more quickly, resulting in high-quality final solutions, and being robust to the initial points and the stochasticity of simulators, by taking advantage of both the analytical network model and the intersection turning flow information. *Am-old* is inferior to *Am-new* but outperforms all others due to the structural information captured by the analytical network model (e.g., Figures 6-4 and 6-5).

Berlin Metropolitan Network

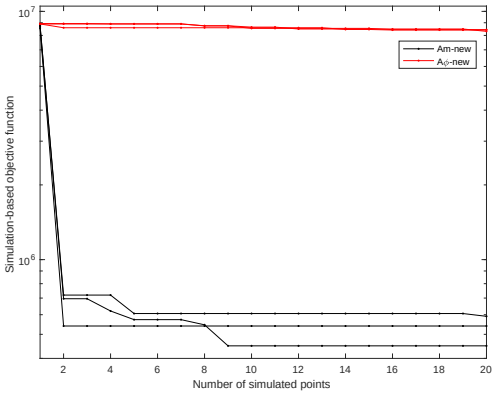
We first compare the performance of *Am-new* and *Aϕ-new*. Figure 6-8 displays the simulation-based objective function of the current iterate as a function of the total number of simulated points. Each subplot corresponds to one scenario with a different



(a) Perturbed initial point 1



(b) Perturbed initial point 2



(c) Random initial point 1

Figure 6-8: Simulation-based objective function estimate of the current iterate as a function of the total number of simulated points

initial point. We consider two scenarios with perturbed initial points and one scenario with a random initial point. In total, there are 6 curves in each subplot representing three runs of *Am-new* and three runs of *Aϕ-new*, respectively. The y-axis uses a logarithmic scale.

Given different initial points, the initial estimates of the simulated objective function (i.e., y-axis values when $x = 1$) are very different. For example, the objective function for the random initial point is 35.6% greater than the average of the other two scenarios, which reflects the inferior quality of the random initial point. For all scenarios, *Am-new* outperforms *Aϕ-new* by achieving a significant smaller objective function across all iterations regardless of the initial point supplied. The most sig-

nificant improvement is made by *Am-new* when the second point, which is obtained by solving the calibration problem using only the analytical network model, is simulated. In contrast, *Aϕ-new* makes small improvements starting from the initial point. The magnitude of the objective function for the final solutions derived by *Am-new* is very similar. However, *Aϕ-new* always ends up with greater objective function values especially when worse initial points (e.g., the random initial point) are used. We then calculate the average objective function value for a given method across all scenarios and all runs, which is 4.82×10^5 for *Am-new*, 6.89×10^6 for *Aϕ-new*, and 7.20×10^6 for the initial points. Compared to the initial points, *Am-new* and *Aϕ-new* on average reduce the objective function value by 93.3% and 4.3%, respectively. In contrast to *Aϕ-new*, *Am-new* further improves it by additional 93.0%. In conclusion, *Am-new* has a superior performance by consistently achieving better solutions and is more robust to both the initial conditions and the stochasticity of the simulator than *Aϕ-new* thanks to the embedded analytical network model.

Furthermore, we examine the quality of algorithmic solutions in terms of their corresponding traffic measurements evaluated by the simulator. Figure 6-9 (resp. Figure 6-10) compares the "true" link counts (resp. intersection turning flows) and the simulated traffic counts (resp. intersection turning flows) for the solutions derived by algorithms *Aϕ-new*, *Am-old*, *Am-new*, and the initial points, denoted by $Init$. Each subplot represents on algorithm, and includes all scenarios, which are differentiated by color. The scenarios with the perturbed initial point 1, the perturbed initial point 2, and the random initial point 1 are marked in red, green, and black, respectively. A dashed blue diagonal line is displayed as a reference.

Both types of traffic measurements corresponding to $Init$ inaccurately overestimate the field data, among which those corresponding to the random initial point are the most inaccurate (i.e., the black markers). The subplots for *Aϕ-new* inherit the pattern presented in the subplots for the initial points, which indicates the dependency/sensitivity of the general-purpose algorithm on/to the initial points. In other words, they are non-robust to the initial conditions. The subplots for *Am-new* and *Am-old* show the best overall approximation to the field data, and *Am-new* outper-

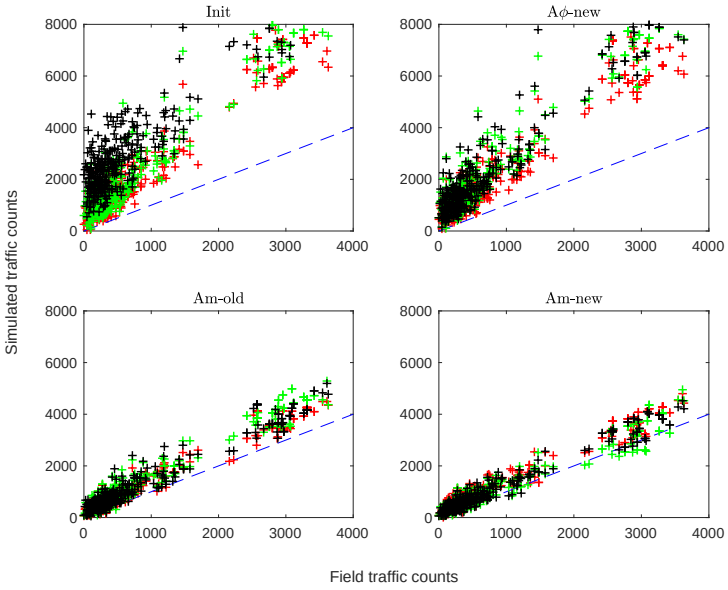


Figure 6-9: Comparison of the "true" link counts and the simulated traffic counts for the algorithmic solutions

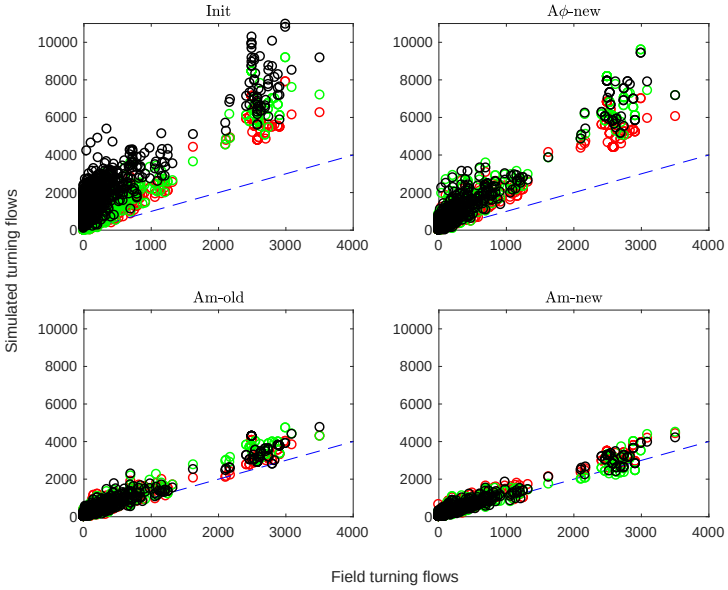


Figure 6-10: Comparison of the "true" turning flows and the simulated turning flows for the algorithmic solutions

forms *Am-old* in terms of both traffic measurements. Therefore, *Am-new* outperforms all other algorithms by benefiting from the added values of both the analytical model and the intersection turning flow data, and the effect of the latter is more obvious in the presence of the former.

We then compare the algorithms' ability to replicate the traffic measurements quantitatively in terms of the root mean square normalized (RMSN) error. The RMSN for traffic counts is defined in Equation (5.8) of Chapter 5. Equation (5.8) can be adjusted to address intersection turning flows by replacing y_i, $\mathcal{I}$, and $\hat{E}[F_i]$ with g_{ij}, $\mathcal{L}_2$, and $\hat{E}[G_{ij}]$, respectively. RMSN quantifies the overall error of calibration. Lower values usually indicate a better fit to the field data. We calculate RMSN for traffic counts and turning flows and list the statistics in Tables 6.3 and 6.4, respectively. For each scenario, the statistics for an algorithm are calculated as the average of three replications of the experiment. Clearly, the algorithms titled *Am* show consistently good performance regardless of the initial conditions, however, the algorithms titled $A\phi$ tend to perform better with higher-quality initial points (e.g., the perturbed initial points). The average RMSN (row 'Average across all initial points') for an algorithm is calculated as the mean across all scenarios. The row at the bottom titled 'Improvement compared to *Init*' is the percentage improvement in terms of the average RMSN of an algorithm over *Init*. Our conclusions drawn from Figures 6-9 and 6-10 hold. *Aϕ-old* and *Aϕ-new* improve the fit to traffic counts and turning flows by about 20% and 26%, respectively. However, *Am-old* and *Am-new*

Table 6.3: Comparison of the Root Mean Square Normalized (RMSN) errors (%) of traffic counts

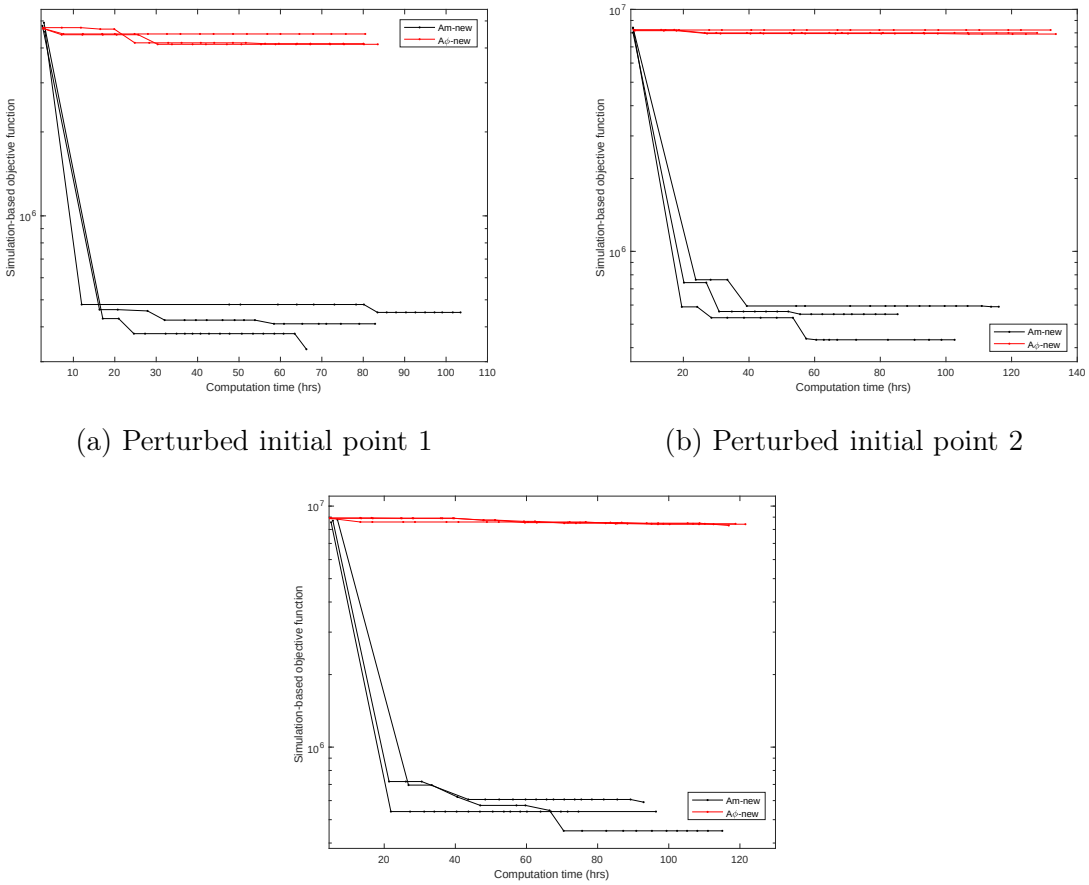
	Init	*Aϕ*-old	*Aϕ*-new	*Am*-old	*Am*-new
Perturbed initial point 1	140.95	132.52	127.82	26.64	25.83
Perturbed initial point 2	187.89	168.29	166.57	36.27	28.66
Random initial point 1	278.31	188.43	193.35	35.18	23.79
Average across all initial points	202.38	163.08	162.58	32.70	26.09
Improvement compared to *Init*	N.A.	19.42	19.67	83.84	87.11

Table 6.4: Comparison of the Root Mean Square Normalized (RMSN) errors (%) of turning flows

	Init	*Aϕ*-old	*Aϕ*-new	*Am*-old	*Am*-new
Perturbed initial point 1	172.54	160.99	155.49	40.41	36.88
Perturbed initial point 2	229.43	204.97	202.66	47.99	44.60
Random initial point 1	398.64	226.89	234.27	48.36	35.42
Average across all initial points	266.87	197.62	197.47	45.59	38.97
Improvement compared to *Init*	N.A.	25.95	26.01	82.92	85.40

improve the fit to both traffic measurements by more than 82%. This reveals the remarkable added value of embedding the analytical network model. The algorithms, *Am-new* and *Aϕ-new*, that use the turning flow data are able to further improve the fit, compared to those without using the new data. In contrast to *Am-old*, *Am-new* improves the fit to traffic counts (resp. turning flows) by additional 20% (resp. 14%). However, *Aϕ-new* achieves slightly better fit than *Aϕ-old* for both traffic measurements. Therefore, the added value of the turning flow data is highlighted for algorithms that embed structural information from the analytical network model. In a word, *Am-new* outperforms all others by taking advantage of both the analytical model and the additional data source.

Last but not least, we evaluate the computational performance of *Am-new* in comparison with *Aϕ-new*. Figure 6-11 displays for each algorithmic run, the simulation-based objective function of the current iterate as a function of the computation time in hours. This figure differs from Figure 6-8 in having the computation time as the x-axis. For a fixed computational budget of 20 simulated points, the computation time for *Am-new* varies in the range of $[66, 116]$ hours while that for *Aϕ-new* falls in the range of $[80, 134]$ hours. It is worth noting that for *Am-new* the most computationally intensive step is from the first iteration to the second one, which corresponds to the step where subproblem optimization is solved for the first time. Solving this high-dimensional subproblem optimization requires about 20 hours, however, it leads to the most significant reduction in the objective function across all algorithmic iterations. Nevertheless, the computation time for solving the subproblem optimization in the subsequent iterations is much shorter for *Am-new*. Note also that this subproblem optimization can be made significantly faster by selecting suitable initial values for the endogenous variables of the analytical network model; this has been such as not to tailor too much the algorithm to the specific problem instance. For *Aϕ-new*, some iterations take a much longer time than the average. This is primarily due to the sampling of model improvement points (i.e., Step 2b in Figure 2-2). On average, *Aϕ-new* samples about 40% more model improvement points than *Am-new*. These model improvement points are sampled randomly from the feasible region, which are



(a) Perturbed initial point 1

(b) Perturbed initial point 2

(c) Random initial point 1

Figure 6-11: Simulation-based objective function estimate of the current iterate as a function of the computation time (hrs)

likely to be inconsistent with the land-use and activity patterns of the area, therefore, require a longer time to simulate. Thanks to the analytical network model, *Am-new* maintains computationally efficiency even with the incorporation of the more detailed data source.

To summarize, *Am-new* appears to be the most robust, effective, and efficient algorithm among the four algorithms under consideration. The Berlin case study demonstrates the scalability of the proposed algorithm and the added values of both the analytical model and the turning flow information. The incorporation of the more detailed data further improves the fit to field data by an average of 20% (resp. 14%), as measured by RMSN errors of traffic counts (resp. intersection turning flows). *Am-*

new is an algorithm that is capable of incorporating additional traffic data and thus has the potential to be extended to address other emerging data sources.

6.4 Conclusions and Discussion

This chapter presents an enhanced metamodel algorithm for the offline OD demand calibration of large-scale stochastic traffic simulators that leverages emerging data sources of higher resolution. Specifically, we propose a computationally efficient demand calibration algorithm that takes advantage of the added value of using intersection turning flow data collected by AVI sensors. The algorithm is based on an SO metamodel approach, which uses an analytical metamodel to approximate the simulator and then uses the metamodel for optimization. In this chapter, we use a link-independent metamodel to approximate the objective function. The metamodel integrates information from both the simulator and the analytical network model. A computationally efficient analytical network model with endogenous route choice, which maps the calibration vector to corresponding traffic measurements analytically, is used.

The case studies compare the performance of our proposed algorithm to three other benchmark algorithms. We demonstrate that the proposed algorithm is able to handle multiple traffic data types and outperforms the benchmark algorithms in terms of the quality of solutions, computational efficiency, and robustness to both the initial conditions and the stochasticity of simulators. Through pair-wise comparative analyses of all algorithms, we showed that incorporating more detailed traffic data significantly improves the fit to field data, and integrating the structural information derived from the analytical network model maintains the computational efficiency and improves the robustness of the SO algorithm. According to the Berlin case study, the proposed algorithm yields an average of 93% improvement in the quality of the final solution, as measured by its objective function estimates, and simultaneously improves the fit to field data by more than 10%, as measured by the RMSN errors for both traffic counts and turning flows, in contrast to a general-purpose algorithm

that does not embed the analytical network model.

Chapter 7

Conclusions and Future Research

Traffic simulators are the mathematical modeling of transportation systems through the application of computer software to better plan, design, and operate transportation systems. This thesis is shaped by the emerging trend of using simulators for urban transportation system modeling. Simulation models have grown in complexity as more detailed information is wanted by transportation agencies, academic institutions, and consulting firms. This added complexity poses a challenge for calibrating model input parameters based on real traffic data.

7.1 Conclusions

This thesis presents computationally efficient offline demand calibration algorithms for large-scale stochastic traffic simulation models. First, the demand calibration problem is formulated as a simulation-based optimization (SO) problem. Then, metamodel solution approaches are provided. The metamodel approaches embed problem-specific structural information from an analytical network model, which describes analytically how the calibration input is related to the calibration objective function. The proposed approaches are shown to achieve significant improvements in the computational efficiency, the robustness to both the initial conditions and the stochasticity of simulators, and the quality of final solutions obtained. Moreover, these approaches are scalable and thus are applicable to large-scale real world instances.

In addition, these approaches enable the use of multiple heterogeneous traffic data sources, including those collected by emerging technologies such as AVI and AVL systems. Given all these features, the proposed metamodel calibration algorithms are suitable for a broad family of simulators, are of particular interest for the calibration of inefficient simulators, and are likely to be supported by big data in the future.

Chapter 3 formulates the behavioral model calibration problem as an SO problem, proposes a novel metamodel solution approach, and implements it for the calibration of a behavioral parameter governing drivers' route choice. The analytical metamodel combines information from a queueing-based analytical network model and information from the traffic simulator. The queueing-based analytical network model captures assignment endogenously through the integration of the multinomial logit (MNL) model. The proposed algorithm is validated by considering synthetic experiments on a toy network. It is then used to address a calibration problem with real data for a large-scale network: the Berlin metropolitan network with over 24,300 links and 11,300 nodes. The performance of the proposed approach is compared to a traditional benchmark method that differs only in that the metamodel does not use information from the analytical model. The proposed approach significantly improves the computational efficiency of the calibration algorithm with an average reduction in simulation runtime until convergence of more than 80%. The results illustrate the scalability of the approach and its suitability for the calibration of large-scale computationally inefficient network simulators.

Chapter 4 extends the algorithm proposed in Chapter 3 to address a 2-dimensional calibration of the supply models. Specifically, link flow capacity and link storage capacity are calibrated from traffic counts. A simplified analytical network model that represents route choice exogenously is presented as a system of nonlinear equations. Its dimension scales linearly with the number of links in the network and scales independently of the link attributes (e.g., link lengths). The proposed supply calibration algorithm is benchmarked against an SO algorithm that differs only in that the metamodel does not use information from the analytical model. The performance of the two methods is compared with experiments on both a toy network and the Berlin

metropolitan network, considering a range of congestion levels. The case studies indicate that the use of the analytical network model allows the algorithm to: converge more frequently, converge faster, and be more robust to both the quality of the initial points and to the stochasticity of the simulator. More specifically, for the Berlin case study, the proposed supply calibration algorithm improves by a factor of 3 the convergence frequency, by 20% the convergence speed, and by 72% the total simulation runtime required.

Chapter 5 tackles a high-dimensional calibration problem concerning the origin-destination (OD) demand matrices. The proposed analytical network model is formulated as a simple system of linear equations that scales linearly with the number of links in the network, and scales independently of other link attributes, such as link lengths. This leads to an algorithm that is computationally efficient and suitable for high-dimensional, non-convex, SO problems of large-scale road networks. Experiments on a small toy network illustrate the ability of the analytical network model to approximate the simulation-based objective function and to identify the location of the global optimum. We carry out a case study to calibrate the demand for the morning peak hour for the Berlin (Germany) metropolitan area. Compared to the benchmark method that differs only in that the metamodel does not use information from the analytical network model, the proposed method yields an average 70% improvement in the quality of the solution, as measured by its objective function estimates, while simultaneously reducing the computational runtimes by an average of 30%. The various experiments indicate that the analytical structural information yields an algorithm that is robust to both the quality of the initial points and the stochasticity of the simulator. The structural information enables the algorithm to identify solutions with significantly improved performance at the first iteration, when little or even no simulation information is available.

Chapter 6 explores opportunities within the context of OD demand calibration by taking advantage of high-resolution emerging data sources. Traditional OD demand calibration resorts to computationally inefficient "black-box" algorithms, largely restricted to conventional traffic data, such as traffic counts that are typically scarce,

under-specifying the problem. This chapter integrates more detailed data sources to improve performance of the calibration algorithm. As a first step, we use intersection turning flows to enhance an SO calibration. This enhanced algorithm is designed to identify solutions with good performance within very limited computational budgets by exploiting the transportation problem-specific structure, for both conventional traffic data (i.e., traffic counts) and more detailed data (i.e., intersection turning flows). This information is derived from a fundamental diagram-based analytical network model, upon which a metamodel is constructed and used to approximate the traffic simulation model. Validation on a toy network and a case study on a real-world large-scale network in the metropolitan area of Berlin, Germany demonstrate that the enhanced OD calibration algorithm further improves the quality of solutions, maintains the computational efficiency of the calibration algorithm, is robust to both the initial points and the stochasticity of simulators, and is capable of handling multiple heterogeneous data sources. The intersection turning flows refine the objective function by reducing the total number of local minima. For instance, the inclusion of intersection turning flows further enhances performance compared to the approach that does not use this data by improving the fit to field data by an average of 20% (resp. 14%), as measured by the RMSN errors of traffic counts (resp. intersection turning flows).

7.2 Future Research

This thesis points to several directions for future research and applications.

7.2.1 Extensions for General Applicability

While the demand calibration framework developed in this thesis is generic, the case studies involve necessary additional assumptions and simplifications regarding the time-dependency, the scope of calibration, and the types of input field data, etc. The methodology can be extended for broader applications by relaxing some or all of these assumptions.

With the increase in the availability, diversity, and quality of travel data, comes increasing interest and relevance of the joint calibration of demand and supply parameters (e.g., Balakrishna, 2006; Antoniou *et al.*, 2007; Vaze *et al.*, 2009). This metamodel framework can be extended to allow for other input parameters of demand models, supply models, or for joint calibration of both models. This requires the development of appropriate analytical network models that relate all the calibration input parameters to the objective function. Challenges and difficulties associated with this are regarding dimensionality, scalability, and efficiency. If the extension of the analytical model deems computationally inefficient, then the current analytical network model can be used as is, and the additional input parameters can be included in the polynomial error term of the metamodel.

In this thesis, only normal automobile traffic (i.e., road transportation) is considered. However, urban transportation is a complex system consisting of multiple systems, including public transit, pedestrian, bike, and (eventually) autonomous vehicles (AV), etc. To fully understand the travel demand, the current calibration framework can be extended to account for all relevant transportation modes. This requires a traffic model that integrates multiple transportation systems and captures the interactions among them.

Another future research direction is to investigate the use of the metamodel calibration methodology to enhance the efficiency and robustness of traditional data-driven calibration algorithms such as Kalman filters and particle filters.

7.2.2 Improvements for Enhanced Performance

In addition to extensions for general applicability, the work of this thesis has potential to be improved for better performance in terms of computational efficiency, accuracy, and reliability. Research along this line can focus on the improvement of either the problem formulation, or the metamodel SO algorithm, or both.

The current problem formulation is based on traffic measurements, such as expected traffic flow/demand and expected intersection turning flows, that are first-order moments of network performance. One approach to improve the problem formu-

lation is to account for higher-order distributional information. For example, second-order distributional information such as the variance of traffic flow can be included in the problem formulation. In order to use the metamodel SO calibration framework, the derivation of distributional traffic measurements from both the analytical network model and the simulation model is required. Chen *et al.* (2012) address a traffic signal control problem by formulating the objective function in terms of the expectation and the standard deviation of total link travel time and demonstrate that this formulation can lead to enhanced network reliability and enhanced network robustness. However, a major challenge is to develop an analytical and tractable approximation of the distribution of the main network performance measures.

In this thesis, the weight for the prior OD demand matrix, δ_1, and the weight for the term addressing the additional traffic data, δ_2, are pre-determined based on the relative confidence of the particular data over traffic counts. The prior OD demand matrix is usually estimated from historical records (e.g., census, survey) that are likely outdated, therefore, a very small value (e.g., 0.01) is assigned to δ_1 according to the literature. On the other hand, a relative large value (e.g., 1) is assigned to δ_2. It is interesting to explore the importance of the weights via sensitivity analysis. In other words, the robustness of the demand calibration regarding the selection of the weights can be evaluated and better values for the weights can be obtained through fine-tuning.

Chapter 6 develops an enhanced demand calibration algorithm by leveraging an additional high-resolution data source: intersection turning flows. In the near future, it is likely that demand calibration will be expanded by advances in big data (e.g., social media data, cell-phone traces, and GPS trajectories) with higher resolution information. The extension of the formulation to leverage more detailed data sources, such as subpath travel times and partial vehicle trajectories, for enhanced performance is of interest. Big data techniques, e.g., Hadoop and Spark, at some point can also be integrated into the metamodel SO calibration framework to improve the computational efficiency, especially for processing large datasets as input. However, one challenge associated with this is the derivation of an analytical approximation for

the new data (e.g., trajectories) and the selection of error metrics (i.e., loss function).

The methodology developed in this thesis can also be used in combination with other dimensionality reduction techniques (e.g., sensitivity analysis, principal component analysis) to further boost scalability and efficiency. For instance, dimensionality reduction techniques can be applied first to transform the high-dimensional calibration parameter vector (e.g., OD demand matrices) into a low-dimensional variable space. Then, metamodel SO techniques can be used to tackle the reduced calibration problem more effectively and efficiently.

The work of Qin and Mahmassani (2004) discusses the importance of accounting for sample selection bias in the formulation of the calibration problem. This topic has not received much attention in the literature, yet is important and if taken into account has the potential to improve the quality of the calibration. In addition, Treiber and Kesting (2013) discuss additional interesting open questions in the field of calibration.

7.2.3 Other Related Problems

Currently, demand calibration still primarily depends on traffic counts that are collected by conventional sensors (e.g., inductive loop traffic detectors) at specific locations in the network, whose deployment is crucial to the success of calibration since traffic counts from these sensors can provide different information on OD flows. Ideally, sensors can be placed in a way to ensure the uniqueness of OD flows or installed on road segments associated exclusively to routes for a specific OD pair, which usually requires a large number of sensors being deployed. In practice, the total number of sensors deployed is limited by budget constraints. Thus, demand calibration usually appears to be under-specified.

A practically significant problem that is closely related to the work of this thesis is the identification of optimal spatial distributions of sensors. Given a set of candidate locations, the problem is to make binary choice of sensor deployment at each location in order to minimize the total number (or cost) of sensors deployed and to maximize the calibration accuracy. When there is a clearly defined budget, an addi-

tional constraint is imposed to ensure that the budget is not exceeded. This problem is challenging for the following reasons:

- The set of candidate locations for sensor placement is huge, especially for large-scale real-world networks;
- The measurements from different sensors are highly correlated;
- The impact from other types of traffic data also needs to be considered. Additional data sources are becoming widely available, which complement traffic counts and contribute marginally to the accuracy of calibration.

To combat these, one future research direction is to apply sensitivity analysis techniques to address the sensor coverage problem by exploring the relationship between the number of sensors and the level of OD coverage in a network.

Recognizing the significance of sensor location and its relationship to the quality of demand calibration, it is important to establish a connection between the two critical problems.

Appendix A

Metamodel Simulation-based Optimization (SO) Calibration Algorithm

This appendix presents the metamodel simulation-based optimization (SO) calibration algorithm. The algorithm is described step by step in Algorithm 1 using the notation of Osorio and Bierlaire (2013).

The following notations are defined for a given iteration k: current iterate as x_k, trust region radius as Δ_k, step size as s_k, metamodel coefficient vector as ν_k, metamodel as m_k, total number of simulation runs carried out up until and including iteration k as n_k, and total number of successive trial points rejected as μ_k.

The constants η_1, $\bar{\gamma}$, γ_{inc}, $\bar{\tau}$, $\bar{d}$, $\bar{\mu}$, Δ_{max} are given such that: $0 < \eta_1 < 1$, $0 < \bar{\gamma} < 1 < \gamma_{inc}$, $0 < \bar{\tau} < 1$, $0 < \bar{d} < \Delta_{max}$, $\bar{\mu} \in \mathbb{N}^*$. Set the total number of simulation runs permitted, n_{max}, this determines the computational budget. Set the number of simulation replications per point $\bar{r}$.

Algorithm 1 Metamodel SO calibration algorithm

Initialization

Set $k = 0$, $n_0 = 0$, $\mu_0 = 0$. Determine the initial point x_0 and the initial trust region radius Δ_0 ($\Delta_0 \in (0, \Delta_{max}]$). Evaluate $f_A(x_0)$ by solving the analytical network model and compute $\hat{f}(x_0)$. Include the new simulation observation in the set of sampled points, i.e., set $n_0 = n_0 + \bar{r}$.

Analytical-only calibration

Solve Problem (2.4)-(2.5) using only the analytical network model and without using any simulation information. This is done by setting the metamodel equal to f_A. Let $\tilde{x}$ denote the solution to this problem. Evaluate $f_A(\tilde{x})$ and compute $\hat{f}(\tilde{x})$. Set $n_0 = n_0 + \bar{r}$. Set the initial current iterate to this solution, i.e., set $x_0 = \tilde{x}$.

Initial metamodel fitting

Solve Problem (C.1) to fit the metamodel m_0.

while $n_k < n_{max}$ **do**

  Step calculation

  Solve Problem (2.4)-(2.5) subject to a trust-region constraint (i.e., $\|s_k\| \leq \Delta_k$), let $x_k + s_k$ denote the solution, which is referred to as the trial point.

  Acceptance or rejection of the trial point

  Evaluate $f_A(x_k + s_k)$ by solving the analytical network model and compute $\hat{f}(x_k + s_k)$. Set $n_k = n_k + \bar{r}$. Compute:

$$\rho_k = \frac{\hat{f}(x_k) - \hat{f}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

  if $\rho_k \geq \eta_1$ and $\hat{f}(x_k) - \hat{f}(x_k + s_k) > 0$ **then**

    Accept the trial point: $x_{k+1} = x_k + s_k$, $\mu_k = 0$;

  else

    Reject the trial point: $x_{k+1} = x_k$, $\mu_k = \mu_k + 1$.

  end if

  Solve Problem (C.1) to re-fit the metamodel m_{k+1}.

  Model improvement

  Compute $\tau_{k+1} = \frac{\|\nu_{k+1} - \nu_k\|}{\|\nu_k\|}$. If $\tau_{k+1} < \bar{\tau}$, then sample a new point x, which is uniformly and randomly drawn from the feasible space. Evaluate $f_A(x)$ by solving the analytical network model and compute $\hat{f}(x)$. Include this new observation in the set of sampled points and set $n_k = n_k + \bar{r}$. Solve Problem (C.1) to update the fit of the metamodel m_{k+1}.

  Trust region radius update

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}, & \text{if } \rho_k > \eta_1 \\ \max\{\bar{\gamma}\Delta_k, \bar{d}\}, & \text{if } \rho_k \leq \eta_1 \text{ and } \mu_k \geq \bar{\mu} \\ \Delta_k, & \text{otherwise.} \end{cases}$$

  if $\rho \leq \eta_1$ and $\mu_k \geq \bar{\mu}$ **then**

    Set $\mu_k = 0$. Set $n_{k+1} = n_k$, $\mu_{k+1} = \mu_k$, $k = k + 1$.

  end if

end while

The general-purpose method $A\phi$ differs from the proposed method Am as follows: (i) the step "Analytical-only calibration" (lines 3 and 4) is not carried out; (ii) the metamodel does not contain a problem-specific component f_A derived from the analytical network model by setting the metamodel equal to the general-purpose metamodel ϕ.

This general SO calibration algorithm described above can be customized to a specific algorithm developed in Chapters 3 to 6 by setting x to the specific calibration vector (e.g., θ in Chapter 3, α in Chapter 4, and d in Chapters 5-6). In addition, the metamodel fitting is different for link-dependent (e.g., Chapters 3 to 5) and link-independent (e.g., Chapter 6) metamodels.

Appendix B

Additional Experimental Results

B.1 Numerical Values of the Solutions Derived by Each Method of Chapter 3

Table B.1 (resp. B.2) displays the numerical values of the solutions derived by each method and each experiment of the toy (resp. Berlin) network of Chapter 3.

Table B.1: Numerical values of the solutions (in units hr^{-1}) derived by each method for each experiment of the toy network

			Am			$A\phi$	
$\theta^* = -5$	$\theta_0 = 0$	-4.7	-5.3	-4.0	-5.2	-5.3	-5.5
	$\theta_0 = -40$	-5.7	-5.5	-5.0	-5.2	-5.5	-5.3
	$\theta_0 = -60$	-4.8	-5.5	-5.8	-5.1	-5.0	-5.4
$\theta^* = -20$	$\theta_0 = 0$	-18.1	-18.4	-18.7	-18.3	-19.1	-18.5
	$\theta_0 = -40$	-19.6	-18.1	-18.6	-18.6	-17.7	-18.3
	$\theta_0 = -60$	-18.1	-18.5	-19.5	-18.3	-18.5	-18.0
$\theta^* = -55$	$\theta_0 = 0$	-56.1	-55.9	-55.9	-57.2	-56.4	-56.0
	$\theta_0 = -40$	-55.8	-56.3	-55.2	-56.0	-56.2	-56.4
	$\theta_0 = -60$	-55.8	-55.5	-56.6	-55.8	-56.3	-56.0

Table B.2: Numerical values of the solutions (in units hr^{-1}) derived by each method for each experiment of the Berlin network

	Am			$A\phi$		
$\theta_0 = 0$	-2.5	-1.2	-2.1	0.0	0.0	-1.1
$\theta_0 = -40$	-4.7	-4.7	-1.6	0.0	0.0	-2.8
$\theta_0 = -60$	-1.4	-4.7	-3.5	0.0	0.0	-0.4

B.2 Analysis of the Root Mean Square Normalized (RMSN) Errors for the Synthetic Toy Network Experiments of Chapter 6

Tables B.3 to B.4 list the RMSN statistics for traffic counts and intersection turning flows, respectively, for the toy network. For each scenario, the statistics for an algorithm are calculated as the average of three replications of the experiment. The average RMSN for an algorithm is calculated as the mean across all scenarios. The row "Improvement compared to *Init*" represents the percentage improvement of an algorithm over *Init* in terms of the average RMSN.

Table B.3: Comparison of the Root Mean Square Normalized (RMSN) errors (%) of traffic counts

	Init	$A\phi$-old	$A\phi$-new	Am-old	Am-new
Perturbed initial point 1	22.13	0.70	3.49	2.38	1.42
Perturbed initial point 2	18.56	1.22	0.26	5.61	1.61
Random initial point 1	77.75	7.34	7.34	3.40	1.20
Random initial point 2	21.67	2.26	0.69	3.06	0.43
Average across all initial points	35.03	2.88	2.95	3.61	1.17
Improvement compared to *Init*	N.A.	91.78	91.58	89.69	96.66

Table B.4: Comparison of the Root Mean Square Normalized (RMSN) errors (%) of turning flows

	Init	$A\phi$-old	$A\phi$-new	Am-old	Am-new
Perturbed initial point 1	22.78	28.88	19.33	21.72	1.65
Perturbed initial point 2	30.65	28.00	18.05	12.76	1.65
Random initial point 1	85.43	68.29	68.29	20.08	1.38
Random initial point 2	57.30	16.35	41.67	12.83	0.38
Average across all initial points	49.04	35.38	36.84	16.85	1.27
Improvement compared to *Init*	N.A.	27.85	24.88	65.64	97.41

Appendix C

Implementation Details

C.1 Link Attributes for the Synthetic Toy Networks

Tables C.1 to C.4 list the attributes of each link of the synthetic toy networks used in Chapters 3 to 6, respectively.

Link index	Nodes connected	Length (km)	Maximum speed (km/h)	Signalized
1	$1 \rightarrow 2$	2.5	72	un-signalized
2	$2 \rightarrow 3$	7.5	54	signalized
3	$3 \rightarrow 5$	2.5	54	un-signalized
4	$2 \rightarrow 4$	7.1	72	un-signalized
5	$4 \rightarrow 5$	7.1	72	un-signalized
6	$5 \rightarrow 6$	2	72	un-signalized

Table C.1: Toy network link properties of Chapter 3

Link index	Nodes connected	Flow capacity (veh/hr)	Length (km)	Maximum speed (km/h)
1	$1 \rightarrow 2$	1,800	2.5	72
2	$2 \rightarrow 3$	800	7.5	108
3	$3 \rightarrow 5$	800	2.5	108
4	$2 \rightarrow 4$	1,200	7.1	54
5	$4 \rightarrow 5$	1,200	7.1	54
6	$5 \rightarrow 6$	1,800	2	72

Table C.2: Toy network link properties of Chapter 4

Link index	Nodes connected	Flow capacity (veh/hr)	Length (km)	Maximum speed (km/h)
1	$1 \rightarrow 3$	1,800	5	72
2	$2 \rightarrow 5$	1,800	5	72
3	$3 \rightarrow 4$	800	8.6	108
4	$5 \rightarrow 4$	800	8.6	108
5	$3 \rightarrow 6$	1,000	21	54
6	$4 \rightarrow 7$	2,000	7	108
7	$5 \rightarrow 8$	1,000	21	54
8	$7 \rightarrow 6$	800	8.6	108
9	$7 \rightarrow 8$	800	8.6	108
10	$6 \rightarrow 9$	1,800	5	72
11	$8 \rightarrow 10$	1,800	5	72

Table C.3: Toy network link properties of Chapter 5

Link index	Nodes connected	Flow capacity (veh/hr)	Length (km)	Maximum speed (km/h)
1	$1 \rightarrow 2$	2,000	1	111.6
2	$2 \rightarrow 3$	2,000	0.9144	111.6
3	$3 \rightarrow 4$	2,000	1	111.6
4	$2 \rightarrow 5$	2,000	0.2438	111.6
5	$5 \rightarrow 6$	2,000	0.3048	111.6
6	$6 \rightarrow 3$	2,000	0.2438	111.6
7	$7 \rightarrow 8$	2,000	1	111.6
8	$8 \rightarrow 5$	2,000	0.2438	111.6
9	$9 \rightarrow 6$	2,000	0.2438	111.6
10	$6 \rightarrow 10$	2,000	0.2438	111.6
11	$10 \rightarrow 11$	2,000	1	111.6

Table C.4: Toy network link properties of Chapter 6

C.2 Travel Demand Representation

The transport simulator MATSim uses a plan (or population) file as input that describes the travel demand of individual travelers in terms of a list of daily plans, which is a list of chained activities and trips for a given day. Each trip specifies an origin link and a destination link. We refer to a pair of such links as an OD link pair. OD demand calibration problems are traditionally defined in terms of zone to zone demand. A given link-level demand is aggregated into a zone-level demand by associating each link to its zone. A zone-level demand is disaggregated into a link-level demand by using a fixed and exogenous proportionality factor, which defines the proportion of demand for a given OD zone pair that is allocated to a given OD link

pair. This proportionality factor is estimated based on the daily plan file without the need for any simulation.

C.3 Metamodel Fitting Process

The iteration-specific metamodel parameters β_k are determined by solving a least squares regression problem using the simulated traffic measurements obtained both at the current iteration as well as at all previous iterations. In this section, the notation of Appendix A is used. The following additional notation is needed to formulate the regression problem.

x_k	current iterate;
$\hat{f}(x)$	simulation-based estimate of the objective function for point x;
$w_k(x)$	weight for point x;
w_0	exogenous (fixed) weight coefficient, set to 0.01;
$\mathcal{S}_k$	set of points simulated up until iteration k;
d	dimension of the calibration vector.

The least squares problem is formulated as follows.

$$\min_{\beta_k} \sum_{x \in \mathcal{S}_k} \left\{ w_k(x)(\hat{f}(x) - m_k(x; \nu_k)) \right\}^2 + w_0^2 \left((\nu_{k,0} - 1)^2 + \sum_{i=1}^{d} \nu_{k,i}^2 \right). \tag{C.1}$$

The first term represents the weighted distance between the objective functions predicted by the metamodel and those estimated by the simulator. The weight for the point x is defined as in Osorio and Bierlaire (2013):

$$w_k(x) = \frac{1}{1 + \|x - x_k\|_2}, \tag{C.2}$$

where a point's weight is inversely proportional to its distance from the current iterate, x_k, aiming to improve the local (in the vicinity of the current iterate) fit of metamodel. To ensure the least square matrix is of full rank, the second term of Problem (C.1)

accounts for the distance between the parameter vector, ν_k, and initial values. The initial values used correspond to an initial metamodel that is solely based on the analytical network model.

For the algorithms of Chapters 3 and 4, the terms $\hat{f}$ and m_k in Problem (C.1) refer to the function f as defined in the objective function (3.4) and the entire summation as defined in the objective function (3.6), respectively; in Chapter 5, they refer to the function f as defined in the objective function (5.1) and the function m_k^{obj} as defined in the objective function (5.4); and in Chapter 6, they refer to the function f as defined in the objective function (6.1) and the function m_k as defined in the objective function (6.5) but without the last term $\delta_1 \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} (\tilde{d}_z - d_z)^2$.

Bibliography

Alibabai, H. and Mahmassani, H. S. (2008). Dynamic origin-destination demand estimation using turning movement counts. *Transportation Research Record*, **2085**, 39–48.

Antoniou, C. (2004). *On-line Calibration for Dynamic Traffic Assignment*. Ph.D. thesis, Massachusetts Institute of Technology.

Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. N. (2006). Dynamic traffic demand prediction using conventional and emerging data sources. In *IEE Proceedings-Intelligent Transport Systems*, pages 97–104. IET Digital Library.

Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. (2007). Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models. *IEEE Transactions on Intelligent Transportation Systems*, **8**(4), 661–670.

Antoniou, C., Balakrishna, R., Koutsopoulos, H. N., and Ben-Akiva, M. (2011). Calibration methods for simulation-based dynamic traffic assignment systems. *International Journal of Modeling and Simulation*, **31**(3), 227–233.

Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F., and Ben-Akiva, M. (2015). W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Part C*, **59**, 129–146.

Asakura, Y., Hato, E., and Kashiwadani, M. (2000). Origin-destination matrices estimation model using automatic vehicle identification data and its application to the Han-Shin expressway network. *Transportation*, **27**(4), 419–438.

Balakrishna, R. (2006). *Off-line calibration of dynamic traffic assignment models*. Ph.D. thesis, Massachusetts Institute of Technology.

Balakrishna, R., Ben-Akiva, M. E., and Koutsopoulos, H. N. (2007). Offline calibration of dynamic traffic assignment: simultaneous demand-and-supply estimation. *Transportation Research Record*, **2003**, 50–58.

Barceló, J. (2010). *Fundamentals of traffic simulation*. International Series in Operations Research and Management Science. Springer, New York, USA.

Barceló, J., Mercade, L. M., Marques, L., and Carmona, C. (2010). A Kalman-filter approach for dynamic OD estimation in corridors based on Bluetooth and Wi-Fi data collection. In *12th World Conference on Transportation Research WCTR*.

Barceló, J., Montero, L., Bullejos, M., Serch, O., and Carmona, C. (2012). Dynamic OD matrix estimation exploiting Bluetooth data in urban networks. *Recent Researches in Automatic Control and Electronics*, pages 116–121.

Ben-Akiva, M., Koutsopoulos, H., Toledo, T., Yang, Q., Choudhury, C., Antoniou, C., and Balakrishna, R. (2010). Traffic simulation with MITSIMLab. In *Fundamentals of Traffic Simulation*, pages 233–268. Springer.

Ben-Akiva, M., Gao, S., Wei, Z., and Wen, Y. (2012). A dynamic traffic assignment model for highly congested urban networks. *Transportation Research Part C*, **24**, 62–82.

Bierlaire, M. and Crittin, F. (2004). An efficient algorithm for real-time estimation and prediction of dynamic OD tables. *Operations Research*, **52**(1), 116–127.

Bocharov, P. P., D'Apice, C., Pechinkin, A. V., and Salerno, S. (2004). *Queueing theory*, chapter 3, pages 96–98. Modern Probability and Statistics. Brill Academic Publishers, Zeist, The Netherlands.

Box, M. J. (1965). A new method of constrained optimization and a comparison with other methods. *The Computer Journal*, **8**(1), 42–52.

Caceres, N., Wideberg, J. P., and Benitez, F. G. (2007). Deriving origin-destination data from a mobile phone network. *IET Intelligent Transportation Systems*, **1**(1), 15–26.

Cao, P., Miwa, T., Yamamoto, T., and Morikawa, T. (2013). Bi-level generalized least squares estimation of dynamic origin-destination matrix for urban network with probe vehicle data. *Transportation Research Record*, **2333**, 66–73.

Cascetta, E. and Nguyen, S. (1988). A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B*, **22**(6), 437–455.

Cascetta, E., Inaudi, D., and Marquis, G. (1993). Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science*, **27**(4), 363–373.

Chen, X., Osorio, C., and Santos, B. F. (2012). A simulation-based approach to reliable signal control. In *Proceedings of the International Symposium on Transportation Network Reliability (INSTR)*. Available at: http://web.mit.edu/osorioc/www/papers/osoCheSanReliableSO.pdf .

Chen, X., Xiong, C., He, X., Zhu, Z., and Zhang, L. (2016). Time-of-day vehicle mileage fees for congestion mitigation and revenue generation: a simulation-based optimization method and its real-world application. *Transportation Research Part C*, **63**, 71–95.

Chong, L. and Osorio, C. (2018). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*, **52**(3), 637–656.

Chu, L., Liu, H. X., Oh, J. S., and Recker, W. (2003). A calibration procedure for microscopic traffic simulation. *IEEE Transactions on Intelligent Transportation Systems*, **2**, 1574–1579.

Cipriani, E., Florian, M., Mahut, M., and Nigro, M. (2011). A gradient approximation approach for adjusting temporal origin-destination matrices. *Transportation Research Part C*, **19**(2), 270–282.

Cipriani, E., Nigro, M., Fusco, G., and Colombaroni, C. (2014). Effectiveness of link and path information on simultaneous adjustment of dynamic OD demand matrix. *European Transport Research Review*, **6**(2), 139–148.

Ciuffo, B. and Azevedo, C. L. (2014). A sensitivity-analysis-based approach for the calibration of traffic simulation models. *IEEE Transactions on Intelligent Transportation Systems*, **15**(3), 1298–1309.

Ciuffo, B., Punzo, V., and Montanino, M. (2015). Global sensitivity analysis techniques to simplify the calibration of traffic simulation models. Methodology and application to the IDM car-following model. *IET Intelligent Transport Systems*, **8**(5), 479–489.

Coifman, B. and Krishnamurthy, S. (2007). Vehicle reidentification and travel time measurement across freeway junctions using the existing detector infrastructure. *Transportation Research Part C*, **15**, 135–153.

Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). *Introduction to derivative-free optimization*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, USA.

Corthout, R., Himpe, W., Viti, F., Frederix, R., and Tampere, C. M. (2014). Improving the efficiency of repeated dynamic network loading through marginal simulation. *Transportation Research Part C*, **41**, 90–109.

Dixon, M. P. and Rilett, L. R. (2002). Real-time OD estimation using automatic vehicle identification and traffic count data. *Computer-Aided Civil and Infrastructure Engineering*, **17**(1), 7–21.

Djukic, T. (2014). *Dynamic OD demand estimation and prediction for dynamic traffic management*. Ph.D. thesis, Delft Univeristy of Technology.

Djukic, T., Bugeda, J. B., Bullejos, M., Mercadé, L. M., Cipriani, E., van Lint, H., and Hoogendoorn, S. (2015). Advanced traffic data for dynamic OD demand estimation: The state of the art and benchmark study. In *TRB 94th Annual Meeting Compendium of Papers*, pages 1–16.

Eisenman, S. M. and List, G. F. (2004). Using probe data to estimate OD matrices. In *Proceedings of the Intelligent Transportation Systems Conference*, pages 291–296.

Flötteröd, G. (2016). Queueing representation of kinematic waves. *The Multi-Agent Transport Simulation MATSim, Ubiquity, London.*, pages 347–351. DOI: http://dx.doi.org/10.5334/baw.50 .

Flötteröd, G. and Bierlaire, M. (2009). Improved estimation of travel demand from traffic counts by a new linearization of the network loading map. In *Proceedings of the European Transport Conference*, Noordwijkerhout, The Netherlands.

Flötteröd, G. and Bierlaire, M. (2013). Metropolis-Hastings samplings of paths. *Transportation Research Part B*, **48**, 53–66.

Flötteröd, G., Bierlaire, M., and Nagel, K. (2011). Bayesian demand calibration for dynamic traffic simulations. *Transportation Science*, **45**(4), 541–561.

Frederix, R., Viti, F., Corthout, R., and Tampère, C. M. (2011). New gradient approximation method for dynamic origin-destination matrix estimation on congested networks. *Transportation Research Record*, **2263**, 19–25.

Frederix, R., Viti, F., Himpe, W. W., and Tampère, C. M. (2014). Dynamic origin-destination matrix estimation on large-scale congested networks using a hierarchical decomposition scheme. *Journal of Intelligent Transportation Systems*, **18**(1), 51–66.

Frejinger, E., Bierlaire, M., and Ben-Akiva, M. (2009). Sampling of alternatives for route choice modeling. *Transportation Research Part B*, **43**(10), 984–994.

Ge, Q. and Menendez, M. (2014). An efficient sensitivity analysis approach for computationally expensive microscopic traffic simulation models. *International Journal of Transportation*, **2**(2), 49–64.

Ge, Q. and Menendez, M. (2016). Global sensitivity analysis of traffic simulation models with dependent input variables. In *Transportation Research Board 95th Annual Meeting*, Washington DC, USA.

Ge, Q., Ciuffo, B., and Menendez, M. (2014). An exploratory study of two efficient approaches for the sensitivity analysis of computationally expensive traffic simulation models. *IEEE Transactions on Intelligent Transportation Systems*, **15**(3), 1288–1297.

Ghali, M. O. and Smith, M. J. (1995). A model for the dynamic system optimum traffic assignment problem. *Transportation Research Part B*, **29**(3), 155–170.

Gomes, G., May, A., and Horowitz, R. (2004). Congested freeway microsimulation model using VISSIM. *Transportation Research Record*, **1876**, 71–81.

Hazelton, M. L. (2008). Statistical inference for time varying origin-destination matrices. *Transportation Research Part B*, **42**(6), 542–552.

Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., and Bayen, A. M. (2010). Evaluation of traffic data obtained via GPS-enabled mobile phones. *Transportation Research Part C*, **18**(4), 568–583.

Holland, J. H. (1975). *Adaption in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI.

Horni, A. and Nagel, K. (2016). More about configuring MATSim. *The Multi-Agent Transport Simulation MATSim, Ubiquity, London.*, pages 35–44. DOI: http://dx.doi.org/10.5334/baw.4 .

Horni, A., Nagel, K., and Axhausen, K. W. (2016). Introducing MATSim. *The Multi-Agent Transport Simulation MATSim, Ubiquity, London.*, pages 3–8. DOI: http://dx.doi.org/10.5334/baw .

Hou, T., Mahmassani, H. S., Alfelor, R. M., Kim, J., and Saberi, M. (2013). Calibration of traffic flow models under adverse weather and application in mesoscopic network simulation. *Transportation Research Record*, **2391**, 92–104.

Hourdakis, J., Michalopoulos, P., and Kottommannil, J. (2003). Practical procedure for calibrating microscopic traffic simulation models. *Transportation Research Record*, **1852**, 130–139.

Huang, E. (2010). *Algorithmic and implementation aspects of on-line calibration of dynamic traffic assignment*. Master's thesis, Massachusetts Institute of Technology.

Huyer, W. and Neumaier, A. (2008). SNOBFIT-stable noisy optimization by branch and fit. *ACM Transactions on Mathematical Software (TOMS)*, **35**(2), 9.

Iqbal, M. S., Choudhury, C. F., Wang, P., and Gonzalez, M. C. (2014). Development of origin-destination matrices using mobile phone call data. *Transportation Research Part C*, **40**, 63–74.

Jha, M., Gopalan, G., Garms, A., Mahanti, B. P., Toledo, T., and Ben-Akiva, M. E. (2004). Development and calibration of a large-scale microscopic traffic simulation model. *Transportation Research Record*, **1876**, 121–131.

Kattan, L. and Abdulhai, B. (2006). Noniterative approach to dynamic traffic origin-destination estimation with parallel evolutionary algorithms. *Transportation Research Record*, **1964**, 201–210.

Kim, H., Baek, S., and Lim, Y. (2001). Origin-destination matrices estimated with a genetic algorithm from link traffic counts. *Transportation Research Record*, **1771**, 156–163.

Kunde, K. K. (2002). *Calibration of mesoscopic traffic simulation models for dynamic traffic assignment*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Leclercq, L. (2005). Calibration of flow-density relationships on urban streets. *Transportation Research Record*, **1934**, 226–234.

Lee, G., You, S. I., Ritchie, S. G., Saphores, J. D., Sangkapichai, M., and Jayakrishnan, R. (2009). Environmental impacts of a major freight corridor. *Transportation Research Record*, **2123**(1), 119–128.

Lee, J. B. and Ozbay, K. (2009). New calibration methodology for microscopic traffic simulation using enhanced simultaneous perturbation stochastic approximation approach. *Transportation Research Record*, **2124**, 233–240.

Little, J. D. C. (1961). A proof for the queuing formula: L= λ W. *Operations Research*, **9**(3), 383–387.

Little, J. D. C. (2011). Little's law as viewed on its 50th anniversary. *Operations Research*, **59**(3), 536–549.

Lu, C. C., Zhou, X., and Zhang, K. (2013). Dynamic origin-destination demand flow estimation under congested traffic conditions. *Transportation Research Part C*, **34**, 16–37.

Lu, L., Xu, Y., Antoniou, C., and Ben-Akiva, M. (2015). An enhanced SPSA algorithm for the calibration of dynamic traffic assignment models. *Transportation Research Part C*, **51**, 149–166.

Milkovits, M., Huang, E., Antoniou, C., Ben-Akiva, M., and Lopes, J. (2010). DynaMIT 2.0: The next generation real-time dynamic traffic assignment system. In *Advances in System Simulation (SIMUL), 2010 Second International Conference on*, pages 45–51. IEEE.

Miller, H. J. and Shaw, S.-L. (2001). *Geographic information systems for transportation*. Oxford University Press.

Mimbela, L. E. Y. and Klein, L. A. (2000). *A summary of vehicle detection and surveillance technologies used in intelligent transportation systems*. U.S. Federal Highway Administration, Intelligent Transportation Systems. Available at: http://www.fhwa.dot.gov/ohim/tvtw/vdstits.pdf .

Mishalani, R. G., Coifman, B., and Gopalakrishna, D. (2002). Evaluating real-time origin-destination flow estimation using remote sensing-based surveillance data. In *Applications of Advanced Technologies in Transportation*, pages 640–647.

Nagel, K. and Flötteröd, G. (2012). Agent-based traffic assignment: going from trips to behavioral travelers. In R. M. Pendyala and C. Bhat, editors, *Travel Behaviour Research in an Evolving World*, chapter 12, pages 261–293. Emerald Group Publishing, Bingley, United Kingdom.

Newell, G. (1993). A simplified theory of kinematic waves in highway traffic, part I: general theory. *Transportation Research Part B*, **27**(4), 281–287.

Nie, Y. (2006). *Variational inequality approach for inferring dynamic origin-destination travel demand*. Ph.D. thesis, University of California, Davis.

Nigro, M., Cipriani, E., and del Giudice, A. (2018). Exploiting floating car data for time-dependent origin-destination matrices estimation. *Journal of Inteligent Transportation Systems*, **22**(2), 159–174.

Osorio, C. (2010). *Mitigating network congestion: analytical models, optimization methods and their applications*. Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne.

Osorio, C. (2017). High-dimensional offline OD calibration for stochastic traffic simulators of large-scale urban networks. *Technical report, Massachusetts Institute of Technology*. Under review. Available at: http://web.mit.edu/osorioc/www/papers/osoODCalib.pdf .

Osorio, C. and Atastoy, B. (2017). Efficient simulation-based toll optimization for large-scale networks. *Technical report, Massachusetts Institute of Technology*. Under review. Available at: http://web.mit.edu/osorioc/www/papers/osoAtaTollSO.pdf .

Osorio, C. and Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems. *Operations Research*, **61**(6), 1333–1345.

Osorio, C. and Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation. *Transportation Science*, **49**(3), 623–636.

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

Osorio, C. and Nanduri, K. (2015a). Energy-efficient urban traffic management: a microscopic simulation-based approach. *Transportation Science*, **49**(3), 637–651.

Osorio, C. and Nanduri, K. (2015b). Urban transportation emissions mitigation: coupling high-resolution vehicular emissions and traffic models for traffic signal optimization. *Transportation Research Part B*, **81**, 520–538.

Osorio, C. and Selvam, K. (2017). Simulation-based optimization: achieving computational efficiency through the use of multiple simulators. *Transportation Science*, **51**(2), 395–411.

Prakash, A. A., Seshadri, R., Antoniou, C., Pereira, F. C., and Ben-Akiva, M. E. (2017). Reducing the dimension of online calibration in dynamic traffic assignment systems. *Transportation Research Record*, **2667**, 96–107.

PTV (2008). *VISSIM 4.30 User's Manual*. Planung Transport Verkehr AG, Karlsruhe, Germany.

Qian, Z. and Zhang, H. M. (2011). Computing individual path marginal cost in networks with queue spillbacks. *Transportation Research Record*, **2263**, 9–18.

Qin, X. and Mahmassani, H. S. (2004). Adaptive calibration of dynamic speed-density relations for online network traffic estimation and prediction applications. *Transportation Research Record*, **1876**, 82–89.

Shao, H., Lam, W. H. K., Sumalee, A., and Hazelton, M. L. (2015). Estimation of mean and covariance of stochastic multi-class OD demands from classified traffic counts. *Transportation Research Part C*, **59**, 92–110.

Shen, W., Nie, Y., and Zhang, H. M. (2007). On path marginal cost analysis and its relation to dynamic system-optimal traffic assignment. *Transportation and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17*.

Søndergaard, J. (2003). *Optimization using surrogate models - by the Space Mapping technique*. Ph.D. thesis, Technical University of Denmark.

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, **37**(3), 332–341.

Stathopoulos, A. and Tsekeris, T. (2004). Hybrid meta-heuristic algorithm for the simultaneous optimization of the O-D trip matrix estimation. *Computer-Aided Civil and Infrastructure Engineering*, **19**(6), 421–435.

Sun, J. and Feng, Y. (2011). A novel OD estimation method based on automatic vehicle identification data. In *Intelligent Computing and Information Science*, pages 461–470, Berlin, Heidelberg. Springer.

Tavana, H. (2001). *Internally-consistent estimation of dynamic network origin-destination flows from intelligent transportation systems data using bi-level optimization*. Ph.D. thesis, University of Texas at Austin.

Treiber, M. and Kesting, A. (2013). Microscopic calibration and validation of car-following models - a systematic approach. *Procedia: Social and Behavioral Sciences. Paper selected for the 20th International Symposium on Transportation and Traffic Theory (ISTTT 2013)*, **80**, 922–939.

Tympakianaki, A. (2018). *Demand estimation and bottleneck management using heterogeneous traffic data.* Ph.D. thesis, KTH Royal Institute of Technology.

Tympakianaki, A., Koutsopoulos, H., and Jenelius, E. (2015). c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin-destination matrix estimation. *Transportation Research Part C*, **55**, 231–245.

Van Der Zijpp, N. and De Romph, E. (1997). A dynamic traffic forecasting application on the Amsterdam beltway. *International Journal of Forecasting*, **13**(1), 87–103.

Vaze, V., Antoniou, C., Wen, Y., and Ben-Akiva, M. (2009). Calibration of dynamic traffic assignment models with point-to-point traffic surveillance. *Transportation Research Record*, **2090**, 1–9.

Verbas, I. O., Mahmassani, H. S., and Zhang, K. (2011). Time-dependent origin-destination demand estimation: challenges and methods for large-scale networks with multiple vehicle classes. *Transportation Research Record*, **2263**, 45–56.

Yamamoto, T., Miwa, T., Takeshita, T., and Morikawa, T. (2009). Updating dynamic origin-destination matrices using observed link travel speed by probe vehicles. In *Proceedings of the 18th International Symposium on Transportation and Traffic Theory*, pages 723–738, Boston, MA.

Yang, H. (1995). Heuristic algorithms for the bilevel origin-destination matrix estimation problem. *Transportation Research Part B*, **29B**, 231–242.

Yang, H., Sasaki, T., and Ilda, Y. (1992). Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research Part B*, **26B**, 417–434.

Zhang, C. and Osorio, C. (2018). Efficient offline calibration of origin-destination (OD) demand for large-scale stochastic traffic models. *Technical report, Massachusetts Institute of Technology*. Under review. Available at: http://web.mit.edu/osorioc/www/papers/zhaOsoODcalib.pdf .

Zhang, C., Osorio, C., and Flötteröd, G. (2016a). An efficient algorithm for the supply calibration of large-scale stochastic traffic simulators. In *Proceedings of the Symposium of the European Association for Research in Transportation (hEART)*.

Zhang, C., Osorio, C., and Flötteröd, G. (2016b). Supplementary technical report to the manuscript: Efficient calibration techniques for large-scale traffic simulators. Technical report, Massachusetts Institute of Technology. Available at: http://web.mit.edu/osorioc/www/papers/zhaOsoFlo16_techRep.pdf .

Zhang, C., Osorio, C., and Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B*, **97**, 214–239.

Zhang, H. M., Nie, Y., and Qian, Z. (2008). Estimating time-dependent freeway origin-destination demands with different data coverage sensitivity analysis. *Transportation Research Record*, **2047**, 91–99.

Zhong, R. X., Fu, K. Y., Sumalee, A., Ngoduy, D., and Lam, W. H. K. (2016). A cross-entropy method and probabilistic sensitivity analysis framework for calibrating microscopic traffic models. *Transportation Research Part C*, **63**, 147–169.

Zhou, X. and Mahmassani, H. S. (2006). Dynamic origin-destination demand estimation using automatic vehicle identification data. *IEEE Transactions on Intelligent Transportation Systems*, **7**(1), 105–114.

Zhou, X. and Mahmassani, H. S. (2007). A structural state space model for real-time traffic origin-destination demand estimation and prediction in a day-to-day learning framework. *Transportation Research Part B*, **41**, 823–840.

Zhou, X., Qin, X., and Mahmassani, H. S. (2003). Dynamic origin-destination demand

estimation with multiday link traffic counts for planning applications. *Transportation Research Record*, **1831**, 30–38.

Zhou, X., Lu, C., and Zhang, K. (2012). Dynamic origin-destination demand flow estimation utilizing heterogeneous data sources under congested traffic conditions. In *91st Transportation Research Board Annual Meeting*, Washington DC, USA.

Ziemke, D. (2013). *Demand generation for multi-agent transport simulations based on an econometric travel behavior model and a traffic-count-based calibration algorithm*. Master's thesis, Technische Universität Berlin.

Ziemke, D., Nagel, K., and Bhat, C. (2015). Integrating CEMDAP and MATSim to increase the transferability of transport demand models. *Transportation Research Record*, **2493**, 117–125.