

Efficient calibration techniques for large-scale traffic simulators

WORKING PAPER

Carolina Osorio^{*a}, Chao Zhang^{†a}, and Gunnar Flötteröd^{‡b}

^aDepartment of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

^bDepartment of Transport Science, KTH Royal Institute of Technology, 11428 Stockholm, Sweden

Abstract

Transportation simulators have been increasingly popular and widely applied to analyze transportation problems that are not amenable to study by other means. Model calibration is a crucial prerequisite for transportation simulators to reliably reproduce and predict traffic conditions. This paper considers the calibration of stochastic transportation simulators that provide detailed representations of actual traffic conditions and thus are computationally costly to evaluate. It formulates the calibration problem as a simulation-based optimization (SO) problem addressed by the metamodel approach, in which the metamodel combines information from the simulator with information from an analytical differentiable and tractable network model that relates the calibration parameters to the simulation-based objective function. We have evaluated the performance of the proposed framework on a toy network and a large-scale metropolitan network with 24335 links and 11345 nodes in Berlin, Germany. The results show 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%.

1 Introduction

This paper focuses on the calibration (i.e., the estimation of the input parameters) of simulation-based road traffic models. The problem of model calibration has been extensively studied by the transportation community. A survey of both analytical and simulation-based calibration problems and algorithms is given in Balakrishna (2006). The most extensively studied calibration problem is, arguably, that of the calibration of origin-destination (OD) matrices, with seminal work such as Cascetta and Nguyen (1988), Cascetta *et al.* (1993) and more recent work such as Zhou and Mahmassani (2007, 2006), Zhou (2004), Ashok and Ben-Akiva (2002), Ashok (1996), as well as the

*osorioc@mit.edu

†chaoz@mit.edu, **corresponding author**

‡gunnar.floetteroed@abe.kth.se

deployment of path flow estimators (PFEs), with the seminal work of Bell *et al.* (1997). Recent PFE reviews include Flötteröd (2008) and Buisson *et al.* (2012).

This paper 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, (iv) the desire to evaluate performance under equilibrium conditions, which requires running sequentially multiple simulation-based assignment iterations (Nagel and Flötteröd, 2012).

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 a low computational cost. That is, algorithms that can identify good solutions within few algorithmic iterations, i.e., with few simulated (near-)equilibrium responses.

We first discuss the main challenges of addressing calibration problems. We then state how the approach proposed in this paper addresses these challenges. The traffic model is often stochastic. It 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. A review of state-of-the-art simulation models is presented in Barceló (2010). Thus, the mapping between the calibration parameters and the probability distribution of a given performance metric (e.g., the objective function of a calibration problem) is an intricate function.

This mapping is often non-convex and may contain multiple local minima. 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 approximated). The use of traditional statistical and numerical algorithms is of limited use for calibration problems, since the underlying simulators often lack the strong assumptions required by these methods (e.g., normality, ergodicity) (Buisson *et al.*, 2012). Hence, the traditional approach to calibration has been the use of simulation-based optimization (SO) algorithms. Most SO algorithms are general-purpose algorithms, they are not tailored to the complexities of transportation simulators. They have been used extensively for calibration, several reviews include Ben-Akiva *et al.* (2012), Balakrishna (2006), Antoniou (2004). Algorithms frequently used for calibration include simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992), genetic algorithms (GA) (Holland, 1975), particle filters and Kalman filters.

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 few algorithmic iterations, i.e., they are not computationally efficient. Yet they are used by the transportation community under tight computational budgets, i.e., under small-sample size conditions. One approach to derive computationally efficient algorithms, is to exploit the structure of the underlying calibration problem. General-purpose SO algorithms exploit limited problem-structure (e.g., at most they are based on numerical linearizations). 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.

Traditional SO algorithms, although designed to guarantee asymptotic properties, are typically used for calibration under tight computational budgets. This makes them sensitive to the initial

points. Given the difficulty of identifying good initial points to the calibration problem, there is a need for algorithms that perform well under tight computational budgets while being robust to the quality of the initial points. Additionally, the lack of computationally efficient calibration algorithms has led recent calibration research to focus on the design of dimensionality reduction methods (e.g., sensitivity analysis methods) (Ge *et al.*, 2016, Ge and Menendez, 2016, 2014, Ciuffo and Azevedo, 2014).

This paper addresses the following challenges.

Computational efficiency We propose an algorithm that can identify points with good performance within few algorithmic iterations. Therefore, it is a computationally efficient algorithm that reflects well the computational conditions under which calibration problems are addressed by both the transportation research and practice communities. This is achieved by designing an algorithm that exploits the transportation-specific structure of the calibration problem.

Analytical structural information The algorithm embeds an analytical approximation of the simulator. This contributes to a largely unresolved methodological challenge which is the formulation of tractable measurement equations that link available surveillance field data to the simulator’s calibration parameters.

Robustness to initial conditions The algorithm can identify good solutions within few iterations regardless of the initial points. It is robust to the quality of the initial points.

Stochasticity The algorithm is a simulation-based optimization algorithm that accounts for the simulator’s stochasticity.

This paper formulates the calibration problem as a simulation-based optimization problem and uses a metamodel SO algorithm. The paper formulates a novel metamodel suitable for demand calibration problems. The algorithm is used to address a calibration problem for a large-scale Berlin metropolitan network. Section 2 presents the proposed methodology, followed by case studies on both a toy network and a Berlin metropolitan network (Section 3). We conclude with a brief discussion (Section 4) and present additional details in the Appendix.

With the increase in the availability, diversity and quality of travel data, comes an 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). The formulation of the proposed framework for the calibration of supply parameters has been recently derived (Zhang *et al.*, 2016a). Thus, we expect the general ideas presented in this paper to be suitable for the joint calibration of demand and supply parameters.

This paper illustrates the use of the proposed algorithm with the calibration of a single demand parameter. Hence, the results do not indicate the suitability of the approach for high-dimensional problems. Nonetheless, the metamodel ideas proposed in this paper have shown to be scalable for traffic management problems (e.g., Osorio and Chong, 2015, Chen *et al.*, 2012). We expect these ideas to also scale well for calibration problems.

The case studies show that the proposed algorithm identifies good solutions within a couple of iterations. Hence, it can also be used as a technique to identify good initial solutions to launch or initialize traditional general-purpose calibration algorithms. This is particularly important when using the general-purpose algorithms under tight computational budgets, which are by design sensitive to the initial conditions.

2 Methodology

Section 2.1 formulates the calibration problem of interest. We focus on the use of a family of optimization algorithms known as metamodel simulation-based optimization (SO) algorithms. Section 2.2 briefly describes the main ideas of metamodel SO algorithms. Section 2.3 formulates the calibration problem as a metamodel SO problem. It then formulates a suitable metamodel.

2.1 Formal problem statement

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 s and i index the simulation time bins of size 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 $\Delta_{nis} \in \{0, 1\}$ the stochastic binary indicator of traveler n crossing link i in time step s , the stochastic simulated link flow rate F_{is} on link i in time step s is

$$F_{is} = \frac{1}{T} \sum_{n=1}^N \Delta_{nis} \quad (1)$$

with the expectation

$$E[F_{is}] = \frac{1}{T} \sum_{n=1}^N \delta_{nis} \quad (2)$$

where $\delta_{nis} = E[\Delta_{nis}]$ is the probability that traveler n crosses link i in time step s (Flötteröd and Bierlaire, 2009). This probability is in turn dependent on the network conditions x , in particular travel times of the various routes, which in turn depend on the travel behavior (e.g., departure time, rout choice), requiring typically to solve (2) iteratively. This is subsequently expressed by writing $\delta_{nis} = \delta_{nis}(\theta; x)$ with θ a parameter vector of the underlying behavioral 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 x , followed by a simulation of the corresponding vehicle flows through the network, which in turn yields updated network conditions.

Let $y_{i,s}$ be the number of vehicles counted in the field on link i in time step s . A traditional non-linear least squares formulation of the calibration problem is then to minimize the following objective function:

$$f(\theta) = \min_{\theta \in \Theta} \sum_{i \in \mathcal{I}} \sum_s (y_{i,s} - E[F_{i,s}(\theta; x)])^2. \quad (3)$$

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

Formulation (3) illustrates the main challenges of the calibration problem described in Section 1. The function $E[F_{i,s}(\theta; x)]$ has no closed-form expression available, since x (e.g., the travel times) and $E[F_{i,s}(\theta; x)]$ (i.e., the expected link flows) can only be estimated by evaluating the stochastic simulation model. Thus, Problem (3) cannot be solved using traditional analytical and deterministic optimization approaches. The function $E[F_{i,s}(\theta; x)]$ is a non-linear function that describes complex

spatial-temporal traffic phenomena in the stochastic traffic simulator and lacks sound mathematical properties such as convexity.

The purpose of this paper 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 $E[F_{i,s}(\theta; x)]$. 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 complex intersections).

2.2 Metamodel simulation-based optimization methods

The following summary of ideas developed by Osorio and Bierlaire (2013) is needed to make the proposed calibration approach concrete. The broad family of SO problems considered can be formulated as follows.

$$\min_{\theta \in \Theta} f(\theta, z; \hat{q}) \quad (4)$$

Problem (4) consists of two components: a simulation-based objective function f , and a feasible region Θ . The objective function f is not available in closed-form, it can only be estimated via simulation. It depends on: the calibration parameters θ , endogenous simulation variables z (e.g., link flows, travel times) and exogenous simulation parameters \hat{q} (e.g., network topology). The feasible region Θ 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 (4) by iteratively solving a set of analytical problems. At iteration k the SO algorithm solves an analytical problem with the following form.

$$\min_{\theta \in \Theta} m_k(\theta; \beta_k) \quad (5)$$

The main idea is to replace the simulation-based objective function (f in (4)) with an analytical approximation of it (m_k in (5)). The function m_k is known as the metamodel. It is often a parametric function, with the iteration-specific parameter vector, β_k , often being fitted based on simulated observations. Metamodel SO techniques iterate over two main steps. First, the metamodel is constructed based on the sample of simulated observations. Second, it is used to perform optimization and derive a trial point (e.g., a calibration parameter value). The performance of the trial point can be evaluated by the simulator, which leads to new simulation observations. As new observations become available the accuracy of the metamodel can be improved (Step 1), leading ultimately to better trial points (Step 2). In this paper, we use the derivative-free trust region algorithm of Conn *et al.* (2009). A full description of the algorithm is provided in Section 4.2 of Osorio and Bierlaire (2013). In order to use this framework for calibration, the main challenge is to formulate a metamodel model that: (i) provides a good approximation of the mapping of the decision vector to the objective function, and (ii) is also computationally efficient such that Problem (5), which needs to be solved at every iteration of the algorithm, can be solved efficiently. This paper contributes by formulating a metamodel that achieves these two goals.

Metamodels can be classified as either:

- (i) functional models, these are general-purpose functions suitable to approximate an arbitrary function f (e.g., polynomials);
- (ii) physical models, these are problem-specific functions. In other words, their functional form depends on the specific problem.

We use a metamodel that combines ideas from functional models and from physical models. At iteration k it is defined by:

$$m_k(\theta; \beta_k) = \beta_{k,0} f_A(\theta) + \phi(\theta; \beta_k) \quad (6)$$

Equation (6) defines m_k as a linear combination of a general-purpose parametric function (e.g., a polynomial) denoted ϕ and a problem-specific approximation of f (defined in (4)) denoted f_A . The first element of the vector β_k is denoted $\beta_{k,0}$. Based on (6), 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 traffic model. Hence, the analytical problem solved at every iteration is defined by:

$$\min_{\theta \in \Theta} m_k(\theta; \beta_k) \quad (7)$$

$$h(\theta, \tilde{v}; \tilde{q}) = 0, \quad (8)$$

where h denotes the analytical macroscopic traffic model, with endogenous variables θ and \tilde{v} (e.g., expected link queue-lengths) and exogenous parameters \tilde{q} (e.g., network topology).

This problem differs from Problem (5) in the presence of an additional set of constraints (8). This set of constraints represents the analytical macroscopic traffic 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 traffic model (h (8)) 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 traffic model, such that large-scale networks can be considered.
- (iii) It is computationally efficient to solve. Every iteration of the SO algorithm requires solving Problem (7)-(8), which contains the analytical traffic model as a set of constraints. Hence, the evaluation of the traffic model needs to be computationally inexpensive to evaluate.

The broad family of metamodels defined by (6) have been used to efficiently address large-scale urban traffic management problems while using inefficient yet detailed stochastic microscopic simulators (Osorio and Selvam, 2016, Osorio and Chong, 2015, Osorio and Nanduri, 2015a,b, Chen *et al.*, 2012).

2.3 Metamodel formulation

The novel methodology proposed in this paper 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. The specific SO problem is then formulated as:

$$f(\theta) = \min_{\theta} \sum_{i \in \mathcal{I}} (y_i - E[F_i(\theta; x)])^2 \quad (9)$$

$$\theta_L \leq \theta \leq \theta_U, \quad (10)$$

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

$$\min_{\theta} \sum_{i \in \mathcal{I}} (y_i - m_{i,k}(\theta; \beta_{i,k}))^2 \quad (11)$$

$$\theta_L \leq \theta \leq \theta_U \quad (12)$$

$$h(\theta, \tilde{v}; \tilde{q}) = 0. \quad (13)$$

where h denotes the analytical traffic model used to analytically approximate the link flows.

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

$$m_{i,k}(\theta; \beta_{i,k}) = \beta_{i,k,0} \lambda_i(\theta) + \beta_{i,k,1} + \beta_{i,k,2} \theta, \quad (14)$$

where θ is the (scalar) behavioral parameter, $\lambda_i(\theta)$ is the expected flow for link i approximated by the analytic traffic 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 $\beta_{i,k,1} + \beta_{i,k,2}\theta$). Flötteröd *et al.* (2011) formulate a related calibration problem using a metamodel of the form (14) with $\beta_{i,k,0}$ set to zero.

We now present the analytical traffic model that will yield the analytical approximation of link flows ($\lambda_i(\theta)$). 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 of link i ;
- μ_i service rate of queue i ;
- γ_i external arrival rate of queue i ;
- λ_i arrival rate of queue i ;
- \tilde{t}_i expected travel time of 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 of route r ;
- f_r expected flow on route r ;
- \tilde{p}_{sr} probability that a traveler of 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 of route r .

$$\left\{ \begin{array}{ll} f_r = \sum_{s \in \mathcal{S}} d_s \tilde{p}_{sr} & \forall r \in \mathcal{R} \quad (15a) \\ \tilde{p}_{sr} = \frac{e^{-\theta t_r}}{\sum_{j \in \mathcal{R}_s} e^{-\theta t_j}} & \forall s \in \mathcal{S}, \forall r \in \mathcal{R}_s \quad (15b) \\ t_r = \sum_{i \in \Psi_r} \tilde{t}_i & \forall r \in \mathcal{R} \quad (15c) \\ \tilde{t}_i = \frac{l_i}{v_i} + \frac{\tilde{n}_i}{\lambda_i} & \forall i \in \mathcal{Q} \quad (15d) \\ \tilde{n}_i = \frac{\tilde{\rho}_i}{1 - \tilde{\rho}_i} - \frac{(\ell_i + 1) \tilde{\rho}_i^{\ell_i + 1}}{1 - \tilde{\rho}_i^{\ell_i + 1}} & \forall i \in \mathcal{Q} \quad (15e) \\ \tilde{\rho}_i = \frac{\lambda_i}{\mu_i} & \forall i \in \mathcal{Q} \quad (15f) \\ \lambda_i = \gamma_i + \sum_{j \in \mathcal{Q}} p_{ji} \lambda_j & \forall i \in \mathcal{Q} \quad (15g) \\ p_{ij} = \frac{\sum_{r \in \mathcal{G}_{ij}} f_r}{\sum_{w \in \mathcal{H}_i} f_w} & \forall i \in \mathcal{Q}, \forall j \in \mathcal{Q} \quad (15h) \\ \gamma_i = \sum_{r \in \mathcal{T}_i} f_r & \forall i \in \mathcal{Q} \quad (15i) \end{array} \right.$$

Equation (15a) describes how expected OD demand is mapped to expected route demand through the route choice model. In other words, it defines the expected route flow as a weighted sum of expected OD flows. Equation (15b) is the route choice model. It is a multinomial logit model with a single attribute: expected route travel time. This is a simplification of the route choice model used in the simulation models of the case studies of Section 3. For a detailed description of the route choice models of the simulator, see Zhang *et al.* (2016b). Equation (15c) defines the expected route travel time as the sum of expected link travel times.

Equation (15d) 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 spillback and that all link outflow constraints result from the links 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 spillbacks. 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 the use of 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 (15e) approximates the expected queue-length of a given queue. It is obtained by considering the queue as an isolated M/M/1/ ℓ queue, with finite space capacity ℓ . A derivation of this expression can be found in Appendix A of Osorio and Chong (2015). Equation (15f) defines $\tilde{\rho}_i$, which is known as the traffic intensity of the queue, as the ratio of expected demand to expected supply. Equation (15g) is a flow conservation equation that relates the expected flow on queue i (denoted λ_i) to the sum of expected external flow to queue i (denoted γ_i) and expected demand arising from upstream queues. The expected flow, 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 p_{ij}) is defined by Equation (15h) as the ratio of the expected demand from i to j and the expected demand of queue i . The expected external demand for queue i is defined in Equation (15i) as the sum of the expected demand over all routes that start at queue i .

For the simulation model of the case studies, the set of route alternatives for a given OD pair is endogenous (i.e., it varies with θ and across assignment iterations). Nonetheless, for the purpose of tractability a fixed (i.e., exogenous) route choice 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 Zhang *et al.* (2016b). The use of an exogenous route choice set for the analytical model implies that the polynomial component of the metamodel has to capture simulated congestion-dependent route choice changes. The use of an endogenous, iteration-dependent, route choice set could yield more accurate analytical results, at the cost of a higher computational burden.

In summary, the analytical traffic model used to approximate the expected link flows $\lambda_i(\theta)$ of Equation (14) are obtained by evaluating the system of nonlinear differentiable equations (15). The exogenous parameters of the above system of equations are: $\theta, d_s, \mu_i, \ell_i, l_i, v_i, \mathcal{S}, \mathcal{Q}, \mathcal{R}, \mathcal{R}_s, \mathcal{G}_{ij}, \mathcal{H}_i, \mathcal{T}_i, \Psi_r$. All other variables are endogenous variables that are obtained when solving the above system of equations.

3 Case studies

3.1 Experimental design

We apply the proposed approach to 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 traffic model. For each case study, we compare the performance of two calibration approaches that only differ in the use or not of the analytical traffic model. All other algorithmic details are identical. The first approach is the proposed approach (denoted algorithm Am). It uses the metamodel defined by (14). The second approach (denoted algorithm $A\phi$) considers a metamodel defined for iteration k and link i as:

$$\phi_{i,k}(\theta; \beta_{i,k}) = \beta_{i,k,1} + \beta_{i,k,2}\theta. \tag{16}$$

This metamodel differs from that of (14) in the absence of the macroscopic traffic 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 summarised in Table 1. The main details of the experimental design for each network are displayed in Table 2. The first row of Table 2 considers the lower and upper bounds for θ , this defines the feasible region, Θ . For the toy network, we consider a set of hypothetical 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 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 that of the 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 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

	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 1: Network attributes

	Toy network	Berlin network
Bounds for θ [θ_L, θ_U] (1/hr)	$[-60, 0]$	$[-60, 0]$
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 evaluations per algorithmic run	7500	20000

Table 2: Experimental design

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 stochasticity 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 (Equation (9)) 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 $E[F_i(\theta; x)]$ 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 evaluations (or runs) 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 evaluations: the performance of 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 $30 \times 5 \times 50 = 7500$ simulation runs. Similarly, for the Berlin network each SO run involves $20 \times 10 \times 100 = 20000$ simulation runs.

Table 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.

3.2 Toy network

3.2.1 Network attributes

For the hypothetical toy network, we pick a set of true θ values, θ^* (row 2 of Table 2). We use these values to generate the “real” traffic counts via simulation. The topology of the network is displayed in Figure 1. Each link consists of a single lane road. Table 4 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 second 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.

	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: Simulation runtime statistics in minutes

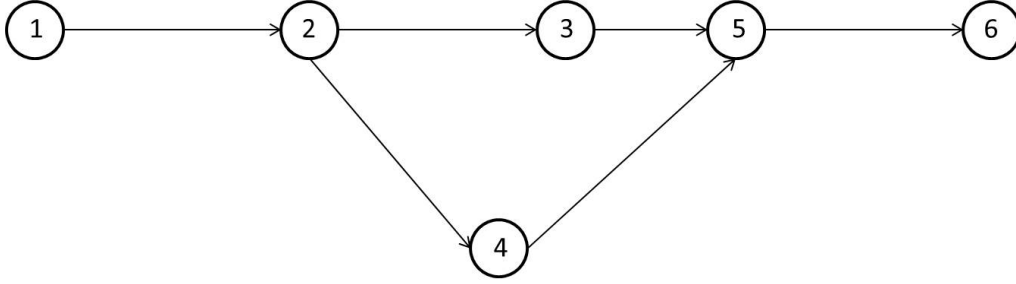


Figure 1: Toy network topology

Link index	Nodes connected	Length (km)	Maximum speed (km/h)	Signalized
1	1 → 2	2.5	72	un-signalized
2	2 → 3	7.5	54	signalized
3	3 → 5	2.5	54	un-signalized
4	2 → 4	7.071	72	un-signalized
5	4 → 5	7.071	72	un-signalized
6	5 → 6	2	72	un-signalized

Table 4: Toy network link properties

3.2.2 Results

We first consider the experiment with $\theta^* = -5\text{hr}^{-1}$. Figure 2 displays the simulation-based objective function and the objective function approximation provided by the analytical traffic model (Equations (15)). The simulation-based objective function is defined by Equation (9). 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:

$$\sum_{i \in \mathcal{I}} (y_i - \lambda_i(\theta))^2. \quad (17)$$

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 as an excellent approximation of the simulation-based objective function.

Figure 2 also displays a green range for θ . This range is a set of θ values that have statistically equivalent objective function values than 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, the alternative hypothesis assumes unequal expectations. For this experiment, the equivalence region is $[-6, -4]$ (in units hr^{-1}).

Figures 3-5, each consider a given initial point, θ_0 . The x-axis displays the iteration of the SO algorithm, the y-axis displays the current iterate (i.e., the best θ value found so far by the algorithm). The black solid lines correspond to proposed method, Am , while the red dashed 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 for both methods. These figures also display the aforementioned equivalence region (in green).

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

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

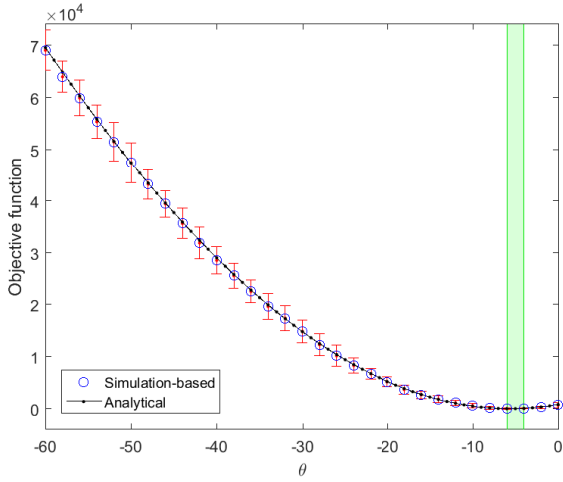


Figure 2: Simulation-based and analytical objective functions for $\theta^* = -5\text{hr}^{-1}$

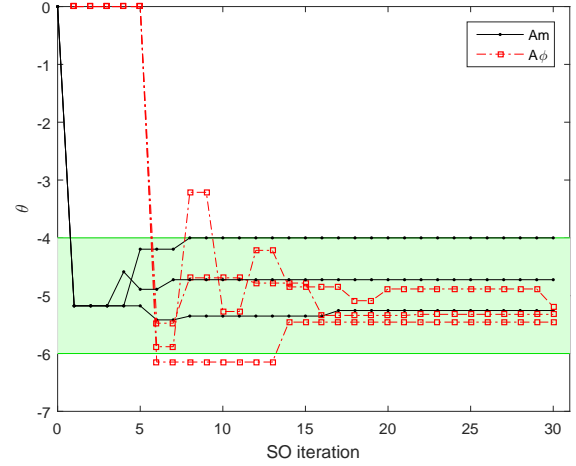


Figure 3: Algorithmic solutions versus iterations, for $\theta^* = -5\text{hr}^{-1}$ and $\theta_0 = 0\text{hr}^{-1}$

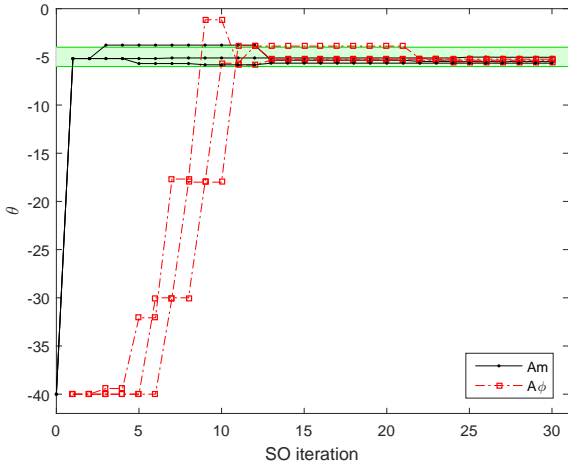


Figure 4: Algorithmic solutions versus iterations, for $\theta^* = -5\text{hr}^{-1}$ and $\theta_0 = -40\text{hr}^{-1}$

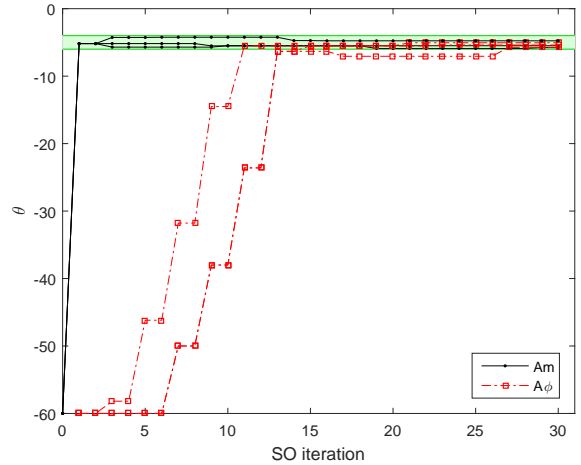


Figure 5: Algorithmic solutions versus iterations, for $\theta^* = -5\text{hr}^{-1}$ and $\theta_0 = -60\text{hr}^{-1}$

- 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 6. Again, the analytical approximation of the objective function provided by the analytical network model (Equations (15)) 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 7-9. Note that in Figure 9, all 3 red curves overlap from iteration 1 until iteration 8. For Figures 7-9, all Am runs converge, while 7 out of 9 $A\phi$ runs converge. Of the 2 $A\phi$ runs that do not converge, they 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 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 11-13.

Note that in Figure 11, 2 of the red curves that overlap from iteration 1 until iteration 11, this

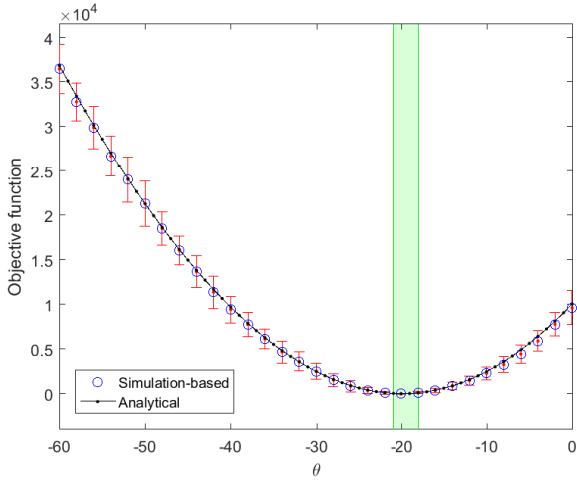


Figure 6: Simulation-based and analytical objective functions for $\theta^* = -20\text{hr}^{-1}$

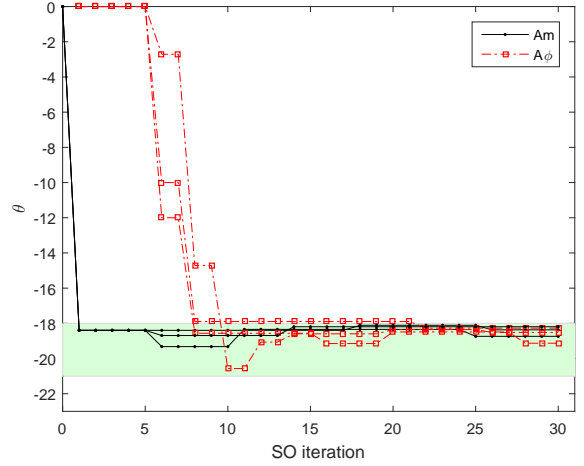


Figure 7: Algorithmic solutions versus iterations, for $\theta^* = -20\text{hr}^{-1}$ and $\theta_0 = 0\text{hr}^{-1}$

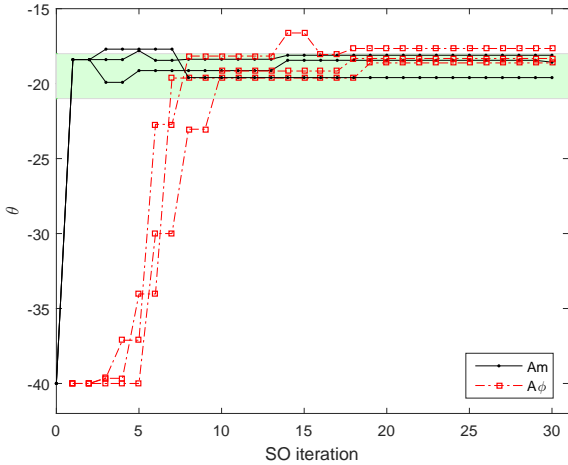


Figure 8: Algorithmic solutions versus iterations, for $\theta^* = -20\text{hr}^{-1}$ and $\theta_0 = -40\text{hr}^{-1}$

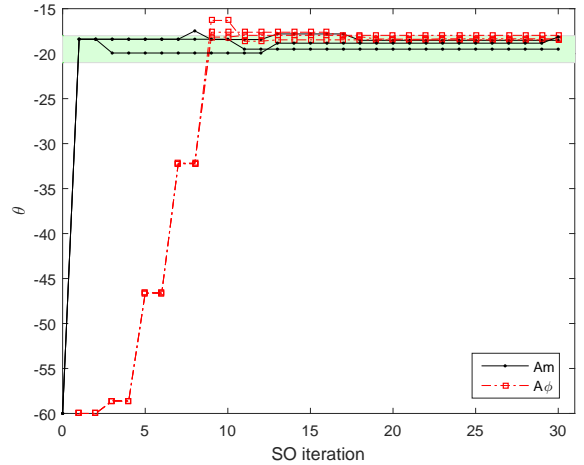


Figure 9: Algorithmic solutions versus iterations, for $\theta^* = -20\text{hr}^{-1}$ and $\theta_0 = -60\text{hr}^{-1}$

occurs for the left-most red curve. As for the previous experiments, all Am runs converge. Eight out of the nine $A\phi$ runs 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 can do so within few algorithmic iterations. Overall the performance of both methods is similar and good. Overall, Am identifies good solutions faster than $A\phi$, and systematically converges.

Let us compare the performance of both methods in terms of their computational efficiency. Table 5 considers all experiments mentioned above. It displays for each experiment and each method, the number of algorithmic iterations until convergence (i.e., the first time the equivalence 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 out 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

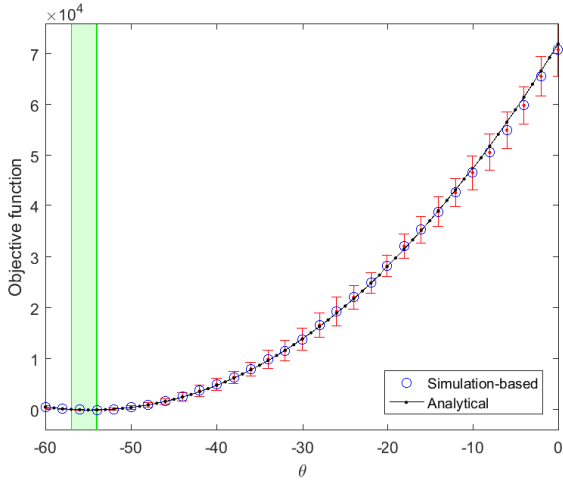


Figure 10: Simulation-based and analytical objective functions for $\theta^* = -55\text{hr}^{-1}$

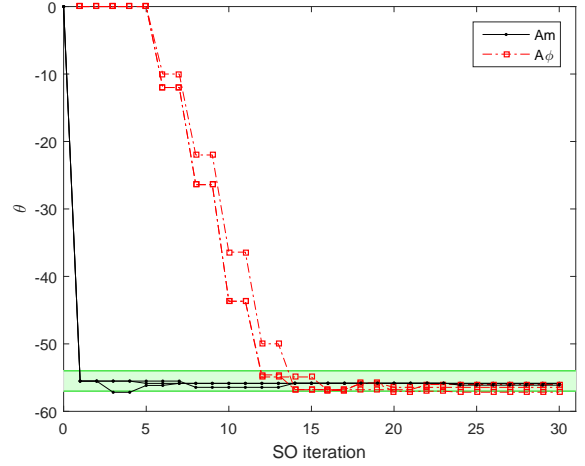


Figure 11: Algorithmic solutions versus iterations, for $\theta^* = -55\text{hr}^{-1}$ and $\theta_0 = 0\text{hr}^{-1}$

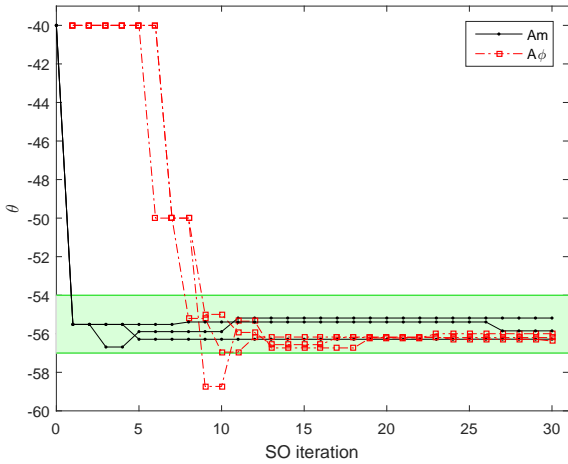


Figure 12: Algorithmic solutions versus iterations, for $\theta^* = -55\text{hr}^{-1}$ and $\theta_0 = -40\text{hr}^{-1}$

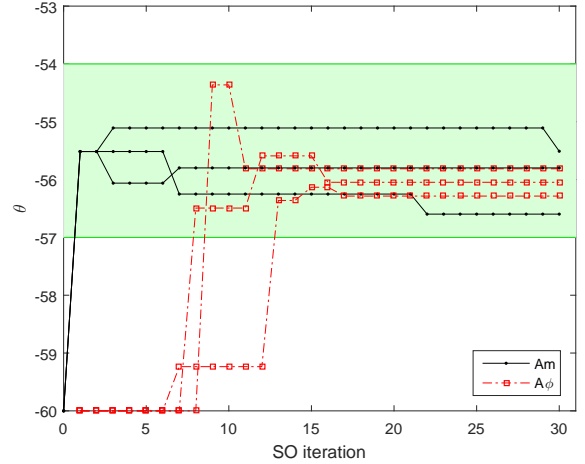


Figure 13: Algorithmic solutions versus iterations, for $\theta^* = -55\text{hr}^{-1}$ and $\theta_0 = -60\text{hr}^{-1}$

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 traffic 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. The method $A\phi$ converges on average at iteration 14.4 (this average includes the non-converged cases, where we input 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 10 of the Appendix.

Table 6 analyzes the convergence of the methods in terms of their simulation runtime until convergence. The runtime is in minutes. 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 at 43.1 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 .

		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 5: 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 6: Simulation runtimes until convergence for the toy network (minutes)

3.3 Berlin metropolitan network

3.3.1 Network attributes

Figure 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 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 24335 links, 11345 nodes and 3635 OD pairs. Real traffic counts from sensors on 346 links are available per hour for 24 hours. We focus on the morning peak hour: 8-9am, during which expected demand is 172900 vehicles. For more data and model details, see Ziemke *et al.* (page 120, Section ‘‘Counts’’ 2015) and Ziemke (pages 57-59 2013).

3.3.2 Results

Figure 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 for 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 (15)) is displayed in Figure 17. Both functions have similar form and seem to be non-convex. Note that both figures differ in the y-axis limits. Hence, the analytical function approximates well the form of the simulation-based function, yet is not scaled properly. The



Figure 14: Metropolitan Berlin simulation network



Figure 15: Metropolitan Berlin road network. Map source: <http://www.mapsofworld.com/germany/states/brandenburg>, downloaded on 10/30/2014

scaling is corrected by the metamodel (term $\beta_{i,k,0}$ of Equation (14)). The minimum of the analytical objective function is obtained at -5hr^{-1} , which is in the green equivalence 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 far by the algorithm) versus algorithmic iteration. We do this for both SO methods: A_m and A_ϕ . The results are presented in Figures 18, 19 and 20.

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

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

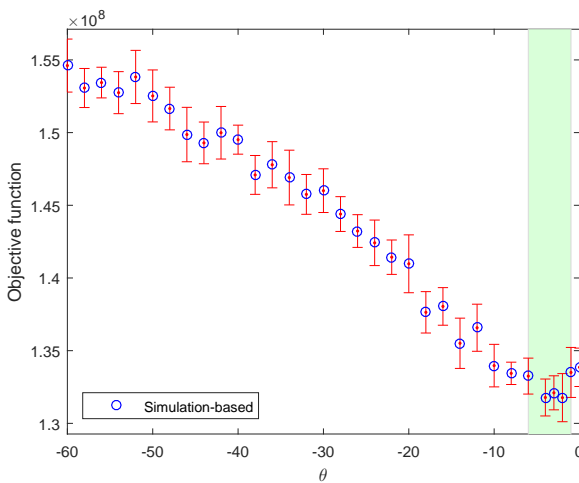


Figure 16: Simulation-based objective function for the Berlin network

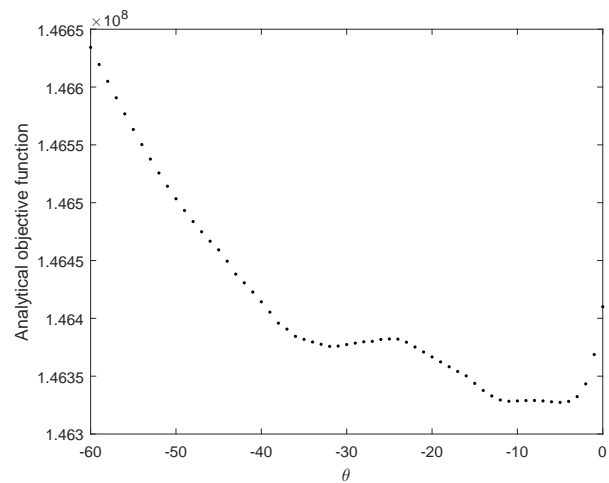


Figure 17: Analytical objective function for the Berlin network

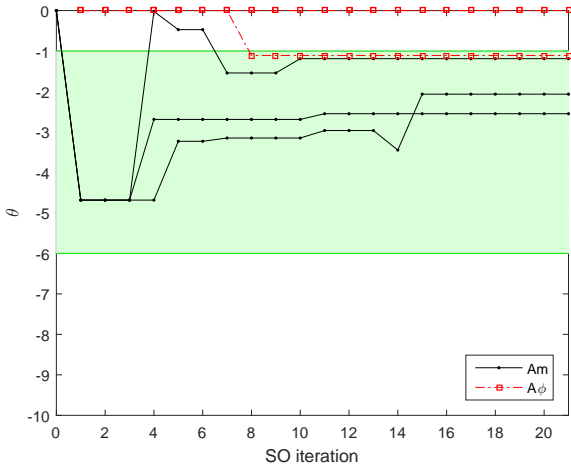


Figure 18: Algorithmic solutions versus iterations, for $\theta_0 = 0\text{hr}^{-1}$

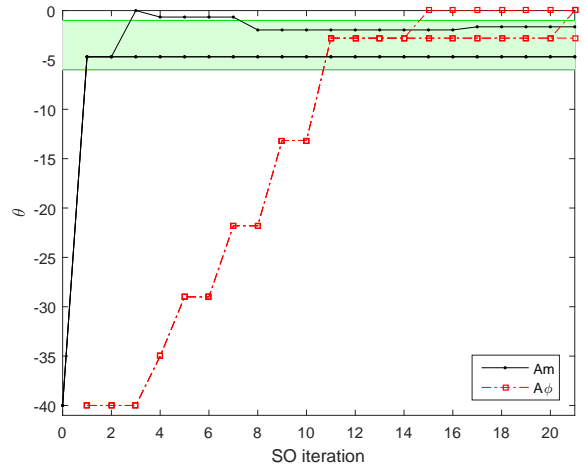


Figure 19: Algorithmic solutions versus iterations, for $\theta_0 = -40\text{hr}^{-1}$

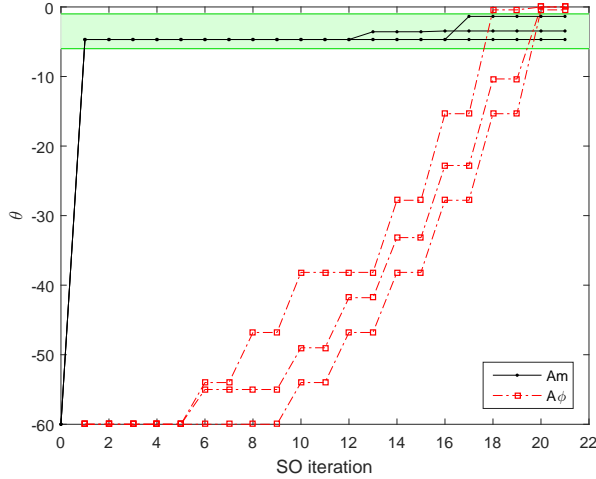


Figure 20: Algorithmic solutions versus iterations, for $\theta_0 = -60\text{hr}^{-1}$

of -0.4hr^{-1} .

For all 3 figures, the following observations hold.

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

We analyze the convergence statistics of the two methods. We proceed as for the toy network. Table 7 considers all Berlin experiments mentioned above. It displays for each experiment and each method, the number of algorithmic iterations until convergence (i.e., the first time the equivalence region is entered and not exited thereafter). If a method upon termination of the algorithm (i.e.,

iteration 20) has not converged, then 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 out of the 9 experiments. On average it converges at iteration 2.4 and at worst at iteration 8. Method $A\phi$ only converges 2 out of the 9 runs. For those two converged experiments, convergence is achieved at iterations 8 and 11, respectively. The corresponding numerical values of the solutions derived by each method for each experiment are presented in Table 11 of the Appendix.

The corresponding statistics in terms of simulation runtimes are displayed in Table 8. If a given method has not converged upon termination of the algorithm, then we indicate the total simulation runtime upon termination (2400 minutes). As before, this underestimates convergence runtime for method $A\phi$ which often does not converge. On average method Am converges within 293.3 minutes (i.e., 4.9 hours), while $A\phi$ does so within 2120 minutes (i.e., 35.3 hours). Method Am achieves on average an 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 as part of the Berlin model, the latter was obtained from prior calibration efforts (Ziemke *et al.*, 2015). Let $\tilde{\theta}$ denote this pre-calibrated value. It has a value of -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 run 50 simulation replications and obtain 50 observations of the objective function. More specifically, for a given θ value, we run 100 assignment iterations to obtain 1 objective function estimate. We repeat this process 50 times to obtain 50 independent simulation observations (or estimates).

Figure 21 displays the cumulative distribution function (cdf) of these 50 observations. Each curve corresponds to a given θ value. The black solid (resp. red dashed) curves correspond to values derived by Am (resp. $A\phi$). The black dotted 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 21 considers the solutions obtained with an initial value of $\theta_0 = 0\text{hr}^{-1}$. 2 out 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 red dashed curves are visible. The right-most red dashed curve represents the

	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 7: Number of algorithmic iterations until convergence for the Berlin network

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

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 22 displays the cdf of each solution obtained when initializing the algorithms with $\theta_0 = -40\text{hr}^{-1}$. Here as well, 2 out of the 3 runs of $A\phi$ yield the same solution of 0. Hence, only 2 red dashed 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 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 out of the 3 runs. Hence, only 2 distinct red dashed 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 is over the 50 simulation replications). In Table 9, we give the numerical values of the best proposed solutions for the three sets of experiments. Figure 24 presents the cdf of the best solutions for all three initial values. It also displays the cdf of pre-calibrated value and of the value obtained by solving the problem with the analytical traffic model only (i.e., no simulation). The black solid 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 black solid curve represents two solutions. The red dashed curves correspond to the three solutions of $A\phi$. The blue dash-dotted curve corresponds to the solution of the analytical traffic model (i.e., no simulation) and the black dotted curve corresponds to the pre-calibrated value.

Figure 24 indicates that the solutions proposed by Am , for each of the three initial values, out-

	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 9: Best solution of each method and each initial θ value

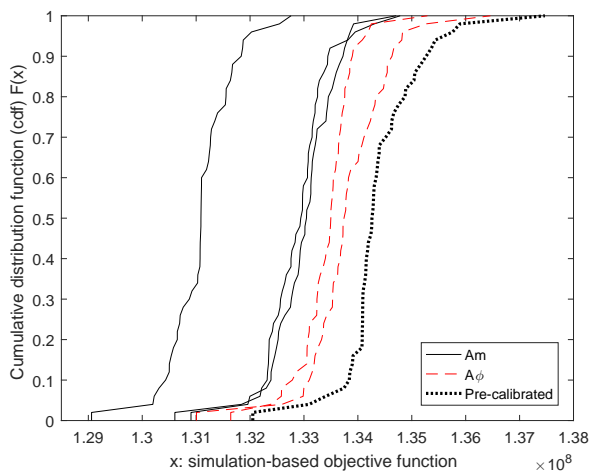


Figure 21: Objective function distributions for each solution of each method, when initialized with $\theta_0 = 0\text{hr}^{-1}$

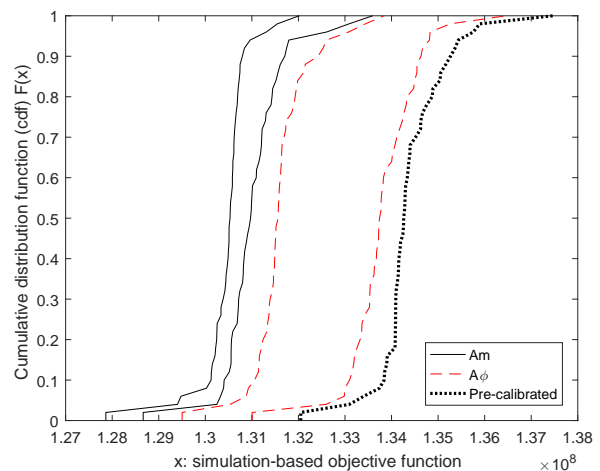


Figure 22: Objective function distributions for each solution of each method, when initialized with $\theta_0 = -40\text{hr}^{-1}$

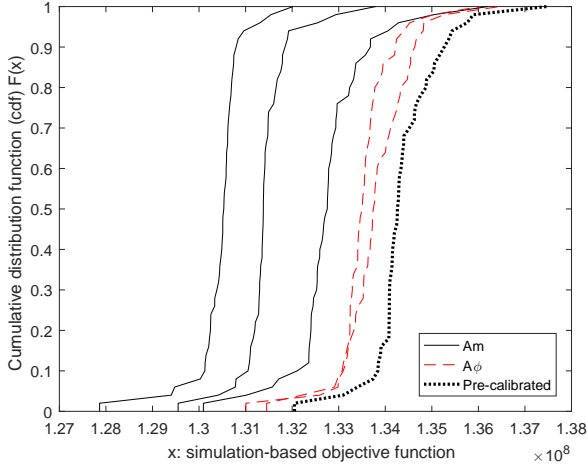


Figure 23: Objective function distributions for each solution of each method, when initialized with $\theta_0 = -60\text{hr}^{-1}$

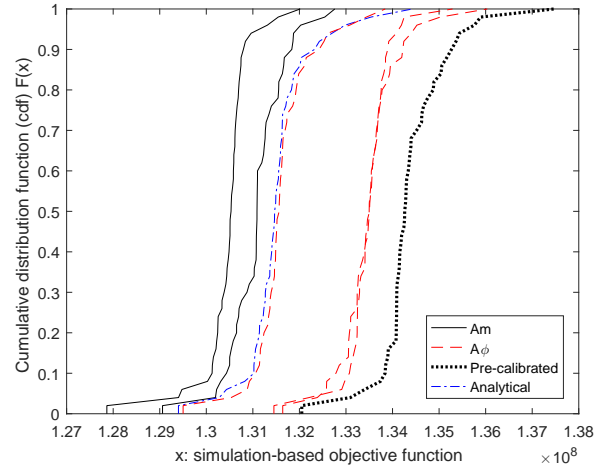


Figure 24: Objective function distributions for the best solutions of each method and each initial θ value

perform all 3 solutions proposed by $A\phi$. It also outperforms the 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 added 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 25 compares the performance of the best Am solution ($\theta = -4.7\text{hr}^{-1}$, the left-most curve in Figure 24) and the pre-calibrated solution. For each solution and each sensor location, we compute the relative error of the count: $(y_i - \hat{E}[F_i(\theta; x)])/y_i$, where y_i represents the field count at location i and $\hat{E}[F_i(\theta; x)]$ represents the simulated estimate of the count at that location. Figure 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 black solid 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 the $x = 0.44$. 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 equivalence region. Figure 26 considers the equivalence 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 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 equivalence 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

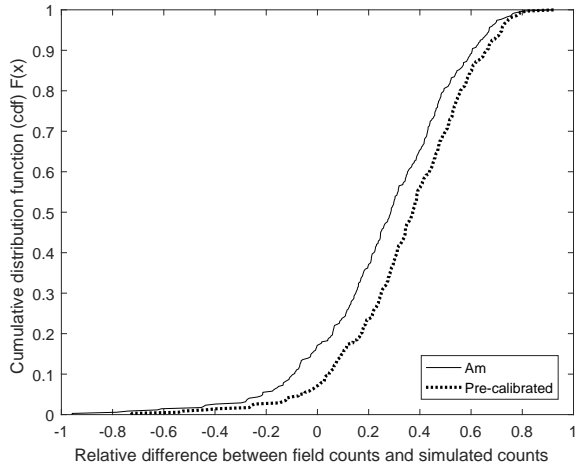


Figure 25: Empirical cdf of the relative difference of traffic counts

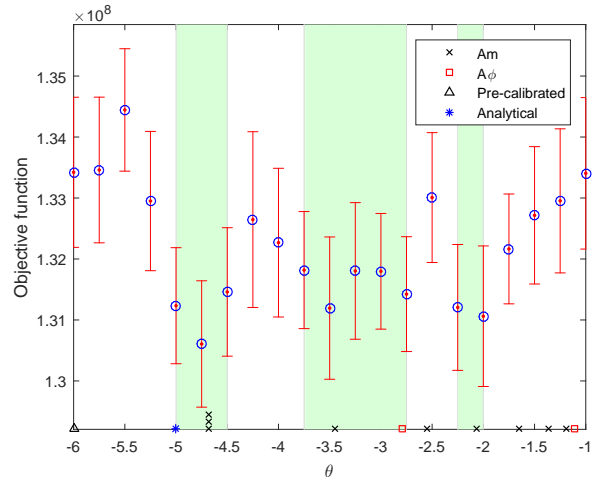


Figure 26: More detailed estimation of the simulation-based objective function in the region $[-6,-1]$ (in units hr^{-1})

points belong to the new equivalence 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 26.

Figure 26 indicates along the x -axis the location of: the 9 solutions proposed by Am (all 9 solutions fall in the initial green equivalence region of $[-6, -1]$ (in units hr^{-1}); they are represented as black crosses), 2 solutions proposed by $A\phi$ (2 solutions fall in $[-6, -1]$ (in units hr^{-1}); they are represented as red squares), the pre-calibrated value $\hat{\theta}$ (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 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 equivalence region, we observe that 5 out 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.

4 Conclusions and discussion

In this paper, 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%.

Ongoing work has formulated a metamodel for a supply calibration problem (Zhang *et al.*, 2016a). A generalization to other data sources and parameters to be calibrated is part of the planned research effort. Ongoing work is also investigating the use of these metamodel ideas to enhance the efficiency and robustness of traditional data-driven calibration algorithms (e.g., Kalman and particle filters).

Acknowledgements

The work of C. Zhang and C. Osorio are partially supported by the U.S. National Science Foundation under Grant No. 1334304. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors thank Prof. Kai Nagel and his group at TU Berlin for the data. In particular, Dominik Ziemke provided modeling and data help. The iMobility laboratory of the KTH Royal Institute of Technology provided access to computing resources used to generate parts of the experimental results.

Appendix

Numerical values of the solutions derived by each method

Table 10 (resp. 11) displays the numerical values of the solutions derived by each method and each experiment of the toy (resp. Berlin) network.

		<i>Am</i>			<i>Aφ</i>		
$\theta^* = -5$	$\theta_0 = 0$	-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 10: Numerical values of the solutions (in units hr^{-1}) derived by each method for each experiment of the toy network

	<i>Am</i>			<i>Aφ</i>		
$\theta_0 = 0$	-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

Table 11: Numerical values of the solutions (in units hr^{-1}) derived by each method for each experiment of the Berlin network

References

- Antonioni, C. (2004). *On-line Calibration for Dynamic Traffic Assignment*. Ph.D. thesis, Massachusetts Institute of Technology.
- Antonioni, 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.
- Ashok, K. (1996). *Estimation and Prediction of Time-Dependent Origin-Destination Flows*. Ph.D. thesis, Massachusetts Institute of Technology.
- Ashok, K. and Ben-Akiva, M. E. (2002). Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. *Transportation Science*, **36**(2), 184–198.
- Balakrishna, R. (2006). *Off-line calibration of dynamic traffic assignment models*. Ph.D. thesis, Massachusetts Institute of Technology.
- Barceló, J. (2010). *Fundamentals of traffic simulation*. International Series in Operations Research & Management Science. Springer, New York.
- Bell, M., Shield, C., Busch, F., and Kruse, G. (1997). A stochastic user equilibrium path flow estimator. *Transportation Research Part C*, **5**(3/4), 197–210.
- Ben-Akiva, M., Gao, S., Lu, L., and Wen, Y. (2012). Combining disaggregate route choice estimation with aggregate calibration of a dynamic traffic assignment model. In *Proceedings of the Fourth International Symposium on Dynamic Traffic Assignment*, Marthas Vineyard, Massachusetts, USA.
- Buisson, C., Daamen, W., Hoogendoorn, S., *et al.* (2012). MULTITUDE review of traffic data collection and estimation techniques and review of methodologies for traffic estimation, calibration and validation. State-of-the-art report 0.97, COST Action TU0903.
- 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)*.
- 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).
- Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM Journal on Optimization*, **20**(1), 387–415.
- Flötteröd, G. (2008). *Traffic State Estimation with Multi-Agent Simulations*. Ph.D. thesis, Berlin Institute of Technology, Berlin, Germany.

- 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., Bierlaire, M., and Nagel, K. (2011). Bayesian demand calibration for dynamic traffic simulations. *Transportation Science*, **45**(4), 541–561.
- 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 Annual Meeting*, Washington DC, USA.
- Ge, Q., Ciuffo, B., and Menendez, M. (2016). An exploratory study of two efficient approaches for the sensitivity analysis of computationally expensive traffic simulation models. *IEEE Transactions on Intelligent Transportation Systems*. Forthcoming.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Horni, A., Nagel, K., and Axhausen, K. W., editors (2016). *The multi-agent transport simulation MATSim*. Ubiquity Press, London, England. Forthcoming.
- Little, J. D. C. (1961). A proof for the queuing formula: $L = \lambda W$. *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.
- 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.
- Osorio, C. (2010). *Mitigating network congestion: analytical models, optimization methods and their applications*. Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne.
- 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 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. (2016). Simulation-based optimization: achieving computational efficiency through the use of multiple simulators. *Transportation Science*. Forthcoming.

- Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, **37**(3), 332–341.
- 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.
- Zhang, C., Osorio, C., and Flötteröd, G. (2016a). An efficient algorithm for the supply calibration of large-scale stochastic traffic simulators. In *Submitted to 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.
- Zhou, X. (2004). *Dynamic origin-destination demand estimation and prediction for off-line and on-line dynamic traffic assignment operation*. Ph.D. thesis, University of Maryland, College Park.
- 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).
- 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.
- 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, The 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.