Efficient offline calibration of origin-destination (OD) demand for large-scale stochastic traffic models

WORKING PAPER

Chao Zhang* and Carolina Osorio

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

Abstract

Traditional calibration algorithms for intricate simulation-based traffic simulators use general-purpose optimization algorithms, which treat the simulator as a black-box. In this paper, we propose to exploit the problem-specific structural information of the intricate (e.g., high-dimensional non-convex) calibration problem to design an algorithm that can quickly identify solutions with good performance. We formulate a simple analytical network model, which analytically captures problem structure through an approximation of the mapping between travel demand and link-level traffic metrics. At every iteration of the algorithm, we use this analytical network information to enable the algorithm to identify good quality solutions and to do so even when few, or no, simulation observations are available. 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, simulation-based optimization 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 demand for the Berlin (Germany) metropolitan area. Compared to the benchmark method, the proposed method yields an average of 65% 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. It enables the algorithm to identify solutions with significantly improved performance at the first iteration, when little or even no simulation information is available.

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

1 Introduction

Urban mobility is becoming increasingly connected and real-time responsive. With this comes an increasingly intricate and interconnected mobility system. This, in turn, is leading to an increased interest in the development and use of traffic simulation models, such as mesoscopic and microscopic traffic simulators, to inform the design and the operations of mobility systems. A recent review of traffic simulators is given in Barceló (2010). Model calibration consists of estimating the simulator's input parameters, so as to have it replicate historical estimates of traffic conditions obtained from surveillance data, such as traffic counts.

Arguably, model calibration is the area of transportation optimization with the most pressing need for algorithms that respond to the needs and requirements of current transportation practice. This is primarily due to the fact that existing algorithms are extensions of general-purpose simulation-based optimization (SO) algorithms. The latter have the advantage of being applicable to an optimization problem in any field (i.e., it need not be a transport simulator). Nonetheless, the use of general-purpose SO algorithms for transportation optimization has two main limitations. First, these algorithms are designed based on asymptotic (e.g., large sample size) properties. However, transportation practitioners use them under tight budgetary constraints: they typically terminate the algorithms once a small number of simulation runs are performed. Second, general-purpose SO algorithms treat the simulator as a black-box. Hence, they do not exploit any problem-specific structural information. Nonetheless, transportation problems and, in particular, transportation calibration problems, have significant problem structure (e.g., such as that induced by the topology of the underlying road network) that could be exploited for improved algorithmic performance.

The focus of this paper is to design algorithms suitable for high-dimensional calibration problems of large-scale networks. Additionally, the focus is on the design of computationally efficient algorithms. These are algorithms that can identify solutions with improved performance, compared to current transportation practice, within few simulation runs. In other words, we design algorithms that are tailored to the current needs of transportation practitioners. The approach in this paper is to achieve computational efficiency by embedding analytical problem structure in the SO algorithm. More specifically, we propose to formulate an analytical network model that approximates the mapping between the decision variables (i.e., calibration parameters) and the simulation-based objective function (i.e., link-level traffic metrics). We then couple information from the analytical network model and the simulator to perform SO. Moreover, we formulate an analytical network model that is both differentiable and scalable (i.e., can be efficiently evaluated for large-scale networks). This further contributes to the computational efficiency of the SO algorithm. The general idea of formulating and using analytical network models to design efficient SO algorithms has been proposed for various traffic signal control problems (Chong and Osorio, 2017; Osorio and Selvam, 2017; Osorio and Chong, 2015; Osorio and Nanduri, 2015a,b; Osorio *et al.*, 2017).

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 calibration problem. 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 origin-destination 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. Third, when using a detailed simulation-based traffic model (rather than an analytical model), there is no closed-form expression for the objective function. Additionally, 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. The use of a stochastic simulator also means that the objective function can only be estimated via simulation. Fifth, high-resolution simulators are computationally inefficient to evaluate. Hence, the evaluation or estimation of the performance of a given point is computationally costly. This is why in practice few simulation evaluations are carried out. Sixth, 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. Nonetheless, these are less efficient, and hence less appropriate for high-dimensional problems than their derivative-based counterparts.

The OD calibration problem is formulated in Section 2.1. 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. Seminal OD calibration papers include Cascetta and Nguyen (1988); Cascetta *et al.* (1993). A summary of recent OD calibration methods is given in Table 1. The table primarily focuses on methods for offline calibration. For each paper, the table indicates whether the problem is formulated as: an OD calibration or a joint supply-demand calibration problem, an online or an offline problem and a time-dependent problem. The table also indicates the network size, in terms of the number of nodes and the number of links, of the largest and/or real-world case study, as well as the dimension of the decision vector. Table cells are left empty whenever the dimension is not directly reported in the paper. All papers use traffic count field data for calibration. This is the most widely available type of traffic data. The last column of the table indicates the papers that consider additional types of field data.

To date, the most popular general-purpose SO algorithm used for model calibration is the Stochastic Perturbation Simultaneous Approximation (SPSA) algorithm of Spall (1992). Although it is a derivative-based SO algorithm, it can efficiently estimate first-order derivative information. It has been used, for instance, in Balakrishna *et al.* (2007); Vaze *et al.* (2009); Lee and Ozbay (2009); Cipriani *et al.* (2011); Ben-Akiva *et al.* (2012). In particular, the work of Balakrishna (2006) compares the performance of several general-purpose SO algorithms. The transportation community has also recently proposed various extensions of the SPSA algorithm, including Cipriani *et al.* (2011); Lu *et al.* (2015); Tympakianaki *et al.* (2015). Another commonly used general-purpose algorithm is the genetic algorithm (GA), which has been used, for instance, in Kim *et al.* (2001); Stathopoulos and Tsekeris (2004); Kattan and Abdulhai (2006); Vaze *et al.* (2009). 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. In this paper, we perform OD calibration for a Berlin metropolitan network with over 11,000 nodes and 24,000 links. In our past work (Zhang *et al.*, 2017), we have 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 paper 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 Zhang *et al.* (2017) is more intricate in that it accounts for

Table 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	Speed, density
Bierlaire and Crittin (2004)	✓		✓		✓	296	618	627	
Jha *et al.* (2004)	✓[1]			✓	✓	1,479	3,756	41,906	
Kattan and Abdulhai (2006)	✓			✓	✓	30	50	400	
Nie (2006)	✓			✓		17	23	4	Link-to-link counts, path travel time
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 time
Lee and Ozbay (2009)	✓			✓	✓	A one-way freeway			Speed
Vaze *et al.* (2009)		✓		✓	✓	825	1,767	6,470	Subpath travel time
Huang (2010)		✓	✓		✓	56	85	638	Speed, density
Cipriani *et al.* (2011)	✓			✓	✓	221	734		Speed
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	
Ben-Akiva *et al.* (2012)		✓		✓	✓	1,698	3,180	69,093	Link travel time
Lu *et al.* (2015)		✓		✓	✓	831	1,040	373,646	
Tympakianaki *et al.* (2015)	✓			✓	✓	N.A.	1,101	1,848	
Zhang and Osorio (this paper)	✓			✓		11,345	24,335	2,585	

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

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 Zhang *et al.* (2017), 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 this paper, we use the derivative-free metamodel SO algorithm of Osorio and Bierlaire (2013). At each iteration, we solve an (approximate) analytical optimization problem, which is constrained by the analytical network model. We propose a differentiable formulation for the analytical network model. This yields a differentiable formulation for the approximate optimization problem. Hence, we can use traditional and efficient gradient-based algorithms to solve this approximate optimization problem. The main contributions of this paper are the following.

Computational efficiency We propose an SO algorithm for offline OD calibration that identifies solutions with improved performance within few simulation evaluations. In other words, it is computationally efficient. This addresses the pressing need of transportation practitioners, which is to design algorithms that work well under tight computational budgets.

Formulation and use of problem-specific analytical structural information To achieve computational efficiency, we formulate and embed within the SO algorithm an analytical network model. The model provides the SO algorithm with problem-specific structural information. Hence, unlike general-purpose SO algorithms, the simulator is no longer treated as a black-box. The analytical network model is formulated as a differentiable and convex model, while the simulator is non-differentiable and non-convex. This allows us to solve, at every iteration of the SO algorithm, an analytical, differentiable and convex optimization problem for which a variety of traditional and efficient gradient-based algorithms can be used to address it.

Scalable algorithm The proposed algorithm is suitable for the efficient calibration of large-scale networks. The analytical network model is formulated as a simple system of linear equations, the dimension of which scales linearly with the number of links in the network and independently of other link attributes (such as link lengths) and of the dimension or structure of the OD matrix. Hence, the analytical network model can be evaluated for large-scale networks in a computationally efficient way. We apply the method for the calibration of a Berlin metropolitan network with 24,335 links, 11,345 nodes and 2,585 OD pairs.

Robustness The case studies of this paper indicate that the use of the analytical network model enables the algorithm to become robust to both the quality of the initial points and to the stochasticity of the simulator. Hence, the performance of the proposed method is similar for various initial points, as well as for various algorithmic runs with the same initial conditions.

A good global approximation We carry out a case study on a simple toy network for which we can estimate the SO objective function across the feasible region. The analytical network model is shown to approximate well the simulation-based objective function in the entire feasible region, i.e., it provides a good global approximation. This shows its added value over other traditional metamodel formulations, such as low-order polynomials, which

typically provide good local yet inaccurate global approximations. In particular, the proposed analytical network model accurately identifies the location of the global minima. This analytical network model allows us to address a non-convex SO problem with numerous local minima via the use of a convex analytical model with a unique (local and) global minimum.

Added value for general-purpose SO algorithms Given the good performance of the proposed algorithm under tight computational budgets, it can be used as a means to initialize other general-purpose calibration algorithms. It can serve to accelerate the convergence of general-purpose algorithms.

The remainder of this paper is organized as follows. Section 2 formulates the calibration problem and the proposed methodology. Section 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 4. Appendix A lists all notation used in the paper; Appendix B gives implementation details; and Appendix C presents the SO algorithm.

2 Methodology

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	expected 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.

$$\min_{d} f(d) = \sum_{i \in \mathcal{I}} \left(y_i - E[F_i(d, u_1; u_2)]\right)^2 + \delta \sum_{z \in \mathcal{Z}} \left(\tilde{d}_z - d_z\right)^2 \tag{1}$$

$$d \geq 0. \tag{2}$$

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 to the prior OD demand information over the traffic count data. Constraint (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 (1)-(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:

$$E[F_i(d, u_1; u_2)] = \sum_{z \in \mathcal{Z}} d_z \sum_{r \in R_z} \Delta_{ri} P_z(r|d, u_1, u_2), \tag{3}$$

where $P_z(r|d, u_1, u_2)$ is the probability that a trip-maker traveling in OD pair z selects 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, itself, on the link travel times, which depend on link flows. Hence, the expected link flow function is typically estimated in an iterative way. In a traffic simulator, these iterations can be interpreted as a learning process over subsequent days, where in each day all trip-makers make route choice 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.

2.2 Metamodel formulation

To address the OD calibration problem, we use a metamodel SO approach. This section formulates a novel metamodel. We then use the metamodel SO algorithm of Osorio and Bierlaire (2013), which is based on the derivative-free trust-region (TR) algorithm of Conn *et al.* (2009). For a review of metamodel approaches, see Osorio (Chap. 5, 2010).

Let us briefly describe the main idea of such an approach. It is an iterative approach, which is summarized in Figure 1. Each iteration, k, consists of two main steps. First, it uses the set of

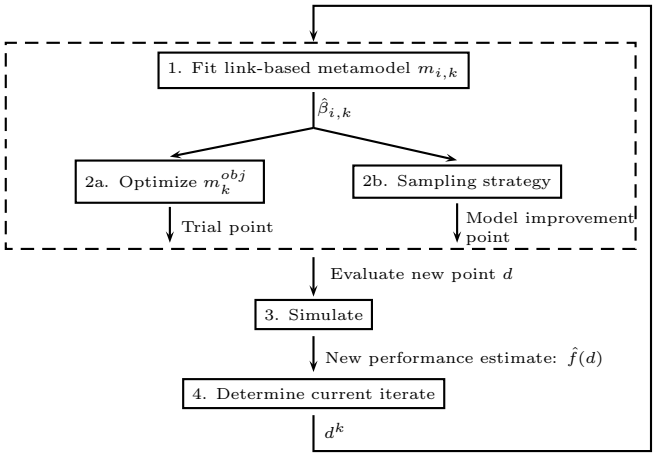


Figure 1: Metamodel simulation-based optimization framework

simulation observations collected so far to fit the parameters of an analytical function, known as the metamodel (Step 1 in Figure 1). A description of how the metamodel is fitted is given in Appendix B.3. The metamodel is then used to formulate and solve an analytical (i.e., not simulation-based) optimization problem (Step 2a of the figure). Second, the solution of this analytical optimization problem is simulated (Step 3 of the figure). These two main steps are iterated until, for instance, the computational budget is depleted. As the algorithm iterates, new points are evaluated by the simulator, i.e., new simulation observations become available, which can lead to metamodels with improved performance.

An additional step in an SO algorithm (Step 2b of the figure) consists of sampling points which are defined by criteria other than metamodel optimization (which is the criteria of Step 2a). Examples of such criteria are improvement of metamodel fit or improvement of geometrical properties of the sampled space. These points are referred to as model improvement points in the figure.

Most often, the metamodel is defined as an analytical approximation of the objective function (f of Equation (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 1) is defined as follows.

$$\min_{d} m_k^{obj}(d) = \sum_{i \in \mathcal{I}} \left(y_i - m_{i,k}(d; \beta_{i,k}, q)\right)^2 + \delta \sum_{z \in \mathcal{Z}} \left(\tilde{d}_z - d_z\right)^2 \tag{4}$$

$$d \geq 0 \tag{5}$$

$$m_{i,k}(d; \beta_{i,k}, q) = \beta_{i,k,0}\lambda_i(d) + \left(\beta_{i,k,1} + \sum_{z=1}^{\text{card}(\mathcal{Z})} \beta_{i,k,z+1} d_z\right) \tag{6}$$

$$\lambda_i(d) = \sum_{z \in \mathcal{Z}} \tilde{p}_{zi} d_z + \sum_{j \in \mathcal{L}} p_{ji} \lambda_j(d). \tag{7}$$

Hereafter, we refer to the above optimization problem as the *metamodel optimization problem*. This problem differs from the initial SO problem (i.e., Problem (1)-(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 is both link- and iteration-specific, and on a vector of exogenous parameters of the analytical network model 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 (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 (6) defines the functional component as a linear (polynomial) function. The physical component is defined by Equation (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 (7)) and the expected demand that arises from trips that arise from upstream links (second summation term in Equation (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 (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 (4)-(7)) that are good solutions to the original SO problem (i.e., Problem (1)-(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 such 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 (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 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 impacts 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 3 indicate that despite these simplifications, the metamodel leads to good analytical approximations for all levels of congestion.

3 Case studies

We use the proposed methodology to address calibration problems for a synthetic toy network (Section 3.1) and for a real-world network of the Berlin metropolitan area (Section 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 (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 (Horni *et al.*, 2016) is used.

3.1 Synthetic toy network

3.1.1 Experimental design

The synthetic toy network is shown in Figure 2. It 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). This is summarized in the first column of Table 2. For each OD pair, there are two alternative routes: a straight route and a route that traverses through nodes 4 and 7. All links are single-lane roads. The full specification of the link attributes is given in Table 8 of Appendix B.2. We focus on a 1 hour time period.

The experimental design is summarized in the first column of Table 3. The table indicates

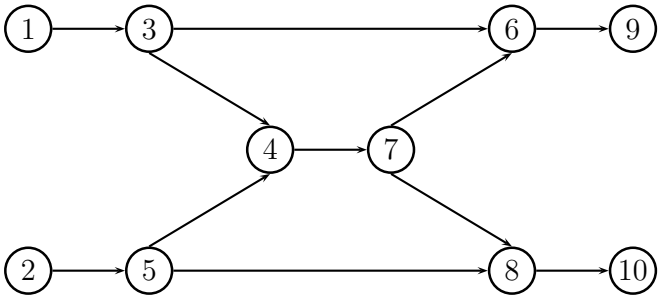


Figure 2: Synthetic toy network

Table 2: Summary of network dimensions

Number of:	Synthetic toy network	Berlin network
links	11	24,335
nodes	10	11,345
OD pairs	2	2,585

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 (1)) is set to 0.01 (row 3). Each algorithm run is terminated once a maximum number of points have been simulated. This maximum number is referred to as the computational budget. It is set to 30 (row 4). For each point (i.e., OD matrix), an estimate of the simulation-based expected link flow is calculated based on a set of 5 independent simulation replications (row 5). For each simulation replication, a sequence of 50 simulation assignment iterations is evaluated (row 6). For each replication, an estimate of the simulation-based expected link flow is calculated as the average of the last 20 assignment iterations. The last row indicates that each run of an algorithm involves evaluating a total of 7,500 simulation calls (i.e., simulation assignment iterations).

The prior OD matrix ($\tilde{d}$ of Equation (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,

Table 3: 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
Computational budget	30	20
Simulation replications	5	10
Simulation assignment iterations	50	100
Total simulation assignment iterations	7,500	20,000

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.

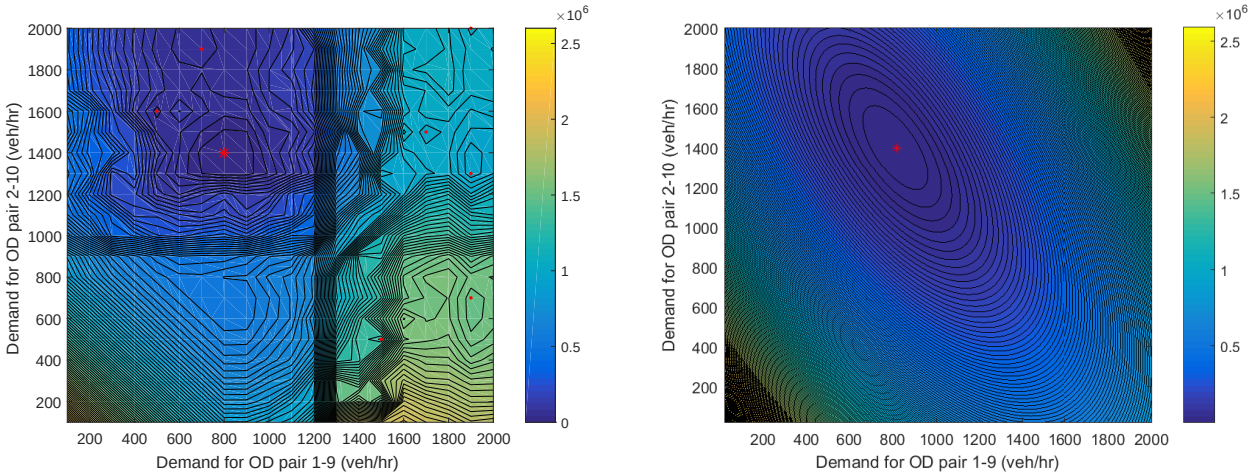
3.1.2 Numerical results

We first consider the asymmetrical true OD demand (i.e., $[800, 1400]$ veh/hr). Figure 3a 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 (1)). This function is estimated by sampling equally spaced and uniformly 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 3b. To compute this function, for a given OD matrix, d, we solve the system of linear equations defined by Equation (7). We then consider Equation (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 3a and 3b indicates that the analytical network model approximates well 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 4 each display 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 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 3a and 4 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 displays six plots. Each plot considers a different scenario. Plots 5a-5c consider the three scenarios with perturbed initial points. Plots 5d-5f 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 Am (resp. $A\phi$) algorithm. The x-axis displays the computational budget consumed so far (i.e., number of points simulated). The y-axis represents



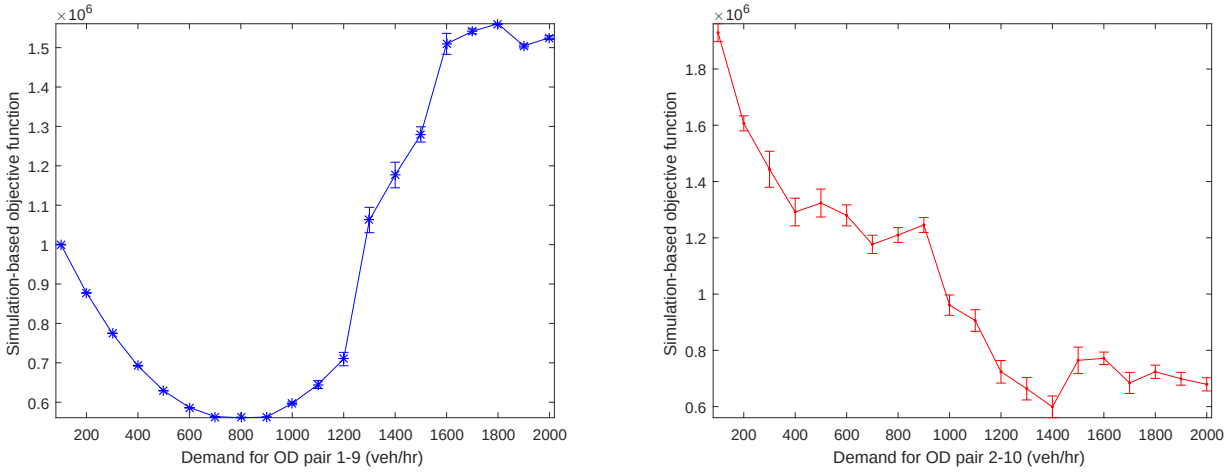
(a) Simulation-based objective function

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

Figure 3: Objective function for true demand $d^* = [800, 1400]$ (veh/hr)

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.

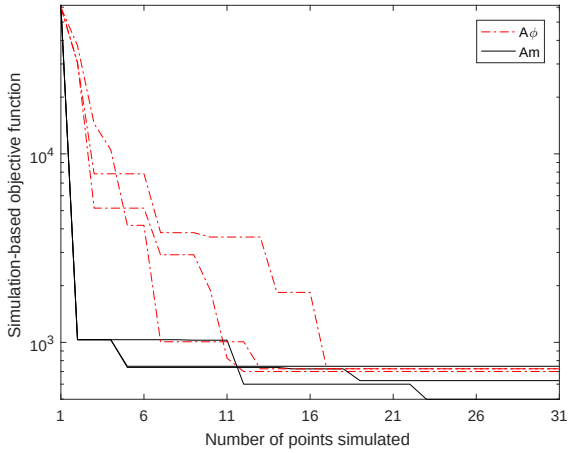
For all Am runs of all plots, the lines of Am achieve a significant reduction of the objective function once the second point is simulated (i.e., at $x = 2$). For Am, the second point to be simulated is that obtained by solving the calibration problem using only the analytical network model. In other words, we minimize the function displayed in Figure 3b. Recall that the y-axis uses a logarithmic scale. Hence, the use of the analytical network model allows to improve the objective function by several 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 only (i.e., the point obtained at $x = 2$). Overall, the benchmark algorithm $A\phi$



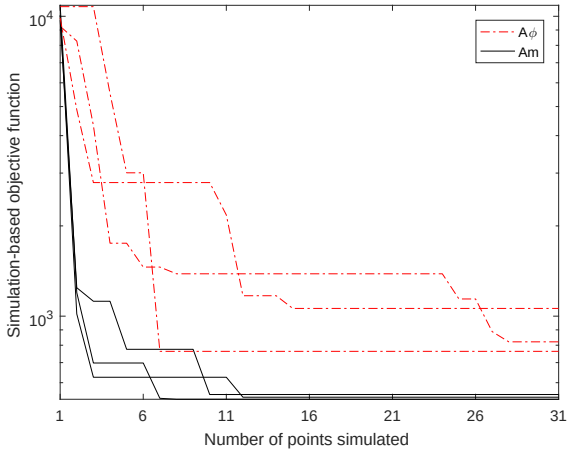
(a) Cut for OD pair 1

(b) Cut for OD pair 2

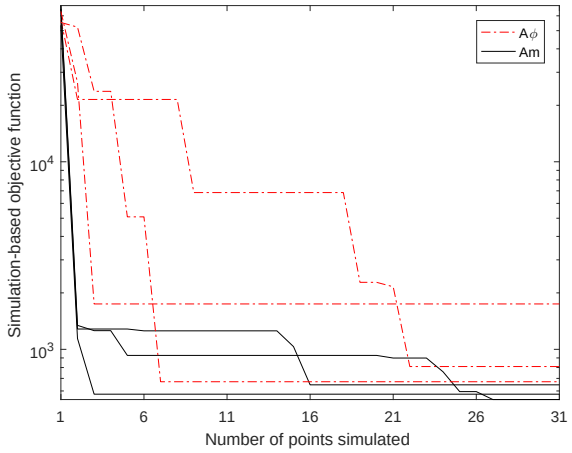
Figure 4: One-dimensional cut of the simulation-based objective function for true demand $d^* = [800, 1400]$ (veh/hr)



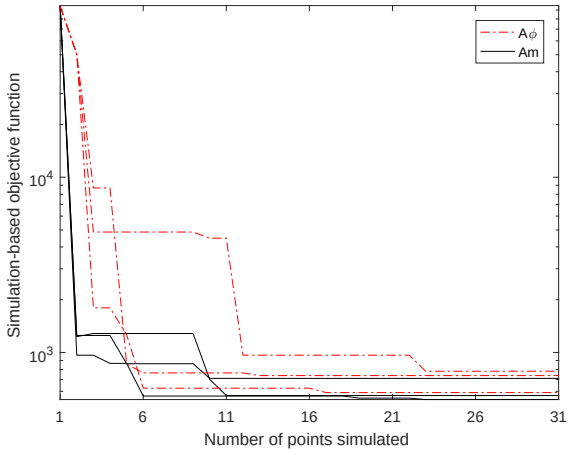
(a) Perturbed initial point 1



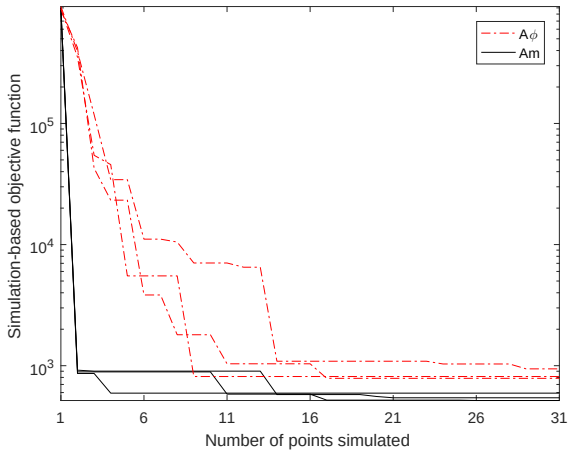
(b) Perturbed initial point 2



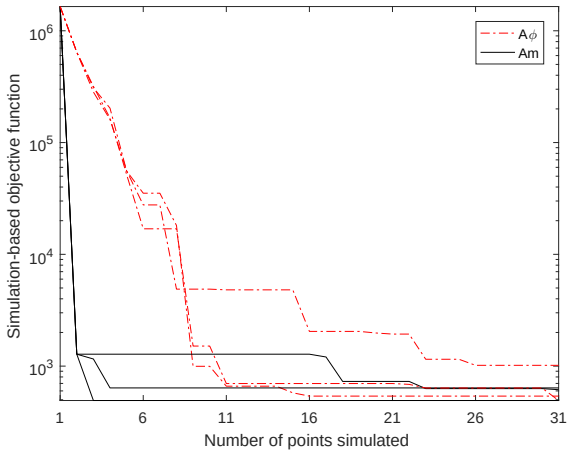
(c) Perturbed initial point 3



(d) Random initial point 1



(e) Random initial point 2



(f) Random initial point 3

Figure 5: 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)

performs well: it reduces the objective function significantly from that of 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 5b, 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 $A\phi$.

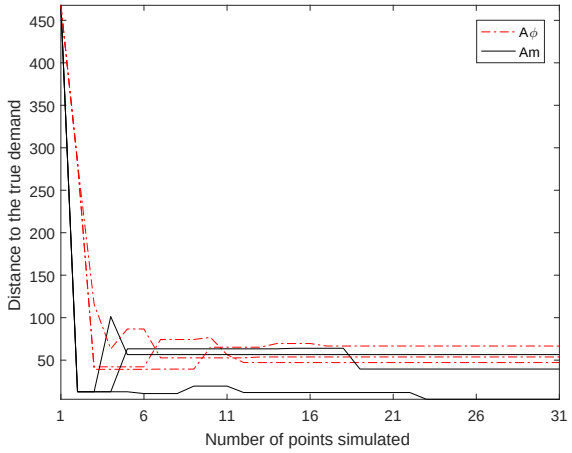
Since we know the true OD demand for these synthetic experiments, we can evaluate the distance of a given demand to the true demand (i.e., the distance to the optimal solution). Figure 6 displays six plots. As before, each plot considers a given scenario. The difference with the previous plots is that Figure 6 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 6b, 6c and 6e. Two of the three Am runs outperform all other runs for plot 6a. For Plots 6d and 6f, the best solution of Am outperforms all other runs.

Figure 7a illustrates the trajectory of a given Am run and a given $A\phi$ run considering the random initial point 3 (i.e., one of the runs in Figures 5f and 6f). 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.

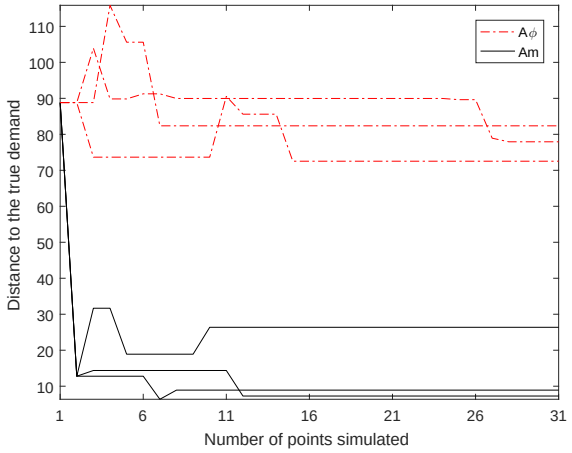
We now evaluate the quality of the solutions proposed by each algorithm for each of the six scenarios. In order to select the "best" solution for a given method and a given scenario, we consider the solutions proposed once the computational budget is depleted (there are 3 such solutions, one for each algorithm run). We then select as the "best" solution, among the 3 proposed solutions, that with the smallest objective function estimate. Figure 8 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 curves of $A\phi$ solutions. This indicates a higher degree of 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 9. 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 which is a global minimum (marked with a red asterisk), while the analytical network model has a single global minimum, which approximates well that of the simulator.

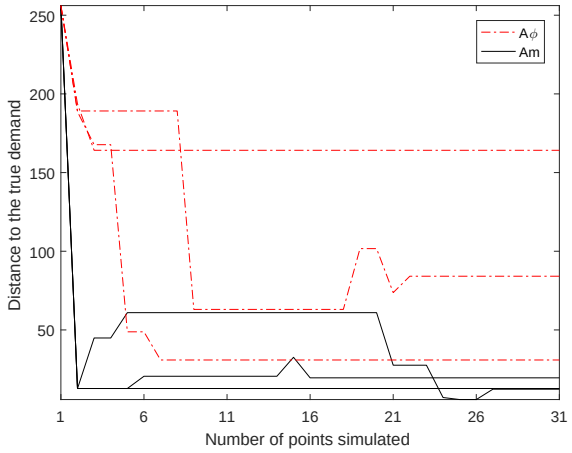
We proceed as before to analyze the performance of the algorithms. Each plot of Figure 10 considers one initial point. Plots 10a-10c (resp. Plots 10d-10f) are associated with scenarios initialized by perturbed (resp. random) points. Similar conclusions as before hold: (i) at the second simulated point, Am yields a significant improvement in performance, which is due to



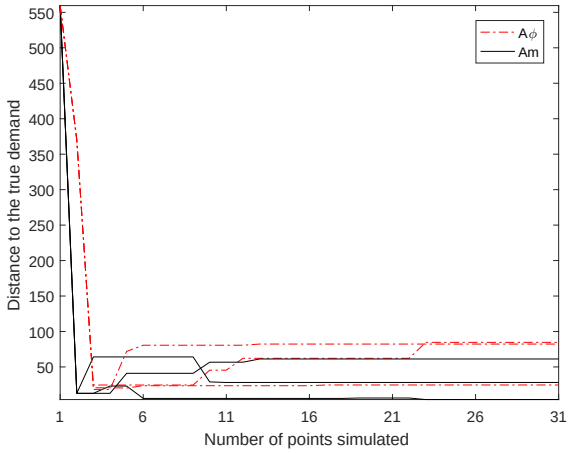
(a) Perturbed initial point 1



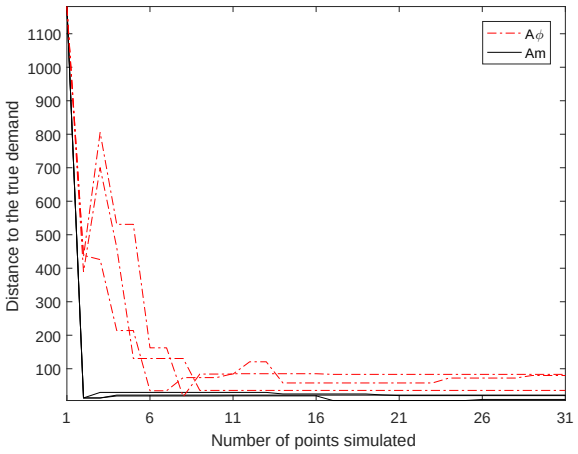
(b) Perturbed initial point 2



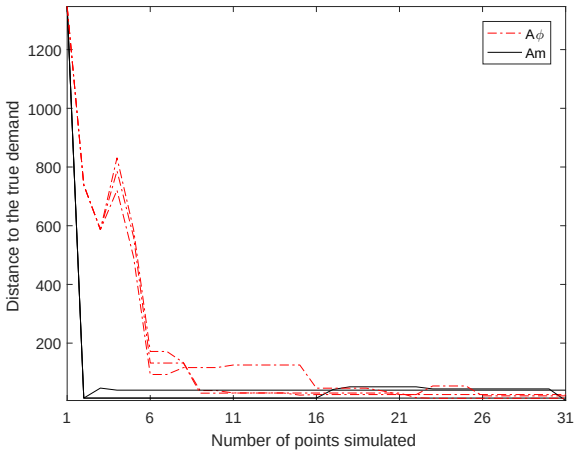
(c) Perturbed initial point 3



(d) Random initial point 1



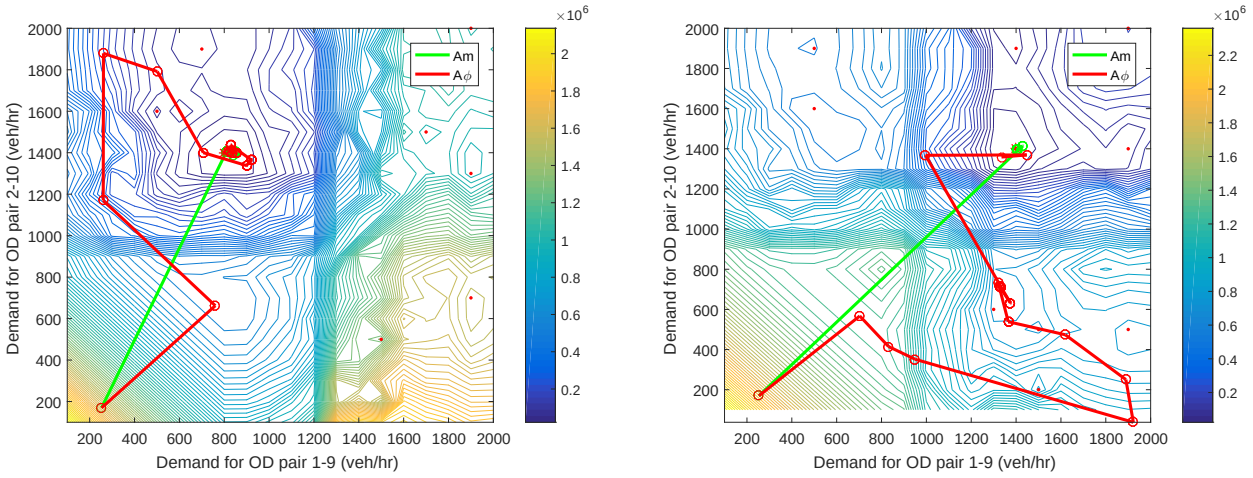
(e) Random initial point 2



(f) Random initial point 3

Figure 6: Distance to the true demand as a function of the total number of simulated points, for true demand $d^* = [800, 1400]$ (veh/hr)

the analytical network model; (ii) as iterations advance, *Am* continues to identify solutions with good performance; and (iii) for most runs, *Am* outperforms *Aϕ* both across iterations and



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

Figure 7: Trajectory of the solutions proposed by each algorithm when initialized with random initial point 3

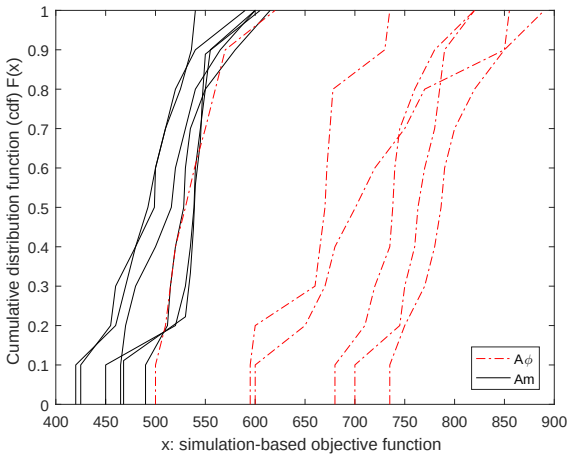
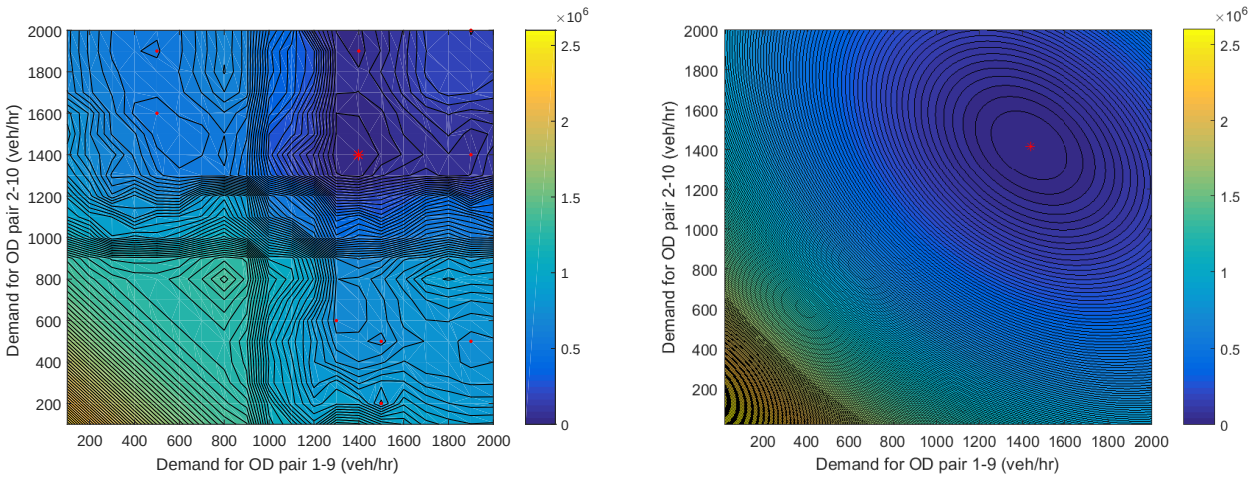


Figure 8: Distribution of the objective function for each of the best solutions proposed by each algorithm, for true demand $d^* = [800, 1400]$ (veh/hr)

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 10e and 10f) have current iterate points with performance similar to that of the initial points up until the last 5-10 points are simulated.

Figure 11 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 11b, 11e and 11f), 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 11f), which upon depletion of the computational



(a) Simulation-based objective function

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

Figure 9: Objective function for true demand $d^* = [1400, 1400]$ (veh/hr)

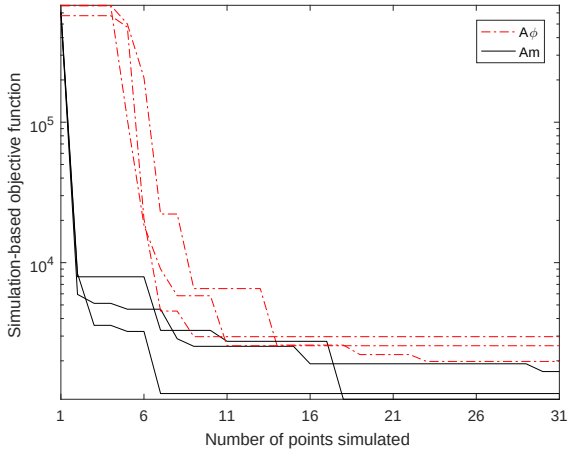
budget, yields a proposed solution that is still far from the true demand (with a distance of ∼800).

Figure 7b illustrates the trajectory of a given Am run and a given $A\phi$ run considering the random initial point 3 (i.e., one of the runs in Figure 10f and 11f). 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.

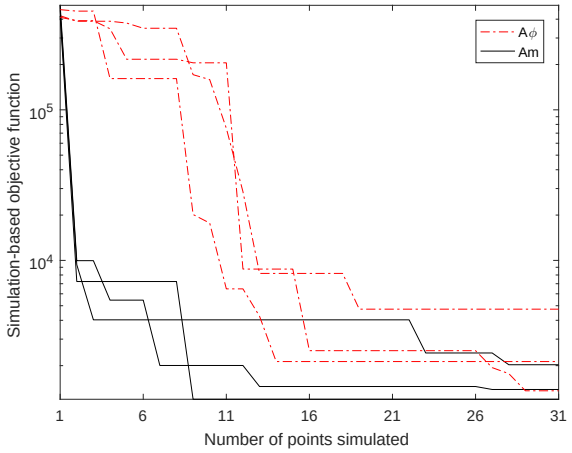
We proceed as before and select, for each scenario and each method, the "best" solution. Figure 12 compares the cdf's of the best solutions across both methods and scenarios. The conclusions are very similar to those of the asymmetrical true OD demand experiments: (i) three of the Am solutions outperform 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 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 model, of Am with regards to both the quality of the initial points and the simulators' stochasticity.

3.2 Berlin metropolitan area

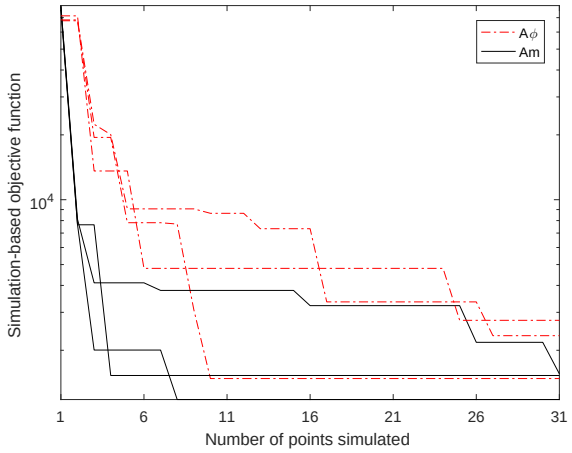
We now study a high-dimensional OD demand calibration problem. We perform OD calibration for the Berlin (Germany) metropolitan area, which consists of two federal states: Berlin and Brandenburg. A map of the road network of this area is shown in Figure 13a. The road network simulation model is displayed in Figure 13b. The network is decomposed into a set of 557 traffic analysis zones (TAZ). These are displayed in Figure 13c. They include 138 LORs (i.e., 'Lebensweltlich Orientierte Räume', i.e., statistical planning region) within the city of Berlin and 419 municipalities in Brandenburg. As summarized in the second column of Table 2, the network consists of a total of 24,335 links, 11,345 nodes and 2,585 OD pairs. The study period is the morning peak period of 8-9 AM. The model considers automobile traffic only. Unlike the synthetic toy network, the link traffic count data is now field data obtained from a set of 346 links; the true OD demand is unknown; and the problem is under-determined. Additionally, the network model is computationally costly to evaluate by the simulator. Hence, there is a



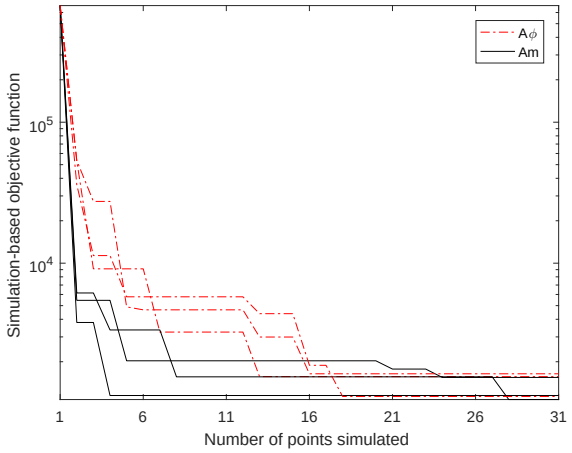
(a) Perturbed initial point 1



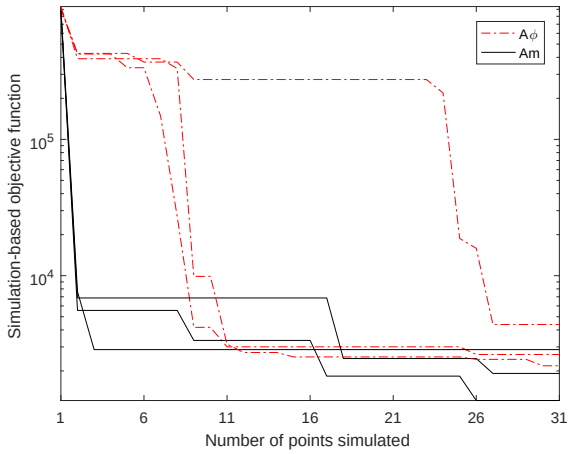
(b) Perturbed initial point 2



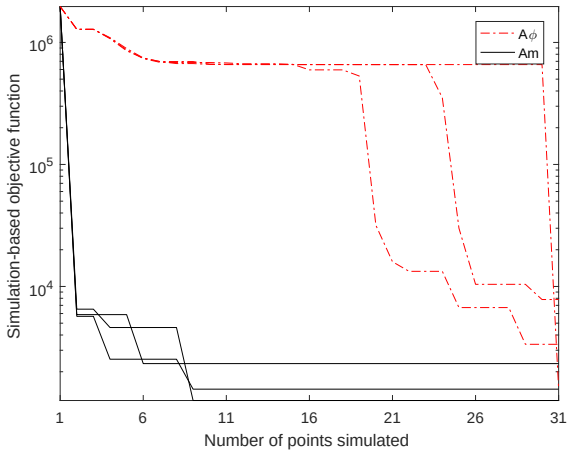
(c) Perturbed initial point 3



(d) Random initial point 1



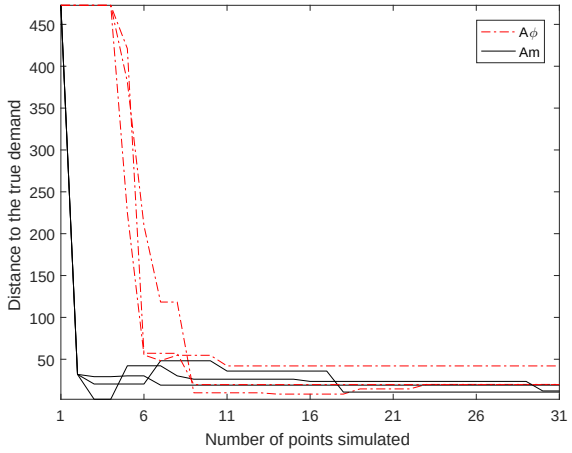
(e) Random initial point 2



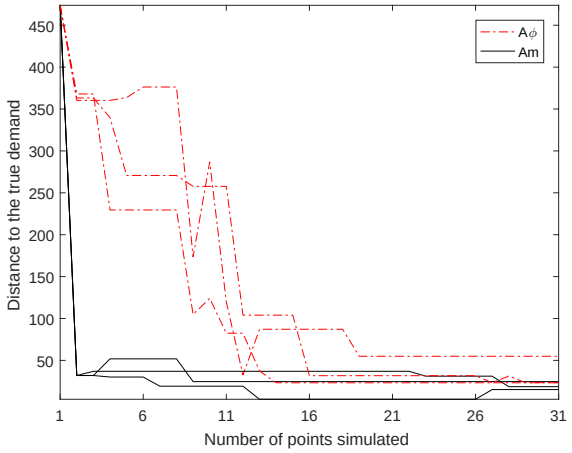
(f) Random initial point 3

Figure 10: 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)

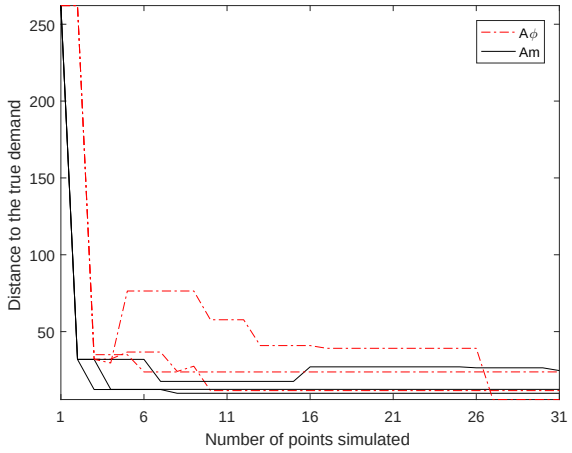
pressing need to design calibration algorithms that can identify points with good performance within small computational budgets. For details regarding the field data, see Ziemke *et al.*



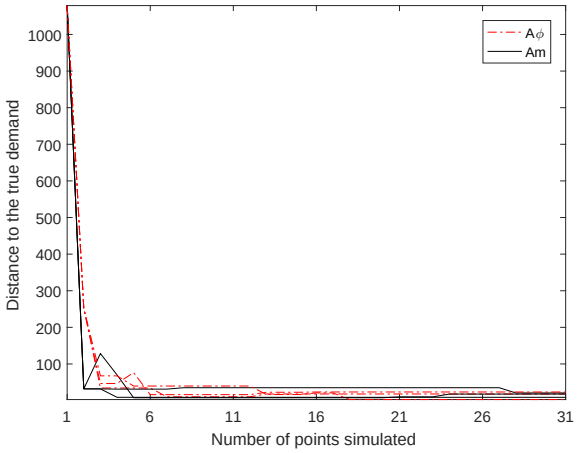
(a) Perturbed initial point 1



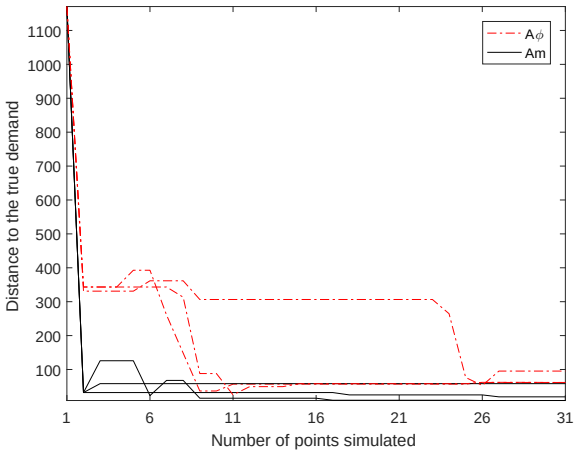
(b) Perturbed initial point 2



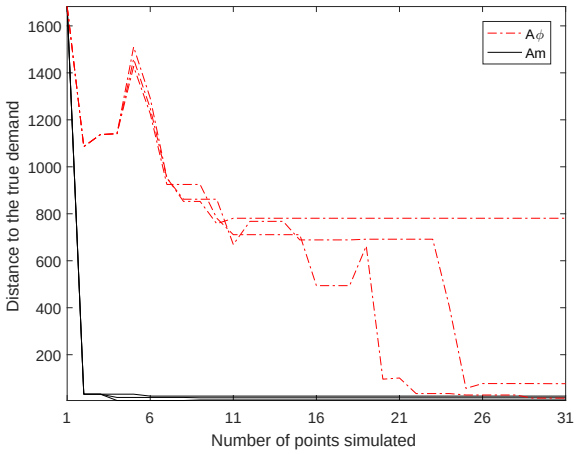
(c) Perturbed initial point 3



(d) Random initial point 1



(e) Random initial point 2



(f) Random initial point 3

Figure 11: Distance to the true demand as a function of the total number of simulated points, for true demand $d^* = [1400, 1400]$ (veh/hr)

(page 8, Section "Counts", 2015) and Ziemke (pages 57-59, 2013).

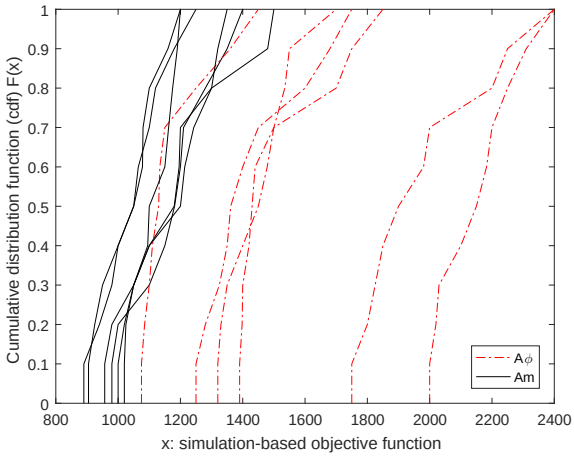
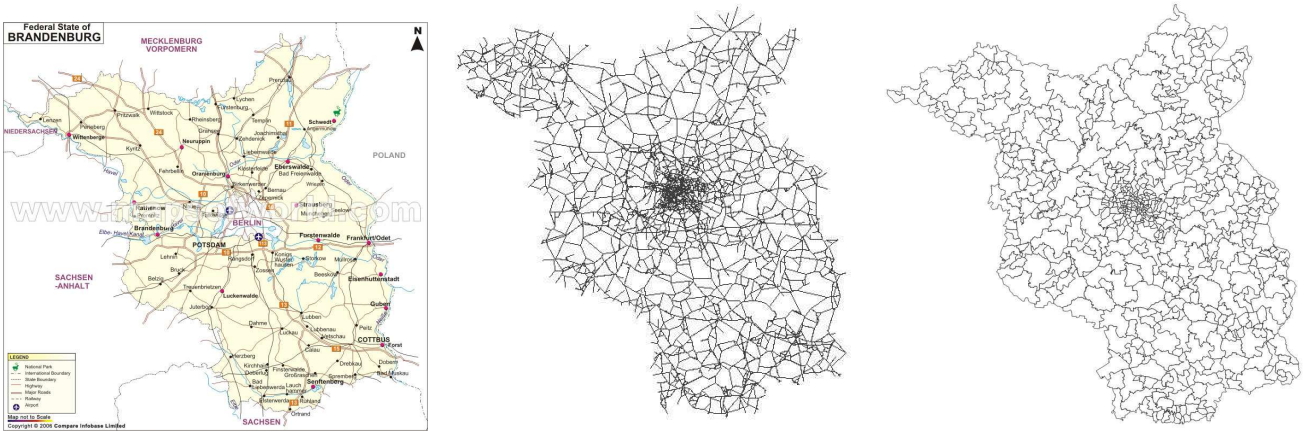


Figure 12: Distribution of the objective function for each of the best solutions proposed by each algorithm, for true demand $d^* = [1400, 1400]$ (veh/hr)

3.2.1 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 3. The demand bounds are the same as for the synthetic case study (row 1 of the table). 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 allow for a maximum of 20 simulated points (row 4), which is the computational budget. To estimate the performance of each point, we run 10 simulation replications (row 5). For each replication, we allow for 100 sequential assignment iterations (row 6). For each replication, an estimate of the simulation-based expected link flow is calculated as the average of the last 20 assignment iterations. This leads to a total of 20,000 (i.e., 20*10*100 = 20,000) simulation assignment iterations for each SO run of each scenario (i.e., each initial point). We consider 3 scenarios, each with a different initial point. Two of them 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



(a) Road network map. Source: http://goo.gl/2kLXzj, downloaded on 06/06/2017

(b) Simulation network model

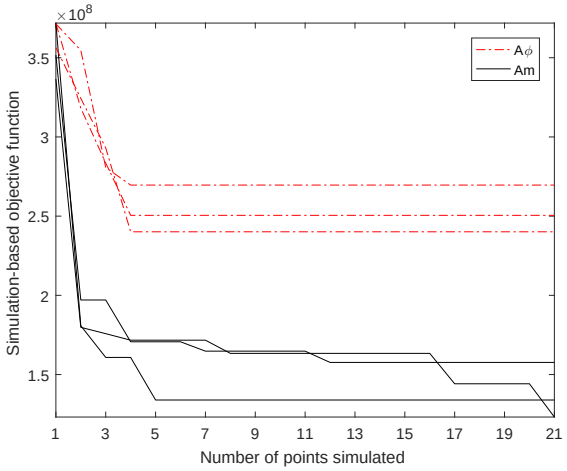
(c) Traffic analysis zones

Figure 13: Berlin metropolitan area network

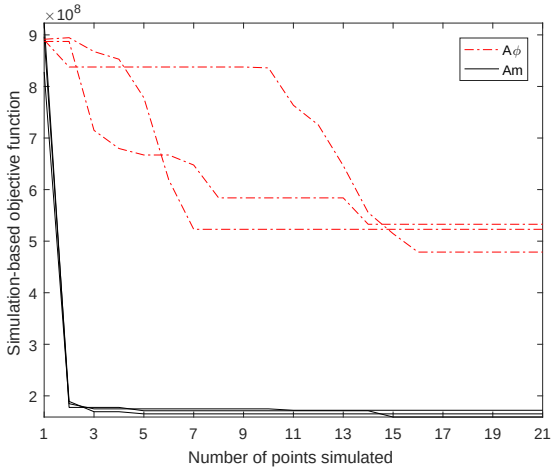
point is randomly drawn from a uniform distribution in $[0, 2000]$. Experiments are carried out on a workstation with a 32-core Dual Intel Xeon Processor E5-2630 v3 and 512GB RAM.

3.2.2 Numerical results

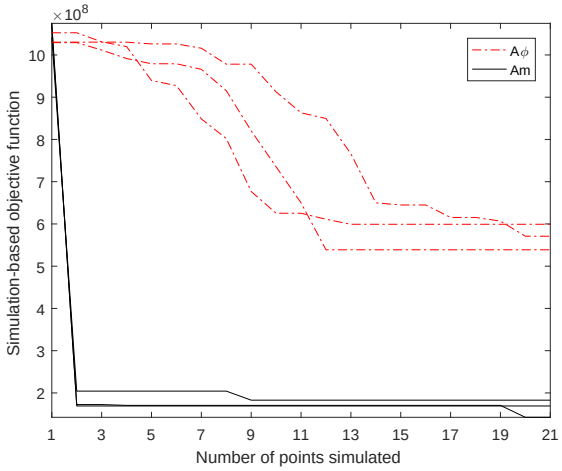
Figure 14 displays one plot for each scenario. Note that the y-axis is no longer logarithmic. For all runs of all three scenarios, Am outperforms $A\phi$ and does so across all iterations (i.e., all x-axis values). Additionally, for all runs and all initial points, upon depletion of the computational 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 65% 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



(a) Perturbed initial point 1



(b) Perturbed initial point 2



(c) Random initial point 1

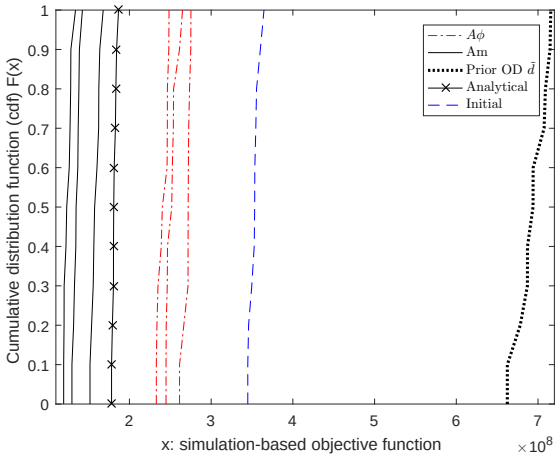
Figure 14: Objective function estimate of the current iterate as a function of the total number of simulated points

identify points with further improved performance. For Figures 14b and 14c, $A\phi$ finds points with improved performance as the iterations advance. Nonetheless, for Plot 14a, for all three $A\phi$ runs, no further progress is made after 4 points are simulated.

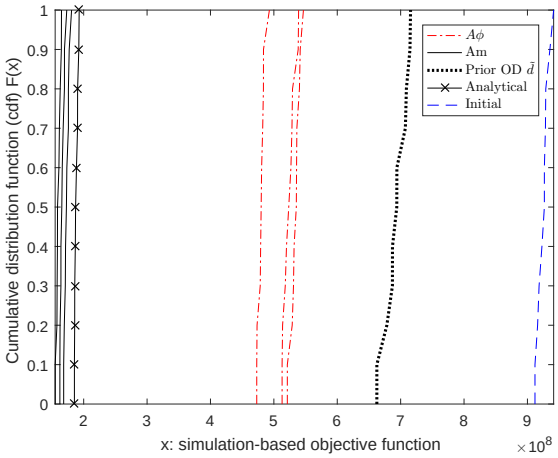
For each scenario, each SO run and each method, we consider the final proposed solution. This corresponds to the current iterate once the computational budget is depleted. For each proposed solution, we run 10 simulation replications. Figure 15 compares the cumulative distribution functions (cdf's) of 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 only using 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 14. Figure 15 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 15, the following observations hold.

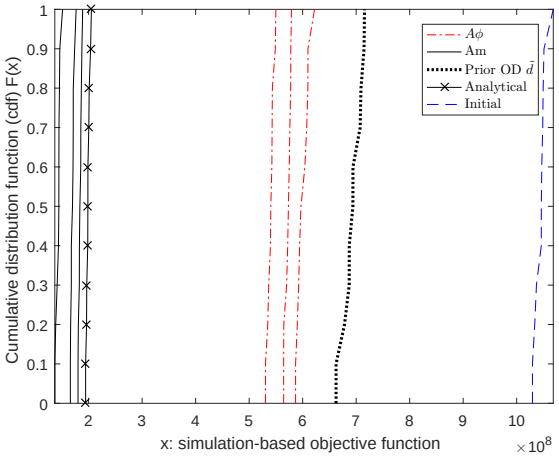
- All solutions of Am outperform all other solutions. They outperform the analytical solution, which indicates the added value of complementing the analytical information with



(a) Perturbed initial point 1



(b) Perturbed initial point 2



(c) Random initial point 1

Figure 15: Distribution of the objective function for each of the best solutions proposed by each method

the simulation-based information. They outperform $A\phi$, which indicates the added value of complementing the simulation-based information with the analytical information.

- The analytical solution outperforms all 3 solutions proposed by $A\phi$. This indicates the remarkable 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 indicates the robustness of Am, provided by the analytical network model, to the quality of the initial points.

Table 4 considers for each initial point, the "best" solution by each method (i.e., lowest 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 from 77% to 82%.

Let us now compare the performance of the methods in terms of their ability to replicate the traffic counts estimated from field data. For each method, we consider all 9 runs displayed in the plots of Figure 15 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 16 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 square normalized error (RMSN) function, which is defined by:

$$RMSN = \frac{\sqrt{\frac{1}{N}\sum_{i\in\mathcal{I}}(y_i - \hat{E}[F_i])^2}}{\frac{1}{N}\sum_{i\in\mathcal{I}} y_i}, \tag{8}$$

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 lists RMSN statistics for the prior OD matrix (column 1), for the best

Table 4: Percentage improvement of the objective function of the "best" solution compared to the prior OD matrix $\tilde{d}$

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

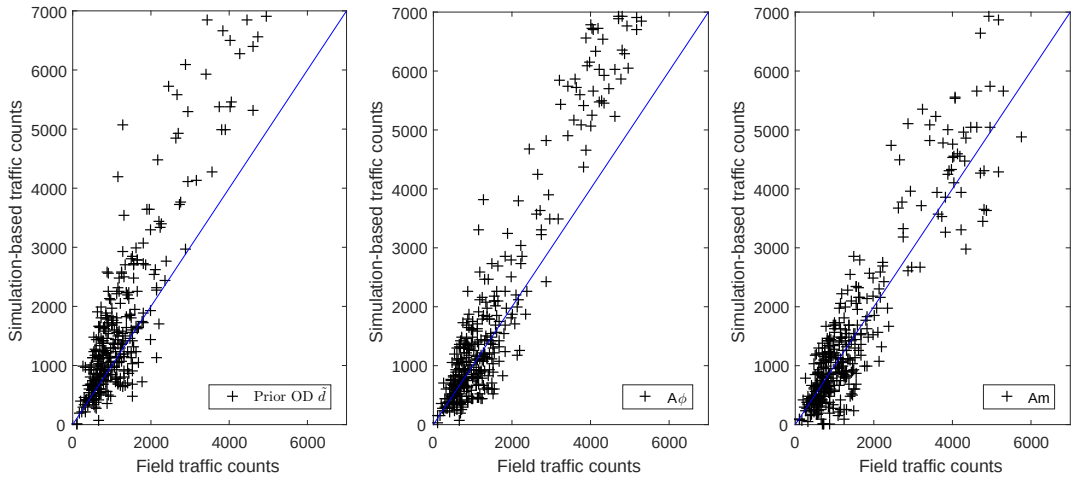


Figure 16: Comparison of field traffic counts and simulated traffic counts

Table 5: Root mean square normalized error (RMSN) statistics

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

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 6 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 solely be attributed to the algorithmic performance of Am and of $A\phi$. 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 (4)-(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., 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 of individual links.

We can combine the results of Tables 4 and 6 to summarize the added value of using the information derived from the analytical network model. It allows the proposed method to improve performance significantly over the benchmark method (i.e., based on Table 4, there

Table 6: 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

is an average of 39.6% improvement for $A\phi$ compared to an average of 79.6% for Am over the prior OD matrix $\tilde{d}$) 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 to identify solutions with significantly improved performance and to do so in a significantly faster way.

4 Conclusions

This paper 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 65% 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, it has allowed the proposed method 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.

Acknowledgments

The work of C. Zhang and C. Osorio is 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. Gunnar Flötteröd for preliminary discussions on this work. The authors thank Prof. Kai Nagel and his group at TU Berlin for the data. In particular, Dominik Ziemke provided valuable modeling and data help.

Appendices

A List of notation

Table 7 displays a list of the notation used in the paper. The variables and parameters in the table are listed in the same order as they appear in the paper.

Table 7: List of all variables and parameters

Notation	Description
z	OD pair index
d_z	expected travel demand for OD pair z
f	simulation-based objective function
i	link index
F_i	flow on link i as defined by the simulator
y_i	expected 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
u_1	endogenous simulation variables
u_2	exogenous simulation parameters
r	route index
$P_z(r\|d, u_1, u_2)$	probability that a trip-maker traveling in OD pair z selects route r
R_z	set of feasible routes for OD pair z
Δ_{ri}	binary indicator that route r traverses link i
k	iteration index
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
d^*	true OD demand vector
Am	proposed metamodel approach

$A\phi$	general-purpose metamodel approach
N	number of links with sensors
$\hat{E}[F_i]$	simulated estimate of the expected flow of link i

B Implementation details

B.1 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.

B.2 Link attributes for the synthetic toy network

Table 8 lists the attributes of each link of the synthetic toy network of Section 3.1.

B.3 Metamodel fitting process

At each iteration k, the parameters of the metamodel, $\beta_{i,k}$, are fitted using simulated observations obtained both at the current iteration as well as at all previous iterations. More specifically, they are determined by solving a least squares problem. To formulate this problem,

Table 8: Synthetic toy network link attributes

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

we introduce the following additional notation. The index k refers to the current SO iteration.

d^k	current iterate;
$\hat{E}[F_i(d)]$	simulation-based estimate of the expected flow on link i for point d;
$w_k(d)$	weight for point d;
w_0	exogenous (fixed) weight coefficient, set to 0.001;
$\mathcal{S}_k$	set of points simulated up until iteration k.

The least squares problem is formulated as follows.

$$\min_{\beta_{i,k}} \sum_{d \in \mathcal{S}_k} \left\{ w_k(d)(\hat{E}[F_i(d)] - m_{i,k}(d; \beta_{i,k}, q)) \right\}^2 + w_0^2 \left((\beta_{i,k,0} - 1)^2 + \beta_{i,k,1}^2 + \sum_{z=1}^{\text{card}(\mathcal{Z})} \beta_{i,k,z+1}^2 \right). \tag{9}$$

The first term represents the weighted distance between the flows predicted by the metamodel and those estimated by the simulator. The weights are defined as in Osorio and Bierlaire (2013):

$$w_k(d) = \frac{1}{1 + \|d - d^k\|_2}. \tag{10}$$

For a given point d, its weight is proportional to its distance from the current iterate, d_k. This aims to improve the local (in the vicinity of the current iterate) fit of metamodel. The second term of Problem (9) accounts for the distance between the parameter, $\beta_{i,k}$, and initial, or prior, values. This second term ensures that the least square matrix is of full rank. The initial values used correspond to an initial metamodel that is solely based on the analytical network model.

C SO algorithm for OD demand calibration

This appendix describes the SO algorithm of Osorio and Bierlaire (2013) as adapted for OD calibration. In the notation below the index k refers to a given SO iteration. We introduce the following additional notation, which is that of Osorio and Bierlaire (2013): Δ_k is the trust region radius, s_k is the step size, n_k is the total number of simulation runs carried out up until and including iteration k, and μ_k is the total number of successive trial points rejected. 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 points simulated, n_{max}, this determines the computational budget. Set the number of simulation replications per point $\bar{r}$. The algorithm is displayed in Algorithm 1.

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 derived from the analytical network model by setting the term $\beta_{i,k,0}$ of Equation (6) to zero.

Algorithm 1 SO algorithm for OD demand calibration

Initialization

Set $k = 0$, $n_0 = 0$, $\mu_0 = 0$. Determine the initial point d^0 and the initial trust region radius Δ_0 ($\Delta_0 \in (0, \Delta_{max}]$). Compute $\lambda_i(d^0)$ by solving the linear System of Equations (7). Estimate $E[F_i(d^0)]$ for each link $i \in \mathcal{I}$. Estimate $f(d^0)$. 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 (4)-(7) using only the analytical network model and without using any simulation information. This is done by setting: $\beta_{i,k,0} = 1, \ \ \beta_{i,k,j} = 0 \ \forall j > 0$. Let $\tilde{d}$ denote the solution to this problem. Estimate $E[F_i(\tilde{d})]$ for each link $i \in \mathcal{I}$ and estimate $f(\tilde{d})$ to yield $\hat{f}(\tilde{d})$. Set $n_0 = n_0 + \bar{r}$. Set the initial current iterate to this solution, i.e., set $d^0 = \tilde{d}$.

Initial metamodel fitting

Solve Problem (9) to fit the metamodel $m_{i,0}$ for each link $i \in \mathcal{I}$.

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

  Step calculation

  Solve Problem (4)-(7) subject to a trust-region constraint (i.e., $\|s_k\| \leq \Delta_k$), let $d^k + s_k$ denote the solution, which is referred to as the trial point.

  Acceptance or rejection of the trial point

  Compute $\lambda_i(d^k+s_k)$ by solving the linear System of Equations (7). Estimate $E[F_i(d^k+s_k)]$ for each link $i \in \mathcal{I}$. Estimate $f(d^k + s_k)$. Set $n_k = n_k + \bar{r}$. Compute:

$$\rho_k = \frac{\hat{f}(d^k) - \hat{f}(d^k + s_k)}{m_k^{obj}(d^k) - m_k^{obj}(d^k + s_k)}$$

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

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

  else

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

  end if

  Solve Problem (9) to fit the metamodel $m_{i,k+1}$ for each link $i \in \mathcal{I}$.

  Model improvement

  Let ν_k denote the concatenation of parameter vectors $\beta_{i,k}$ for all $i \in \mathcal{I}$. Compute $\tau_{k+1} = \frac{\|\nu_{k+1} - \nu_k\|}{\|\nu_k\|}$. If $\tau_{k+1} < \bar{\tau}$, then sample a new point d, which is uniformly and randomly drawn from $[0, 2000]$ (veh/hr). Evaluate $\lambda_i(d)$ by solving the linear System of Equations (7). Estimate $E[F_i(d)]$ and $f(d)$. Set $n_k = n_k + \bar{r}$. Solve Problem (9) to update the fit of the metamodel $m_{i,k+1}$ for each link $i \in \mathcal{I}$.

  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

References

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

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.

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.

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.

Chong, L. and Osorio, C. (2017). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*. Forthcoming. Available at: http://web.mit.edu/osorioc/www/papers/osoChoDynSOsubmitted.pdf .

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.

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., 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.

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

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 .

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

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.

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.

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.

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

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

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

Osorio, C., Chen, X., and Santos, B. F. (2017). Simulation-based travel time reliable signal control. *Transportation Science*. Forthcoming. Available at: http://web.mit.edu/osorioc/www/papers/osoCheSanReliableSO.pdf .

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.

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.

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.

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.

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.

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.

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., 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.

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.