

Highlights

Towards automated calibration of large-scale traffic simulations

Vishal Mahajan, Guido Cantelmo, Constantinos Antoniou

- Performance analysis of calibration with noise and bias in initial OD demand matrices
- Bayesian optimization for automatic tuning of calibration algorithm
- Bagging and SPA to reduce variance in calibrated OD demand estimates
- Open-source calibration platform

Author-submitted version

Towards automated calibration of large-scale traffic simulations

Vishal Mahajan^{a,*}, Guido Cantelmo^b and Constantinos Antoniou^a

^aChair of Transportation Systems Engineering, Technical University of Munich, Arcisstraße 21, Munich 80333, Germany

^bDepartment of Technology, Management and Economics, Transport Division, Technical University of Denmark, Denmark

ARTICLE INFO

Keywords:

calibration
traffic simulation
stochastic approximation
Bayesian optimization
traffic demand
transport supply
large-scale optimization
ensembles

ABSTRACT

Large-scale traffic simulation models are a crucial tool for simulating and evaluating different transport solutions, especially now that our mobility ecosystems are evolving at an unprecedented pace. Due to the scale and complexity of these models, numerous parameters exist that can significantly influence their outputs. Thus, calibration of these models is a prerequisite for a “realistic” assessment of new transport policies. Demand and supply are the two components of the traffic simulation models. Origin-Destination (OD) based demand models are widely adopted in the transport modeling community. Local gradient approximation algorithms are a popular optimization choice for calibrating the OD matrices, usually reconstructed using socio-demographic statistics and traffic data. However, the problem of reconstructing OD matrices is highly under-determined, meaning that multiple plausible solutions exist in terms of data and OD structure. Further, considerable time and manual effort are spent fine-tuning the calibration performance. In this work, we propose an end-to-end methodology for sequential calibration of demand and supply parameters that automates various components in the calibration workflow and leverages new ensemble techniques to increase robustness. First, we propose a simple yet effective heuristic to address the bias in the initial estimates. Then, we use Bayesian optimization to automate fine-tuning of SPSA parameters, followed by Bagging and Stochastic Parameter Averaging (SPA) techniques to reduce the variance in the estimates. Finally, we use Bayesian optimization to calibrate the mesoscopic supply parameters. We test our approach on analytical and DTA simulation (SUMO) models with synthetic and real-world data on the network of Munich. The results show that our approach can provide reliable estimates even when data contain substantial errors. Methodologically, we show that bagging and SPA can dramatically improve the performance of state-of-the-art algorithms such as W-SPSA. This is important in two aspects. First, using parallel computing, bagging can improve performances while not increasing the computational times. Second, bagging and SPA can be used with any stochastic optimization algorithm. Finally, we open-source the developed platform in the interest of open science. The platform can be used to calibrate any network in SUMO and can be extended by incorporating new data, parameters, additional components, or libraries.

1. Introduction

A transportation system comprises different parts and their interactions, which results in travel demand and supply of transport services (Cascetta (2001)). Researchers and practitioners develop transport models to study the effects of an ongoing or new phenomenon on the transport system, e.g., the effect of new technology or a policy change on - how, when, from/ to where people move, and their resultant social, economic, and environmental impacts. While analytical or static transport models do exist, their outputs do not fully capture the complex dynamic interactions that occur on a transport network (Chiu et al., 2011). Dynamic Traffic Assignment (DTA) simulation can represent the short-term traffic flow variations and behavioral choices in a large-scale network (Ben-Akiva et al., 2012). Therefore, traffic simulation models are increasingly preferred in modeling applications. Calibration of transport demand and supply parameters is crucial before the models are applied for analysis and forecasting, as inaccurate parameters translate into unreliable simulation outputs. Calibration is the process of finding the simulation model’s parameters so that the difference between the simulated behavior (counts, travel time, speed) and observed behavior is minimized.

Calibration is formulated as an optimization problem to minimize the value of the objective function subject to constraints (Antoniou et al., 2016). Thus, calibration of traffic simulation models depends on three main factors, namely

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under Grant 415208373 for Project TraMPA-Transport Modelling using Publicly Available Data. The codes developed in this work can be accessed at <https://github.com/vishalmhjn/actrys>

*Corresponding author

 vishal.mahajan@tum.de (V. Mahajan); guica@dtu.dk (G. Cantelmo); c.antoniou@tum.de (C. Antoniou)
ORCID(s):

calibration method [objective or fitness function and its formulation, calibration approach, optimization algorithms, the goodness of fit (GOF) criteria], simulation model [assignment method, level of detail] and data [Measures of Performance (MOP), data sources, aggregation, coverage]. The calibration of DTA models is an active field of research with applications such as demand calibration and real-time traffic management. Demand calibration or estimation/ updating of the origin-to-destination (OD) demand matrices using traffic counts is a well-studied problem for transport modelers. Origin-Destination (OD) demand estimation is a particular case of demand calibration where the link traffic volumes/ flows/ counts are used to estimate the OD matrix (Cascetta et al., 1993). When multiple time-dependent OD matrices are to be calibrated, the problem is also referred to as Dynamic Origin-destination Demand Estimation (DODE) (Cantelmo et al., 2018). Researchers have proposed various methods exploiting the data, models, and problem structure. On the algorithmic side, black-box optimization using approximated gradients is widely used to calibrate OD matrices.

For a large-scale simulation scenario, calibration suffers from the “curse of dimensionality” (Djukic et al., 2012; Cascetta et al., 2013), because the size of the OD matrix is large and thus the number of parameters. This means parameter calibration becomes increasingly difficult with the increase in the number of parameters or OD pairs. Further, the higher the level of error (bias and noise) in a priori OD estimates, it will be challenging to obtain the desired solution. For calibration and validation (Buisson et al., 2014) of the transport models, researchers and practitioners need MOPs. Traffic flow data or link volumes are the commonly used MOP. It is well known that N independent equations are needed to find the unique solution of the system of linear equations with N unknowns. The availability of lesser equations as compared to the number of unknowns leads to an under-determined system. In transport demand calibration, the number of unknowns (OD demand pairs) greatly exceeds the number of equations (observed data). The stochasticity, such as from the gradient approximation or optimization heuristics, vehicle routing in a simulation model further compounds this. In fact, when the number of unknowns equals the number of equations, multiple solutions can still occur due to the nonlinear nature of traffic, not always captured by conventional traffic data (Frederix et al., 2013). The fact that there are multiple solutions might also make the algorithm prone to get trapped in undesired local optima instead of converging to the desired local optima. To reduce the chance of undesired local optima, extensive analysis is needed to check the reliability and robustness of the solutions. All these practical challenges can lead to increased time complexity and computational burden. Moreover, if the calibration approach is not carefully designed, the calibrated OD parameters might be far from the desired solution. This motivates us to apply enhancements to the current demand (OD estimation) and supply calibration framework and propose an end-to-end methodology to find optimal calibrated estimates while keeping the computational burden in check.

This paper applies simple yet effective heuristics and ensemble techniques (borrowed from the machine learning field) to demand (OD estimation) and supply calibration. Specifically, we test two approaches: Bagging and Stochastic Parameter Averaging (SPA). The latter is a novel algorithm developed in this research and inspired by the Snapshot ensembling (Huang et al., 2017) and Stochastic Weight Averaging (SWA) (Izmailov et al., 2018), used in the field of computer science to find the weights of Deep Neural Networks (DNNs) while avoiding local minima. Using multiple experiments, we show that ensembling effectively reduces the variance in the final OD estimates. Further, the averaged estimates are much closer to the true or desired estimates and thus, use the results of multiple local optimizers to land closer to the desired solution. The fact that bagging can be executed on parallel nodes helps achieve these improvements without increasing time complexity. In addition, we propose automatic tuning of the calibration algorithm and thus reduce the manual effort and time spent in doing so hitherto. The remainder of the paper is structured as follows: section 2 concisely reviews the literature on this topic, section 3 introduces indirect OD estimation and supply calibration, section 4 introduces the methodology of our study, section 5 provides details on experimental design and calibration platform description, followed by section 6 with results, followed by conclusion in section 7 with discussion, implications, and limitations of our study.

2. Literature Review

Omrani and Kattan (2012) reviewed DTA model calibration, focusing on the calibration parameters and approach. The traffic simulation parameters belong to two categories: demand model calibration and supply model calibration. Demand model parameters pertain to trip generation, destination, departure time, mode, and pre-trip route choices. OD estimation is a specific case of demand calibration where time-dependent OD matrices are calibrated. On the other hand, supply model parameters pertain to during-trip route choice, link and junction performance functions, traffic flow models, and driving behavior models such as lane-changing and car-following. The nature of these parameters

can change depending on the granularity of the models, such as macroscopic, mesoscopic, and microscopic simulations.

Earlier, the demand models were calibrated considering other supply parameters as constant and vice-versa. These approaches were followed by sequential (or iterative) calibration (Toledo et al., 2014), where supply calibration is followed by demand calibration in a loop. These approaches, however, failed to capture the intrinsic interaction between demand-supply (Toledo et al., 2014). In contrast, simultaneous calibration of all supply and demand parameters is reported to provide the most efficient estimates (Toledo et al., 2014), although at the cost of additional complexity. Another important distinction is between *offline* and *online* calibration procedures. The former calibrates the model parameters given a set of historical observations. After this offline calibration, these parameters can be updated based on the real-time or streaming data for prevailing traffic conditions in an online calibration (Balakrishna et al., 2007; Antoniou et al., 2005). As for the optimization algorithms, global search methods, EA (Evolutionary Algorithms) (Ma and Abdulhai, 2002), are reported to give good quality solutions. On the one hand, global search methods are relatively less popular on large-scale networks, presumably because they are time-consuming and computationally expensive for large-scale problems. The success of global algorithms depends on the properties of the model and might not scale very well on large networks. Only a few studies have used the algorithms' distribution and parallelization to improve the efficiency of these algorithms and demonstrated their application on medium-sized networks (Omran and Kattan, 2018).

On the other hand, researchers use local search heuristics, such as Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall, 1998a), which are efficient in terms of time and computation. Large-scale calibration is a highly under-deterministic or indeterminate problem with multiple possible solutions. Therefore, local search approaches need enhancements, domain knowledge, and sensitivity analysis to obtain the desired solution. Researchers have further tried to incorporate domain knowledge to improve the performance of SPSA. Some of the successful applications of local heuristics are Weighted-SPSA (W-SPSA) (Antoniou et al., 2015; Lu et al., 2015), cluster-SPSA (c-SPSA) (Tympakianaki et al., 2015), adaptive-SPSA (Cantelmo et al., 2014a). Djukic et al. (2012) applied Principal Component Analysis (PCA) to tackle the high dimensionality of the calibrations to capture the input variation with fewer parameters. Subsequently, the potential of dimensionality reduction was demonstrated in PC-Generalized Least Squares (GLS) (Prakash et al., 2017), and PC-SPSA (Qurashi et al., 2020, 2022). Another approach is to assume a prior distribution (quasi-dynamic assumption) of the data to reduce the number of variables artificially (Cascetta et al., 2013) or to divide the problem into sub-tasks (Cantelmo et al., 2014b). Using meta-models to provide more domain knowledge in black-box optimization helps converge faster. For example, Osorio (2019) approximated the network model using an analytical representation and embedded it as a meta-model within the Simulation Optimization (SO) algorithm. This approach gave promising results for large-scale networks. In another recent study by Ho et al. (2023), authors used modified gradients in SPSA and proposed a differentiable Meta-model assisted SPSA (MSPSA) to speed up the convergence of the SPSA.

The first set of challenges pertains to tuning the parameters of calibration algorithms such as SPSA's gain coefficient. In the case of gradient-based optimization, the learning rate decides the convergence rate. The algorithm can be very slow if the learning rate is too small. In contrast, if the learning rate is large, the algorithm can jump beyond the optimum and oscillate or land in an unsuitable local optimum (too far from the starting iterate), leading to high variance. Large learning rate values can also lead to high values in the OD matrix, leading to simulation overload and slow down and even more time to tune the parameters of the optimization algorithm. In the literature, SPSA gain coefficients, i.e., step-size (a) and perturbation vector (c), are predominantly manually selected after some sensitivity analysis. Spall (1998a) suggested that if the parameters to be optimized vary significantly in magnitude, scaling should be applied to the gain coefficients. Such scaling was applied to step-size coefficients of SPSA by Tympakianaki et al. (2018). However, even after scaling, finding the optimal value of gain coefficients requires conducting sensitivity analysis and expensive function evaluations. The set of parameters for a scenario may not be transferable to a new scenario and thus require a fresh and cumbersome sensitivity analysis. Thus, it costs a considerable time to select the optimum parameters. The costly function evaluations limit the application of automatic parameter tuning methods such as Bayesian optimization to OD demand estimation. Although Bayesian optimization works better than random sampling, the former's application will also be slowed due to time-consuming simulations. Thus, we conclude that no existing systematic approach can help to automate the tuning of calibration algorithm parameter selection in the context of OD demand estimation.

Traffic simulators are stochastic systems, implying that the simulation outputs and gradient approximations based on these outputs are also stochastic. Thus, different types of averaging are used to address this stochasticity. For instance, multiple simulations are averaged during each function evaluation to address the variance in the simulation

outputs (e.g., due to randomness in flow propagation and route choice). Random search choice algorithms, such as SPSA, leads to additional stochasticity because the random choice is made in a selection of perturbation vector during the gradient approximation step, which induces randomness in the search process. To address the randomness in gradient approximation, Spall (1998b) recommended that an average of a few gradient evaluations in every single iteration should be used for each gradient approximation. We term this technique as “gradient replications” to differentiate it from “gradient averaging” wherein gradients across current and past iterations are averaged. On this note, Kostic et al. (2017) tested gradient replications and gradient averaging with the SPSA for demand calibration. They found that gradient replications provide better convergence, whereas gradient averaging does not provide meaningful benefit, which supposedly could be due to a highly uneven and complex loss surface. However, in a general context, such averaging is beneficial when the curvature of the objective function starts to flatten along a dimension, e.g., as in the case of the canal or a valley. In such situations, gradient descent-based optimization methods can be very slow in convergence. In these cases, *Momentum* can help to tackle the slow convergence (Ruder, 2016). Momentum tweaks the gradient descent by providing a short-term memory and taking the weighted average of the gradients from the past runs. References to gradient smoothing across iterations for SPSA can be found in Spall (1998b); Spall and Cristion (1994).

Instead of gradients, averaging parameters or iterates (also called weights in machine learning) across iterations is another popular idea. Spall (2003) mentions that the innovation of the seminal work of Stochastic approximation method by Robbins and Monro (1951) is to do a “form of averaging across iterations”. This was followed by maintaining the running average of the iterates in the case of stochastic optimization algorithms (Ruppert, 1988; Polyak and Juditsky, 1992) for better convergence. For iterate averaging to perform better than individual estimates, most individual estimates must land within the local neighborhood of the true or desired estimate. Otherwise, averaging will lead to poorer estimates (Spall, 2003). Different modifications of iterate averaging are also applied in the case of Stochastic Gradient Descent (SGD) based algorithms in machine learning, where the running average of the weights of the neural network helps to smooth the trajectory of the SGD. For instance, Izmailov et al. (2018) proposed Stochastic Weight Averaging (SWA) where an average of the points/ iterates traversed by SGD with cyclical or constant learning rate is used. SWA finds much flatter solutions than SGD, leads to higher test accuracy, and improves the generalization ability of the neural networks.

Another averaging-related method is based on the ensemble concept. An ensemble of models means combining the decisions/ predictions of a set of individual models to provide a better prediction. Dietterich (2000) pointed out that there can be many possible solutions to a problem in case of insufficient data, which is also the case in OD estimation. An ensemble of models can help to average the individual model “votes” and help to obtain optimal predictions. Further, in machine learning, many models use local search to optimize the objective function and can often get stuck in local optima. Therefore, an ensemble made by running multiple models with different initialization can provide better results. Bagging (short for Bootstrap Aggregating) is a common ensemble method. Bagging predictor (Breiman, 1996) is a technique in machine learning where multiple models are trained on subsets of the training data (bootstrapped datasets). Then the final prediction is the average of the predictions of these trained models. Bagging is helpful if the individual models have high variance since the variance of the averaged model is reduced. Breiman (1996) found that for unstable procedures, bagging works well and “can push a good but unstable procedure a significant step towards optimality”. There are different techniques on how to obtain different models. For instance, in the case of Deep Neural Networks (DNN), cosine annealing or cyclic learning rate is used during the training process, and model snapshots at the end of each learning cycle are used for averaging the predictions. This method is known as snapshot ensembling (Huang et al., 2017).

In this work, we aim to address the above challenges in demand and supply calibration in a unified framework, and our contributions are summarized as follows:

- We develop a methodology to fine-tune the calibration algorithm parameters automatically. Substantial research shows in fact that these hyperparameters play a crucial role, but to the best of the author’s knowledge, no methodology exists to estimate them. This is usually done manually, which is time-consuming and unreliable. This helps to push the calibration process towards an automated approach.
- We find that applying Bagging and Stochastic Parameter Averaging (SPA) techniques can improve the robustness of the results. This is important since, typically, solutions obtained by local search calibration algorithms have high variance, and these ensemble techniques can help to reduce such variance in the estimates.

Table 1
Symbols used in the paper

Symbol	Description
T	Number of time intervals
ΔT	Duration of each time interval
X	Time-dependent demand parameters, e.g., time-dependent OD flows in our case, $X = \{X_t\} \forall t \in T$. In this work, we use the terms dynamic OD matrix and demand parameters interchangeably since they are identical.
X^a	A priori or initial or given time-dependent parameter values, $X^a = \{X_t^a\}$
p	Number of OD pairs
Y	Selected supply parameters
Y^a	A priori or initial of selected supply parameters
q	Number of supply parameters
G	Road network and other fixed supply parameters, $G = \{G\}$
f	Traffic simulation model
M^o	Observed time-dependent sensor measurements, $M^o = \{M_t^o\}$, e.g., $M_t^o = \{C_t^o, V_t^o\}$ for count C and speed V measurements
M^s	Simulated time-dependent measurements, $M^s = \{M_t^s\}$, e.g., $M_t^s = \{C_t^s, V_t^s\}$ for count C and speed V measurements
m	Number of link measurements
Z_1, Z_2, Z_3	Goodness of fitness function between simulated and observed measurements, simulated and prior OD estimates, simulated and prior supply parameters, respectively
w_1, w_2, w_3	Decision weights for error functions Z_1, Z_2, Z_3 , respectively in the multi-objective optimization
L	Weighted overall objective function
B^x	Bias factor for OD matrices X
R^x	Randomness factor for OD matrices X
u	Acquisition function for Bayesian optimization
A	OD flow-Link counts assignment matrix
W	Weight-matrix for W-SPSA, $W = J(A)$, where J is a non-linear function
$w_{cut-off}$	threshold value below which the correlation is set as zero
$w_{round-off}$	boolean variable, if True, then the non-zero correlation between the parameter and the sensor is set to 1
a, c	SPSA gain coefficients
A, γ, δ	other SPSA parameters
K, S, B, E	Number of iterations for W-SPSA, sequential calibration, Bayesian optimization, and ensembles, respectively
τ	Error level, which is acceptable and hence defines successful convergence

- We also provide two additional contributions, which from a methodological standpoint, are minor, but have substantial impacts on the calibration output in practice. First, we develop a one-shot heuristic system that reduces intrinsic bias, reducing computational time. Second, we apply a Bayesian optimization framework that effectively estimates the supply parameters.
- The above approaches are developed using open-source tools and software and made available to advance the research in traffic simulation calibration.

3. Indirect OD estimation

3.1. Problem formulation

The offline calibration problem can be formulated using the notation in Table 1, inspired by Antoniou et al. (2015):

Indirect Dynamic Origin-destination Demand Estimation (DODE) is a specific case of transport demand calibration where values of time-dependent OD matrices are the demand calibration parameters. This can be formulated as the minimization of loss or objective function L :

$$\underset{X, Y}{\text{minimize}} \quad L(M^o, M^s, X, Y, X^a, Y^a) \quad (1)$$

which can be operationalized as follows:

$$\underset{X,Y}{\text{minimize}} \sum_{t=1}^T [\mathbf{w}_1 Z_1(\mathbf{M}_t^o, \mathbf{M}_t^s) + \mathbf{w}_2 Z_2(\mathbf{X}_t, \mathbf{X}_t^a) + \mathbf{w}_3 Z_3(\mathbf{Y}_t, \mathbf{Y}_t^a)] \quad (2)$$

subject to:

$$\mathbf{M}_t^s = f(\mathbf{X}_1, \dots, \mathbf{X}_t; \mathbf{Y}_1, \dots, \mathbf{Y}_t; \mathbf{G}) \quad (3)$$

$$l_x \leq \mathbf{X} \leq u_x \quad l_y \leq \mathbf{Y} \leq u_y \quad (4)$$

and Z measures the discrepancy between the two quantities and is called Goodness-of-Fit (GoF) function or distance metric. In the case of measurements, the two quantities are the simulated and observed measurements, whereas, in the case of parameters, they are the parameter's current value and the parameter's prior value. Equation 2 is a type of multi-objective optimization, and \mathbf{w}_1 , \mathbf{w}_2 , and \mathbf{w}_3 are the assigned weights for these objectives. Equation 3 captures the dependence between simulated outputs and the input parameters, which is directly obtained from the DTA traffic simulator.

Z_2 contributes discrepancy of the current estimates from the initial or historical demand estimates, so the optimization algorithm is penalized for exploring far from the initial OD demand values. Furthermore, if the initial values are biased, dependence on initial values in the objective function can prevent the optimization algorithm from reaching the desired optimum. In other words, a misleading specification of OD prior will restrict the algorithm from recovering the desired values. The same is true for prior values of supply parameters. Thus, when prior parameters are heavily biased or unreliable, Z_2 and Z_3 should be set to a small value. But still, the prior demand matrix has certain structural information, such as the relative magnitude of the demand flows among the zones. Prior information about parameters needs to be provided to narrow down the possible solutions.

Since calibration is a constrained optimization problem (equation 4), we must specify the domain of the decision variables, i.e., values in the OD matrices. The equation 4 specifies the domain of the demand and supply parameters; if the domain for the demand variables is wide, the local search algorithm has more flexibility to find solutions, leading to a higher variance in the results. On the other hand, narrow domain specification restricts the search space. These constraints help provide additional information to the optimization algorithm regarding the search space of parameters.

3.2. Stochastic search and approximation with SPSA

3.2.1. Stochastic Approximation

Equation 2 is a form of an iterative optimization problem where the analytical form of the objective function is unknown. To handle this, we move to Stochastic approximation (SA), which is a family of iterative stochastic optimization algorithms used for the minimization of objective functions without an analytical form. Such objective functions can only be estimated from noisy observations or noisy function evaluations, such as in black box systems. In black box systems, only inputs and outputs can be viewed but not the inner mechanism of the system (Bunge, 1963). A general form of SA is:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \quad (5)$$

where $\hat{\theta}_k$ is the decision vector for the k^{th} iteration and $\hat{g}_k(\hat{\theta}_k)$ is the estimate of gradient at $\hat{\theta}_k$. a_k is the step size or gain sequence. There are different approaches to estimating the gradient of the objective function from limited observations or function evaluations. The naïve gradient estimation can be done using finite differences; the gradient is estimated by perturbing the parameters in the decision vector sequentially, i.e., one at a time, evaluating the objective function as many times as there is the number of parameters, and estimating the gradient. Sequential perturbation of the elements of decision vector and function evaluation at those points has a high time complexity due to the high run-time of large-scale traffic simulators.

3.2.2. Simultaneous Perturbation Stochastic Approximation (SPSA)

SPSA algorithm, by Spall (1998b,a), is a gradient approximation based optimization algorithm for stochastic optimization. In SPSA, the gradient is approximated by perturbing all the parameters simultaneously. This leads to only two function evaluations of the objective function per gradient evaluation. Due to this advantage, SPSA is favored for use in simulation-based OD estimation since function evaluation is expensive and the number of OD parameters is large (of the order of thousands). Furthermore, SPSA reduces the computation time by order of p , where p is the number of dimensions or, in our case, the number of OD parameters. The gradient vector in SPSA is approximated as follows:

$$\hat{g}_k(\hat{\theta}_k) = \frac{L(\hat{\theta}_k + c_k \Delta_k) - L(\hat{\theta}_k - c_k \Delta_k)}{2c_k \Delta_k} = \frac{L(\hat{\theta}_k^+) - L(\hat{\theta}_k^-)}{2c_k \Delta_k} \quad (6)$$

where $\hat{\theta}_k^+ = \hat{\theta}_k + c_k \Delta_k$, and $\hat{\theta}_k^- = \hat{\theta}_k - c_k \Delta_k$. Gain sequences are given by $c_k = c/(k+1)^\gamma$ and $a_k = a/(A+k+1)^\alpha$, where c , γ , a , α and A are the SPSA parameters. The magnitude of gain sequences reduces with k . Δ_k is a random perturbation vector sampled from the Bernoulli distribution with values of $+1$ and -1 with equal probabilities.

3.2.3. Weighted - SPSA (W-SPSA)

SPSA does not account for any domain information and parameter correlations while propagating gradients from objective function to parameters. Thus, various extensions of SPSA for DODE are proposed in the literature, as discussed in the previous section. Of the proposed extensions, the Weighted-Simultaneous Perturbation Stochastic Approximation (W-SPSA) exploits the simulator knowledge to map the correlations of the gradients with the parameters. W-SPSA (Lu et al., 2015; Antoniou et al., 2015) uses instead a weight matrix to account for the correlation of the errors in MOP with the parameters (OD flows) during gradient approximation. This enables the use of information from the traffic simulator to discard the gradient signal from uncorrelated measurements. W-SPSA can also be seen as splitting the original problem into multiple smaller SPSA problems (Antoniou et al., 2015). To show how W-SPSA works, we re-write the loss function (2) by omitting the constants (observed measurements and prior values of the parameters), using θ to denote the demand and supply parameters, and setting $w_2 = w_3$, $Z_2 = Z_3$, and $P = p + q$ for the sake of verbosity:

$$L(\theta) = \sum_{t=1}^T [\mathbf{w}_1 Z_1(f(\theta)) + \mathbf{w}_2 Z_2(\theta)] \quad (7)$$

Now, the additive elements of L can be arranged in a $(m + P)T$ array \mathcal{Z} :

$$\mathcal{Z} = [\mathbf{w}_1 z_{1,1}(\theta) \quad \dots \quad \mathbf{w}_1 z_{1,mT}(\theta) \quad \mathbf{w}_2 z_{2,mT+1}(\theta) \quad \dots \quad \mathbf{w}_2 z_{2,(m+P)T}(\theta)] \quad (8)$$

Where z corresponds to the element-wise error function for each parameter or measurement. The gradient estimation in W-SPSA makes use of the correlation between parameters and measurements based on the following $(PT \times (m + P)T)$ dimensional matrix :

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,m} & \dots & w_{1,mT} & \dots & w_{1,(m+P)T} \\ w_{2,1} & w_{2,2} & \dots & w_{2,m} & \dots & w_{2,mT} & \dots & w_{2,(m+P)T} \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ w_{P,1} & w_{P,2} & \dots & w_{P,m} & \dots & w_{P,mT} & \dots & w_{P,(m+P)T} \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ w_{PT,1} & w_{PT,2} & \dots & w_{PT,m} & \dots & w_{PT,mT} & \dots & w_{PT,(m+P)T} \end{bmatrix}$$

where $w_{i,j}$ is the correlation of i^{th} parameter with j^{th} measurement or parameter. Note that these weights $w_{i,j}$ are different from the weights of multi-objective optimization (as in the equation 2), which are denoted by bold symbol \mathbf{w} . The gradient calculation steps for the i^{th} parameter can be written as follows:

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{\sum_{j=1}^{(m+P)T} w_{ij} [\mathcal{Z}_j^+ - \mathcal{Z}_j^-]}{2c_k \Delta_{ki}} = \frac{1}{2c_k \Delta_{ki}} \mathbf{W}_i^\top [\mathcal{Z}^+ - \mathcal{Z}^-] \quad (9)$$

where \mathbf{W}_i is the i^{th} row of the weight matrix, and

$$\mathcal{Z}^+ = [\mathbf{w}_1 z_{1,1}(\theta^+) \quad \dots \quad \mathbf{w}_1 z_{mT}(\theta^+) \quad \mathbf{w}_2 z_{mT+1}(\theta^+) \quad \dots \quad \mathbf{w}_2 z_{(m+P)T}(\theta^+)] \quad (10)$$

$$\mathcal{Z}^- = [\mathbf{w}_1 z_{1,1}(\theta^-) \quad \dots \quad \mathbf{w}_1 z_{mT}(\theta^-) \quad \mathbf{w}_2 z_{mT+1}(\theta^-) \quad \dots \quad \mathbf{w}_2 z_{(m+P)T}(\theta^-)] \quad (11)$$

It can be seen that the gradient for each parameter is computed differently (equation 9) in W-SPSA instead of a single gradient value for all parameters as in the case of SPSA (equation 6). The gradient matrix for all the parameters can be written as follows:

$$\hat{G}_k = \frac{1}{2c_k} \mathbf{W}^\top [\mathcal{Z}^+ - \mathcal{Z}^-] \oslash \Delta_k \quad (12)$$

where \oslash is the operator for element-wise division of matrices. For further details on W-SPSA, we refer the reader to Lu et al. (2015); Antoniou et al. (2015). Finally, momentum can be used with W-SPSA to obtain the running average of the gradients across iterations for efficient convergence. Thus, the update step (equation 5) can be replaced with the following:

$$\begin{aligned} v^{k+1} &= \beta v^k - a_k \hat{g}_k \\ \theta^{k+1} &= \theta^k + v^{k+1} \end{aligned} \quad (13)$$

Where β is the momentum factor with a value between 0 and 1.

Algorithm 1 W-SPSA, source: Lu et al. (2015); Antoniou et al. (2015)

Input: SPSA gain coefficients $\{a, c\}$ and other parameters $\{\gamma, \alpha, A\}$, number of iterations K or error tolerance τ ,

Initial parameter θ_0

Output: θ^\dagger

```

1:  $L_0 \leftarrow L(\theta_0)$ 
2: for  $k \leftarrow 1, 2, \dots, K$  do ▷ minimize  $\sum_{t=1}^T [w_1 Z_1(M_t^o, M_t^s) + w_2 Z_1(X_t, X_t^a) + w_3 Z_2(Y_t, Y_t^a)]$ 
3:    $a_k \leftarrow a/(k + A)^\alpha$ 
4:    $c_k \leftarrow c/(k)^\delta$ 
5:    $W \leftarrow W_k$ 
6:    $\hat{G}_k \leftarrow \frac{1}{2c_k} \mathbf{W}^\top [\mathcal{Z}^+ - \mathcal{Z}^-] \oslash \Delta_k$ 
7:    $\theta_{k+1} \leftarrow \theta_k - a_k \hat{G}_k(\theta_k)$ 
8:    $L_k \leftarrow \sum_{t=1}^T [w_1 Z_1(f(\theta_{k+1})) + w_2 Z_2(\theta_{k+1})]$ 
9:   if  $L_k \leq L_{k-1}$  then
10:      $\theta^\dagger \leftarrow \theta_k$ 
11:   end if
12:   if  $L_k < \tau$  then break
13:   end if
14: end for

```

4. Methodology

4.1. Overview

The complete methodological framework for off-line calibration is summarized in Figure 1. The figure shows the application of the bias-correction heuristic on the initialized parameters. This is followed by automatic SPSA parameter tuning and, finally, ensembling of W-SPSA with sequential demand calibration and supply calibration (only in case of real data scenario). The following sub-sections provide the details on these aspects.

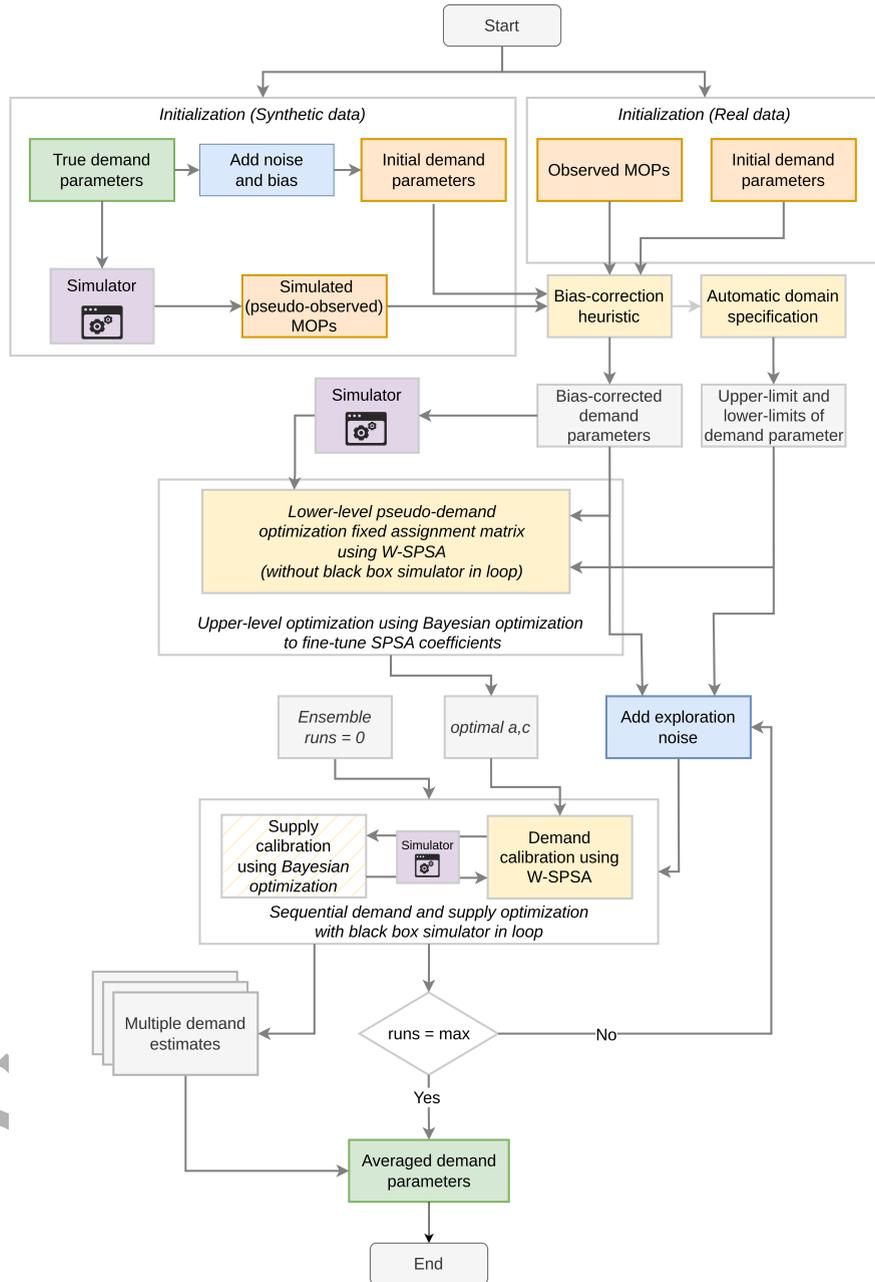


Figure 1: Proposed demand-supply offline calibration framework

4.2. Sequential calibration

Equation 2 implies simultaneous calibration of demand and supply parameters since both sets of parameters are optimized simultaneously in a single objective function. Even though simultaneous calibration of demand and supply parameters provides efficient estimates (Toledo et al., 2014) (since at every iteration, both sets of parameters are consistent), it comes with additional computational complexity and more degrees of freedom. On the contrary, in sequential calibration, demand, and supply parameters are calibrated sequentially. It means the demand parameters are initially calibrated while keeping supply parameters fixed, followed by calibrating supply parameters while keeping the demand parameters fixed. Although this helps to reduce the complexity, this could be time-consuming since the process is repeated till estimates of both sets of parameters are consistent. Therefore, in sequential calibration, Equation 2 can be decomposed into two parts: demand (line 2 in algorithm 2) and supply (line 4) calibration.

Algorithm 2 Sequential demand and supply calibration

Input: weights for sensor counts and prior OD matrices w_1 and w_2 , prior parameters X^a and Y^a , number of sequential iterations S

Output: X, Y

```

1: for  $s \leftarrow 1, 2, \dots, S$  do
2:    $X_s^\dagger \leftarrow \underset{X}{\text{minimize}} \sum_{t=1}^T [w_1 Z_1 (M_t^o, M_t^s) + w_2 Z_2 (X_t, X_t^a)]$   $\triangleright$  Demand calibration
3:    $X_t \leftarrow X_s^\dagger$ 
4:    $Y_s^\dagger \leftarrow \underset{Y}{\text{minimize}} \sum_{t=1}^T [w_1 Z_1 (M_t^o, M_t^s) + w_3 Z_3 (Y_t, Y_t^a)]$   $\triangleright$  Supply calibration
5:    $Y_t \leftarrow Y_s^\dagger$ 
6: end for

```

Sequential calibration provides the advantages of computational simplification of a large optimization problem into two smaller problems. Also, optimization can be flexibly adapted for the demand and supply parameters. This is important because demand and supply have distinct properties, such as a number of parameters, their range of possible values, and parameter sensitivity (Ciuffo et al., 2014) toward simulation outputs. This reason motivates the selection of suitable optimization techniques for each class of parameters. For instance, optimization algorithms scalable to high dimensions, such as SPSA, make sense for demand parameters that are large in number. On the other hand, if the number of supply parameters to be tuned is fewer, other state-of-the-art optimization techniques, such as Bayesian optimization, can be applied.

4.3. Bias-variance decomposition

DODE can be seen as determining the optimal demand and supply parameters based on the given initial conditions (starting parameters), and search process. Due to the estimation process, an error will occur between the estimated demand (or supply) parameters and optimal demand parameters. Now, we define:

- Let $h(\mathbf{x})$ represent the (family of) estimators to be learned from sequential minimization in algorithm 2, where $\mathbf{x} = \{X, Y\}$ are the possible solutions.
- Let $h^*(\mathbf{x})$ be the best estimator i.e., which provides the best values of parameters.
- \mathcal{U} represents the stochasticity of the search process which affects the outcome. This stochasticity can arise due to the characteristics of the optimization algorithm and simulator.
- Then, **bias** is the error between the average estimator (averaged over \mathcal{U}) and the best estimator $h^*(\mathbf{x})$
- Randomness due to \mathcal{U} will give rise to **variance** of a single estimator $h(\mathbf{x})$
- Finally, we have the **noise** or irreducible error, which is the difference between the unobserved true estimator \mathcal{H} and the best estimator $h^*(\mathbf{x})$

Using the Bias-Variance decomposition, the error can be written as:

$$\text{expected error} = (\text{bias})^2 + \text{variance} + \text{noise} \quad (14)$$

where

$$\begin{aligned} (\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{U}}[h(\mathbf{x}; \mathcal{U})] - h^*(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} \\ \text{variance} &= \int \mathbb{E}_{\mathcal{U}} \left[\{h(\mathbf{x}; \mathcal{U}) - \mathbb{E}_{\mathcal{U}}[h(\mathbf{x}; \mathcal{U})]\}^2 \right] p(\mathbf{x}) d\mathbf{x} \\ \text{noise} &= \int \{h^*(\mathbf{x}) - \mathcal{H}\}^2 p(\mathbf{x}, \mathcal{H}) d\mathbf{x} d\mathcal{H} \end{aligned}$$

Initial values or the given parameter values can be seen as belonging to the sub-optimal estimator that needs improvement. Thus, DODE aims to correct initial parameter values to recover the “true” or desired values. If X^* is the best estimate (corresponding to $h^*(\mathbf{x})$) and X^a is the initial/ current or given estimate, then:

$$X^a = X^*((1 - B^x) + R^x \epsilon) \quad (15)$$

where, B^x and R^x & ϵ control the systematic bias and randomness, respectively, in each parameter value. Here R^x is the contribution due to the estimator variance and noise. Thus, the selected estimator should be the one that leads to minimum error. In the following subsection, we provide a step-wise approach to addressing the bias and variance of the estimators:

4.4. One-shot bias correction heuristic

As the true estimator \mathcal{H} is unknown, it is impossible to compute the expected error. Therefore, this sub-section introduces an alternative approach for bias correction (algorithm 3) applicable when the count data from links is available. The functional relationship between the OD flows and sensor counts can be then represented using the following equation (ignoring measurement errors):

$$C = A^\top X \quad (16)$$

where, C is the $(mT \times 1)$ dimensional column matrix for the sensor counts, X is the $(pT \times 1)$ dimensional OD demand column matrix, and A is the $(pT \times mT)$ dimensional assignment matrix of demand onto the sensors. In uncongested networks, link flows depend linearly on the demand because the link costs or assignment matrix in uncongested networks do not depend on the demand. Now we can write the above equation for both the simulation and real scenarios:

$$C_o = A_r^\top X_r \quad C_s = A_s^\top X_s \quad (17)$$

Under the assumptions of the uncongested network and similar demand-link assignment in real-world and simulation, combining the above two equations gives us the following:

$$X_r = B X_s \quad (18)$$

Where B is the factor based on the sensor counts in simulation and measurements. Under assumptions of an uncongested network, we only use a single run of the simulation to upscale or downscale the OD demand matrix for a given time interval. Therefore, it is called a “one-shot”. We approximate B in two ways, as shown in algorithm 3. In first case, we simulate the initial demand X^a and calculate the ratio of the cumulative simulated counts with the cumulative measured counts (Line 5 in algorithm 3) where, $C_{t,m}^s$ and $C_{t,m}^o$ are the simulated counts and observed counts during period t for the m^{th} sensor, respectively, and N_c is the number of sensors in the network. This scalar value is termed the Naïve bias factor, which is used to upscale or downscale the initial values and estimate the intermediate “bias-corrected” OD matrix $\{\hat{X}_t\}$.

The above factor has limitations as it assumes that demand for the current interval only influences the link incidence of the same interval and ignores the correlation of count sensors with the demand. However, in practice, this is not true. To address this, we can also use the simulator knowledge, i.e., the assignment matrix, to obtain an accurate Bias

Algorithm 3 Bias correction heuristic

Input: Initial OD parameters X^a , Other parameters including supply parameters Y , Road network and other fixed supply parameters G , Observed sensor counts C^o

Output: \hat{X}^a

```

1:  $M_t^s \leftarrow f(X_t^a; Y; G)$ 
2:  $C_t^s, S_t^s \leftarrow M_t^s$ 
3: if method=Naïve then
4:   for  $t \leftarrow 1, 2, \dots, T$  do
5:      $\hat{B}_t^x \leftarrow \frac{\sum_{m \in N_c} C_{t,m}^s}{\sum_{m \in N_c} C_{t,m}^o}$ 
6:      $\hat{X}_t^a \leftarrow \frac{X_t^a}{\hat{B}_t^x}$ 
7:   end for
8: end if
9: if method=weighted then
10:   $\hat{B}^x \leftarrow C^s \oslash C^o \cdot W^\top$ 
11:   $\hat{X}^a \leftarrow X^a \oslash \hat{B}^x$ 
12: end if

```

factor. The idea is to estimate the bias factor for the demand flows based on the count sensors which fall along the routes or paths during specific periods for the given demand flows. Thus, the contribution of the uncorrelated count sensors and periods can be omitted. We use the weight matrix (same as the weight matrix in W-SPSA) in line 10 of algorithm 3.

Due to the simplicity of the above heuristic, there is no guarantee that \hat{X}_t^a will lead to a better fit of sensor counts. The proposed method can be applied to the demand corresponding to the off-peak hours before calibrating the demand for the peak hours due to the possibility of congestion. If most of the network during peak hours is uncongested, then the above relationship can be expected to approximate the upscaling or downscaling factor. Therefore, the accuracy of the correction depends on the actual state of the network and how the congestion affects the demand-link assignment within the calibration intervals. Nevertheless, this step is only an intermediate step and provides a principle for initial adjustment in the given estimates. Further fine-tuning is performed by calibration algorithms, which are discussed in the following sections.

Using initial demand i.e., X^a for domain specification can be ineffective since initial values are disturbed due to bias and noise, as shown in equation 15. Instead, we use \hat{X}^a for specifying the domain since they have been partially adjusted for the bias. Further, we specify a domain flexibly depending on each of the values of the parameter, using the $\hat{X}^a l_x \leq X \leq \hat{X}^a u_x$, where, l_x and u_x are the multiplicative factors for specifying the lower bound and upper bound on the parameters. Thus, at least two parameters (l_x and u_x) are needed to specify the domain for the complete set of demand parameters. By using the \hat{X}^a , we take into account the (corrected) prior knowledge about the magnitude of the parameters. The domain specification leads to a fan-shaped domain specification, where the domain is narrow for the smaller values of the parameters, and vice-versa.

4.5. Automatic tuning of SPSA parameters using analytical model

We use an analytical assignment method approximated from the initial simulation run to automatically fine-tune the calibration or optimization algorithm's (such as SPSA) parameters. In this way, we avoid iterating over the computationally expensive simulation-based dynamic assignment. Thus we call it a "simulator out of the loop," i.e., the calibration algorithm does not use DTA or simulation assignment but uses an alternate analytical assignment method. Therefore, we do away with the need to fine-tune the algorithm's parameters with the simulator in the loop, and thus reduce the computational burden and save time. After tuning the calibration algorithm's parameters, we run the calibration with the simulator and similarly call it a "simulator in the loop," i.e., The calibration algorithm involves iteration or looping over the DTA simulator for traffic assignment.

To develop the analytical model, we only use an initial simulation-based assignment to derive the assignment matrix. An assignment matrix is endogenous to the simulator based on the time-dependent OD flows and route choice model and is derived from the incidence of the OD flows on the edges with count sensors. The functional relationship

between the OD flows and link counts can be then represented using the following equation:

$$\hat{C}^s = \mathbf{A}^\top \hat{X} \quad (19)$$

where, \hat{C} are the sensor counts from the analytical assignment, A is the assignment matrix derived from the simulator. We use equation 19, as an approximation of the simulator to fine-tune the algorithm's parameters. This analytical assignment is way faster than running the simulator. This equation can also be seen as a meta-model of the simulation model. This method does not use the sensor or link speeds, since the complex relationship between the link speeds and OD flows is non-linear and cannot be analytically approximated using just the assignment matrix. Thus, to use this approach, sensor counts must be used as MOP in the GoF function. The parameter (ϕ) tuning can be formulated as an optimization problem (equation 20), keeping demand and supply parameters fixed, where, \hat{C}^s is given by equation 19.

$$\phi^\dagger \leftarrow \underset{\phi}{\text{minimize}} \left[\underset{\hat{X}_t}{\text{minimize}} \sum_{t=1}^T [\mathbf{w}_1 Z_1 (C_t^o, \hat{C}_t^s) + \mathbf{w}_2 Z_2 (\hat{X}_t, \hat{X}_t^a)] \right] \quad (20)$$

Overall, the automatic parameters tuning module can be viewed as a hierarchical optimization framework consisting of the following:

1. First-level or inner optimization using calibration algorithm with an analytical model to calibrate the pseudo demand parameters (\hat{X}_t) with a given set of parameters. This is shown by the inner part of the equation 20. We cannot ensure the consistency between the demand and assignment matrix during optimization by using the analytical model (equation 19) instead of the simulator. This is because when there is a change in the demand parameters (\hat{X}), the assignment matrix (\mathbf{A}) is considered fixed during the inner minimization in equation 20. Thus, the calibrated demand parameters here are referred to as pseudo-demand parameters (\hat{X}_t) for the algorithm's parameter tuning. Still, they help decide the appropriate gain coefficient values for optimization based on the magnitude of the parameters.
2. Second-level or outer optimization with Bayesian learning to fine-tune the algorithm's parameters (a_k, c_k) based on the first-level optimization. The reason for using Bayesian optimization is that it is a powerful optimization technique when the objective function is not observed, function evaluations are expensive, and the number of parameters is limited. In this case, the objective function is shown by the outside part of the equation 20. A simple Bayesian optimization algorithm adapted from (Brochu et al., 2010) is presented in Algorithm 4. Bayesian optimization uses an acquisition function u to sample the next data point, deciding between exploration and exploitation (Brochu et al., 2010). By specifying a smooth prior belief, such as Gaussian Process (GP), we can calculate the posterior distribution of the GP by sampling the new data points iteratively. The posterior distribution is the surrogate model of our unobserved objective function (equation 20). The acquisition function samples the points by evaluating the expected value of a surrogate function and selecting the point which maximizes it. We refer the reader to the tutorial on Bayesian Optimization for further details (Brochu et al., 2010).

Algorithm 4 Bayesian optimization adapted from (Brochu et al., 2010)

- 1: **for** $b \leftarrow 1, 2, \dots, B$ **do**
 - 2: Let x represent the gain coefficients $\{a, c\}$, then find x_b by optimizing the acquisition function over the GP:
 $x_b = \underset{x}{\text{argmax}} u(x|D_{1:b-1})$
 - 3: Sample the objective function: $y_b = L(x_b)$ ▷ $L \leftarrow \underset{\hat{X}_t}{\text{minimize}} \sum_{t=1}^T [\mathbf{w}_1 Z_1 (C_t^o, \hat{C}_t^s) + \mathbf{w}_2 Z_2 (\hat{X}_t, \hat{X}_t^a)]$
 - 4: Augment the data $D_{1:b} = \{D_{1:b-1}, (x_b, y_b)\}$
 - 5: **end for**
-

Subsequently, the sequential optimization of demand (and supply) parameters (algorithm 2) is done using the optimal calibration algorithm's parameters obtained by the above hierarchical optimization module.

4.6. Ensembling for variance reduction

Due to the under-determined nature of OD estimation, there can be multiple solutions for a given optimization formulation (Equation at line 2 in algorithm 2) and local-search algorithms, such as SPSA, can result in the distinct local minima resulting in parameters with considerable variance. Due to variance in the spatiotemporal demand patterns, variance in sampling distribution or measurement errors, and simulation behavior stochasticity, some of these solutions can be hypothesized as a manifestation of the desired or “true” solution. Parameter averaging, such as in the bagging technique and SWA, can help to cancel out some of the variance in the individual solution so that the averaged solution is closer to the desired solution.

Algorithm 5 W-SPSA with Bagging

Input: Bias-corrected dynamic OD matrices \hat{X}_t^a , number of bagging ensembles E , exploration parameter σ^2
Output: X^\dagger ▷ Averaged or “bagged” estimate
 1: **for** $e \leftarrow 1, 2, \dots, E$ *in parallel do* ▷ Bagging cycles
 2: $\epsilon \leftarrow \mathcal{N}(0, \sigma^2)$
 3: $\hat{X}^a \leftarrow \hat{X}^a + \epsilon$
 4: $X_e^\dagger \leftarrow \text{minimize } \sum_{t=1}^T [w_1 Z_1(M_t^o, M_t^s) + w_2 Z_2(X_t, \hat{X}_t^a)]$ ▷ Demand calibration using W-SPSA
 5: **end for**
 6: $X^\dagger \leftarrow \frac{1}{E} \sum X_e^\dagger$

4.6.1. Bagging (ensembling with cold restart)

Here we run multiple estimators, such as W-SPSA (in parallel or serial order), and record the final estimates of each run or cycle. Since SPSA is stochastic due to the nature of its search process (see equation 6, where Δ_k is a random vector). Thus, different runs of SPSA with different seeds can lead to different local optima, even if SPSA parameters are kept the same. In all the cycles, the same initial estimate is used, which is why this can be referred to as “cold restart” (algorithm 5), since knowledge from the previous cycle is not used to influence the current cycle. However, we add a small exploratory noise in the initial OD vector to promote the optimization algorithm to find new solutions. With the cold restart, the algorithm has more freedom to explore other possible solutions which are scattered around the desired solution. The final “bagged” estimate is the simple average of all the final estimates from all the W-SPSA cycles. Further, specifically in bagging, individual models can be trained in parallel, thus offsetting the time cost of multiple optimization cycles. For further details on bagging, we refer the reader to Dietterich (2000); Breiman (1996).

4.6.2. Stochastic Parameter Averaging (ensembling with warm restart)

We propose ensembling with warm restart and refer to this approach as SPA (Stochastic Parameter Averaging), inspired by SWA (Izmailov et al., 2018), snapshot ensembling (Huang et al., 2017), and SGD with warm restarts (Loshchilov and Hutter, 2016), for W-SPSA. We use the term “parameter” instead of weight since the former term is more common in traffic calibration literature. In SPA, the gain coefficients are reset after fixed iterations or when the objective function fitness is not changing much. The next optimization cycle uses the iterate from the previous cycle as the initial parameters (algorithm 6); hence, it is referred to as “warm restart”. The resetting of SPSA gain coefficients resembles the cyclic learning rate. The idea is after initial convergence around a probable solution, W-SPSA is further pushed to explore the other solutions for improvement, but in the vicinity of the estimate from the previous cycle. Finally, we take the simple average of cycle estimates to obtain the final “SPA” estimate.

4.7. Calibration of supply parameters

We use Bayesian optimization (Algorithm 4) for calibrating the selected supply parameters (line 4 in algorithm 2). Different data sources, such as point-based, edge-based, and network-based, can be used to calibrate the parameters. The type of supply parameters can vary based on the specific simulator. However, Bayesian optimization is a kind of black-box optimization and thus accesses only the inputs (parameters) and outputs of the objective function. Therefore, supply parameters to be calibrated are selected based on their sensitivity to the output data or corresponding MoPs. If certain parameters are not much sensitive to the outputs, it is not possible to calibrate them with the given data.

Algorithm 6 W-SPSA with Stochastic Parameter Averaging (based on SWA (Izmailov et al., 2018))

Input: bias corrected dynamic OD matrices \hat{X}_t , number of SPA cycles E
Output: X_{SPA} {Averaged SPA estimate}

```

1: for  $e \leftarrow 1, 2, \dots, E$  do
2:    $X_e^\dagger \leftarrow \underset{X}{\text{minimize}} \sum_{t=1}^T [\mathbf{w}_1 Z_1 (M_t^o, M_t^s) + \mathbf{w}_2 Z_2 (X_t, \hat{X}_t^a)]$ 
3:    $X_{SPA} = \frac{(e-1) \cdot X_{SPA} + X_e^\dagger}{e}$ 
4:    $\hat{X}^a = X_e^\dagger$ 
5: end for
6:  $X^\dagger \leftarrow \frac{1}{E} \sum X_e^\dagger$ 

```

5. Experiment design and set-up

5.1. Overview

In this research, the demand parameters are the time-dependent OD matrices. Supply parameters control the traffic propagation and route choice behavior. The details of scenarios with different simulation and data combinations for varying levels of simulation complexity and data are as follows:

1. **Scenario 1:** Analytical assignment with synthetic sensor counts: A randomly generated demand-link assignment matrix is used for mapping OD flows (randomly sampled using a distribution function) to sensor counts using equation 19). In the case of synthetic experiments, where true OD parameters are generated/ known, the algorithm is also validated by the error between the calibrated OD parameters and true OD parameters. The method's performance is evaluated on the fitness of sensor counts and OD matrices. This scenario focuses on obtaining accurate demand estimates (line 2 of algorithm 2), which is why supply parameters are considered fixed. Hence, this scenario is just restricted to demand calibration.
2. **Scenario 2:** SUMO and Munich network with synthetic sensor counts data: Given OD flows are simulated (Moeckel et al., 2020) and corresponding sensor counts are recorded as desired counts. In this case, supply parameters are kept constant and thus not part of the calibration. The method's performance is evaluated on both the fitness to measurements (counts, speeds) and OD matrices.
3. **Scenario 3:** SUMO and Munich network with real-world sensor counts: Given OD flows (Moeckel et al., 2020) are used with sensor counts from real-world data sources (BAST: Bundesanstalt für Straßenwesen, 2023). We use the best-performing approaches in the above scenarios and apply them here. In this case, true OD matrices are not known, and the performance of the algorithm is only evaluated on sensor count fitness. To achieve the best fitness, we calibrate the demand and the supply parameters sequentially (algorithm 2).

5.2. Initialization

Algorithm 7 Initialization

Input: Initial OD parameters X_t (real case) or distribution D_X (synthetic case), Other parameters including supply parameters Y , Road network and other fixed supply parameters G , Observed sensor measurements M_t^o (real case)

Output: X_a, C_t^o, S_t^o

```

1: if scenario = synthetic then
2:    $\mathbf{X}_t^* \sim D_X$ 
3:    $M_t^s \leftarrow f(\mathbf{X}_t^*; Y; G)$ 
4:    $X^a \leftarrow X^*((1 - B^x) + R^x \epsilon)$ 
5:    $C_t^o, S_t^o \leftarrow M_t^s$ 
6: else
7:    $X_a \leftarrow X_t$ 
8:    $C_t^o, S_t^o \leftarrow M_t^o$ 
9: end if

```

▷ Synthetic data scenario

▷ Generate true OD matrix parameters

▷ Generate true sensor measurements

▷ Perturb original parameters

▷ Assign observed measurements

▷ Real data scenario

▷ Assign seed matrix

A “true” OD is sampled from an underlying distribution for the experiments with synthetic data. Based on the empirical findings, we select a right-skewed distribution for sampling the OD demand, so a few OD pairs have many trips mirroring large and dominating zones (such as external zones) within the study area. On the other hand, most zones have a relatively smaller number of trips. The sampled demand matrix (in case of synthetic experiments) or initial OD demand matrix (in case of real scenarios) is given as input to a traffic simulator (Algorithm 7), and corresponding simulation outputs (link counts and link speeds) are recorded.

Subsequently, bias and randomness, proportional to the OD parameter’s magnitude, are added to the true demand values according to equation 15. In this way, a “true” or desired OD matrix is corrupted or disturbed by adding artificial bias and noise. This disturbed OD matrix is used as the initial or given OD matrix (X^a), similar to practical situations where the actual or “true” OD matrix is unknown. However, instead, an error-prone prior estimate is available. Due to errors in the prior demand matrix, we do not use it in the calibration objective function, i.e., we set $w_2=0$ and $w_3=0$ in all the above scenarios (algorithm 2). Thus, optimization is guided by the fitness of counts or speeds measurements ($w_1=1$), but the search is restricted within the domain or structure specified using bias-corrected prior estimates.

5.3. Gradient and performance evaluation

We use primarily Weighted Average Percentage Error (WAPE) (equation 21) as our evaluation criteria for OD fitness and count fitness. WAPE weights the percentage errors based on their magnitude since the scale of the parameters can vary across a wide range. WAPE, also called MAD/ mean ratio, is a preferred alternative over MAPE (Kolassa and Schütz, 2007). This is crucial since the costs of inaccurate estimation of large OD demand flows can be more adverse and thus need to be minimized. Apart from WAPE, we use Root Mean Squared Error (RMSE) for performance evaluation.

$$\text{WAPE} = \frac{\sum (|x - \hat{x}|)}{\sum x} \quad (21)$$

In W-SPSA gradient calculation steps, we scale the estimators (z_1 and z_2) relative to each other using the following method (He et al., 2021):

$$\tilde{z}_2 = z_2 \cdot \frac{\max\{z_1\}}{\max\{z_2\}} = z_2 \cdot \eta_2 \quad (22)$$

where η_2 is the scaling factor. Alternatively, the measurements or parameters can be normalized or standardized before evaluating the estimator. Similar scaling is used for speed measurements if included in the objective function.

5.4. Experiments

We conduct the grid-based evaluation of the effect of the parameters B^x and N^x on the effectiveness of our proposed approach. Since we expect ensembling to be beneficial when the individual estimates are in the neighborhood of each other, by averaging some of the variance can be canceled, and the mean of the estimates is closer to optimal values, as compared to the individual estimates. We hypothesize that with the increase in the magnitude of bias and noise in the initial OD values (X^a), the resulting calibrated estimates can be far from each other, which can lead to reduced effectiveness of the ensembling. This grid-based evaluation helps to define the value of N^x for the following scenarios 2 and 3. We also add randomness to the sensor count measurements and check the impact on the calibrated estimates. The noise is added to mimic random data errors according to $\hat{C}^o = C^o(1 + R^c\epsilon)$. We incrementally add the proposed methodological components to the baseline W-SPSA method and evaluate the improvement. The possibilities are enumerated as follows:

1. **W**: Baseline, using only W-SPSA and manual specification of SPSA parameters.
2. **BC**: Bias correction heuristic
3. **A-W**: W-SPSA with Automatic SPSA’s parameter Tuning (AST).
4. **W-B**: W-SPSA with Bagging (B).
5. **W-SPA**: W-SPSA with Stochastic Parameter Averaging (SPA).

6. **A-W-B**: W-SPSA with Automatic SPSA Tuning, followed by Bagging.
7. **BC-A-W**: Bias correction heuristic, followed by W-SPSA with Automatic SPSA Tuning.
8. **BC-A-W-B**: Bias correction heuristic, followed by Automatic SPSA Tuning for W-SPSA, with bagging

5.5. Computation burden

The computation requirement for convergence of the algorithm depends on many factors. We quantify the computation requirement of our approach in terms of the number of objective function evaluations or traffic assignment instances. Depending on the type of scenario, the type of traffic assignment (analytically or simulation-based) and its time burden can be different. If all the methods are used, then the minimum number of times function evaluation is done can be expressed as $1 + S((3 \cdot E \cdot K) + B)$. This is because we need one evaluation of BC and three evaluations for each W-SPSA (ignoring gradient and simulation replications). We used a desktop PC (8 i7-11700F @ 2.50GHz physical cores and 50 GB RAM) and a workstation (36 Intel Xeon @ 2.60 GHz physical cores and 156 GB RAM). A single analytical traffic assignment requires less than 5 seconds due to its simplicity, whereas a single simulation-based traffic assignment takes around 31 minutes.

5.6. Calibration platform description

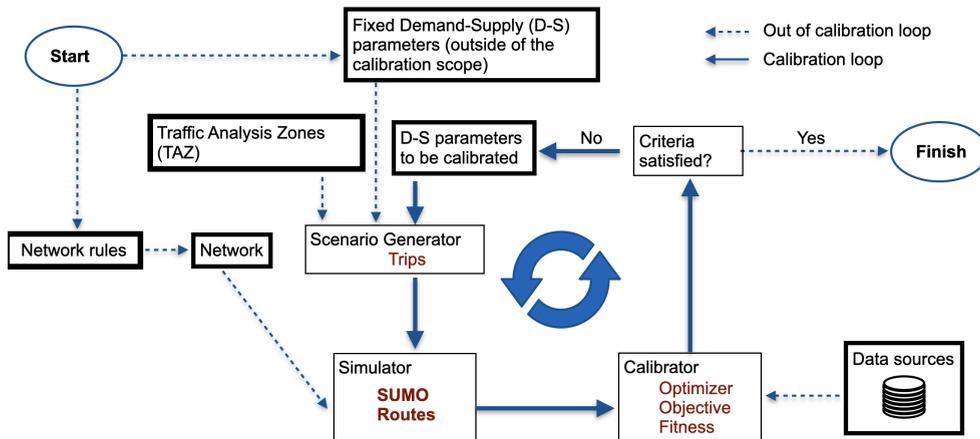


Figure 2: Calibration platform and SUMO simulator coupling in Python

We developed a Python-based platform for the sequential calibration of the demand and supply parameters of the large-scale mesoscopic traffic simulation in Simulation of Urban Mobility (SUMO) (Lopez et al., 2018). Figure 2 shows a schematic representation of the platform. Given the simulation inputs (simulation network, traffic analysis zones, detector locations) and parameters' priors, the platform calibrates the demand according to the proposed methodology. Other parameters that are fixed are, therefore, not part of calibration or outside of the scope of calibration. An initial OD matrix is used to generate trips between edges in different Traffic Analysis Zones (TAZs). The routing algorithm in SUMO assigns routes to these trips. We select a few supply parameters which influence traffic flow, junction delays, and route choice behavior. These parameters are defined below:

1. Automatic or online routing is used for the traffic assignment. According to SUMO (2023a), this routing approach works by giving some or all vehicles the capability to re-compute their route periodically based on the traffic conditions in the network. This kind of routing is also referred to as a “flexible one-shot assignment” (Castiglione et al., 2014). The parameters influencing the routing of vehicles are
 - (a) *re-routing probability*: The probability for a vehicle to have a routing device
 - (b) *re-routing period*: The period with which the vehicle shall be rerouted
 - (c) *re-routing adaptation steps*: The number of adaptation steps for averaging

Table 2
Enumeration of calibration parameters

Simulator → Data →	Analytical 1. Synthetic	Black box 2. Synthetic 3. Real	
Network parameters			
p	2500	5256	5256
T	3	5	5
ΔT	1 hour	1 hour	1 hour
m	500	1166	450
SPSA parameters			
γ	0.01	0.101	0.101
α	0.7	0.602	0.602
Range for c	(0.01, 10)	(0.01, 1)	(0.01, 1)
Range for a	$(1 \times 10^{-6}, 1 \times 10^{-3})$	$(1 \times 10^{-5}, 1 \times 10^{-3})$	$(1 \times 10^{-7}, 1 \times 10^{-2})$
K	100	50	50
W-SPSA weight parameters			
$w_{round-off}$	True	True	True
$w_{cut-off}$	0.01	0.01	0.01
Other parameters			
S		upto 5	
B		upto 100	
E		upto 20	

2. To influence the routing decision, the travel time of different types of edges can be scaled depending on their priority, using the parameter *edge priority factor* (SUMO, 2023c). As a consequence, low-priority edges will receive a penalty and have increased travel times, whereas high-priority edges receive little or no penalty.
3. The parameters which affect other delays (SUMO, 2023b) are
 - (a) *tls_travel-time_penalty*: This is a headway penalty to reduce the maximum flow across a signalized intersection.
 - (b) *meso_minor_penalty*: This is a fixed time penalty when passing a prioritized link.

We implement the W-SPSA by extending the Python SPSA implementation by Mayer (2017). All inputs pertaining to the network specification, count detectors, demand zones, SPSA parameters, etc, for three scenarios, are shown in Table 2. Values of SPSA parameters γ and α are fixed based on initial sensitivity analysis. We select SPSA gain coefficient a and perturbation c parameter for the automatic tuning module and thus $\phi = \{a, c\}$. Their search space is specified in Table 2. The complete platform is implemented using Python and is available on GitHub (see footnote on the front page).

6. Results

6.1. Automatic SPSA parameter Tuning (AST)

As discussed in Section 4.5, the automatic tuning procedure is solved as a hierarchical optimization process. The first step deploys W-SPSA to calibrate the pseudo demand parameters, while the second step uses Bayesian learning to fine-tune the SPSA parameters (a_k, c_k). For the Bayesian learning model, we use Matérn kernel as the Gaussian prior, and Upper Confidence Bound (UCB) as the acquisition function (Brochu et al., 2010). In all scenarios, we specified the parameter space for c as (1e-2, 1e1). The space for a is set to (1e-6, 1e0) for scenario 1, whereas it is set to (1e-7, 1e1) for scenarios 2 and 3. The points are randomly sampled on the logarithmic scale for initial probing, followed by Bayesian optimization. The number of iterations for initial probing/ exploration and number of iterations for Bayesian

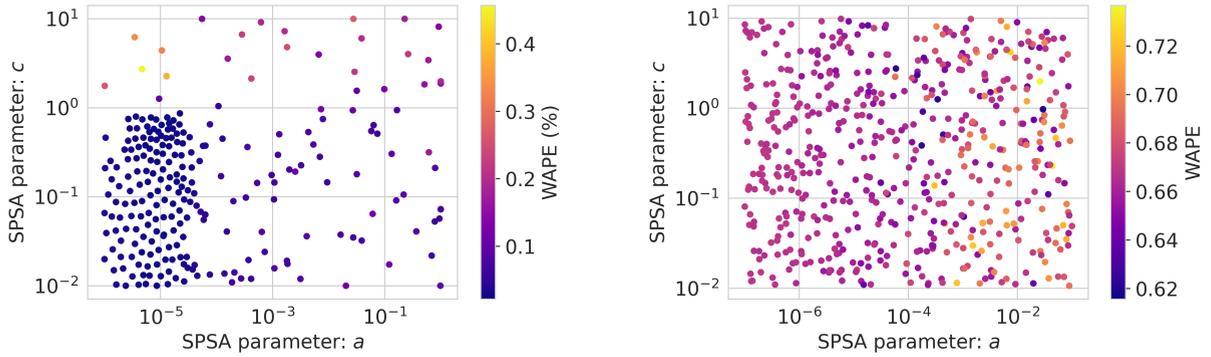


Figure 3: Automatic tuning of SPSA gain coefficients using Bayesian optimization for (left) scenario 1: synthetic simulator and (right) scenario 3: SUMO with real data

optimization was set to $\{50, 100\}$ for scenario 1, and $\{100, 200\}$ for scenarios 2 and 3. In Figure 3, we show the results of automatic SPSA parameter tuning for scenario 1 and scenario 3. The WAPE is lower for scenario 1 (scale of the color bar in Figure 3), compared to scenario 3 since the former involves synthetic data and the analytical simulator has a simpler loss surface without stochasticity. The approximated assignment matrix, in this case, is the same as the true assignment matrix, which is static. In scenario 1, we see that the points are initially probed randomly over the specified space of parameters during exploration, followed by a focused search based on the acquisition function. For scenario 1, we find that values of c and a in the ranges of $(1e-2, 1e0)$ and $(1e-6, 1e-4)$ are effective.

For scenario 3, the loss region is noisy due to errors from real data and analytical approximation of the assignment matrix in place of the actual simulator. This is why parameter combinations do not have a clear boundary of lower error and errors are also high. This stochasticity can be addressed by increasing the number of output averaging and SPSA replications at the cost of additional computation. Still, a fuzzy pattern is evident for c and a in the range of $(1e0, 1e1)$ and $(1e-5, 1e-4)$, where small errors are predominant. Thus, we conclude that automatic tuning of the SPSA parameters using an analytical approximation of the simulator is effective since these values reduce the most error. With these insights, the following scenarios 2 and 3 are instrumented with the above settings of the automatic tuning module.

6.2. Scenario 1: Synthetic data with analytical assignment

We show the results of the grid-based evaluation for bagging effectiveness in Figure 4. At lower levels of randomness (R^x in 30-40%), initial error in demand (X^a) and sensor counts (M^c) are about the same. At higher levels of R^x , the initial error in X^a (OD parameters) increases. During the initial increase in R^x for $R^x < 30\%$, there is a rapid increase in the error for higher values of B^x , whereas the error increase is gradual for smaller values of B^x . The gradual error increase continues for higher values of R^x in the case of lower B^x , but the error is stable for higher B^x . For the sensor counts, the initial error stabilizes or even drops with an increase in R^x . This is because counts are the weighted sum of the demand flows between respective OD zones. Thus, additional randomness in the ODs flows is canceled due to weighted summation. There is no strong correlation between the initial error in OD demand flows and corresponding counts in this range. Secondly, an increase in randomness cancels out the initial bias in some of the parameters and thus results in a small drop in the initial count WAPE.

We notice that W-SPSA can minimize the objective function in all cases of bias and randomness. This is because the sensor counts are used as MOP in the objective function and it is evident that the final count error is lower than the initial count error. Further, the total error computed by equally weighing the error in sensor counts and OD parameters is also lower. For low values of randomness (R^x), the error is dominated by the factor B^x . Results indicate that algorithm manages to correct even high bias values in OD parameters if R^x is small. This is why the initial and final total error gap is highest for low values of R^x .

The box plots in the middle column (Figure 4) show the WAPE of each individual estimate. The fitness of bagged OD estimates is consistently lower than the individual estimates in all cases, which supports the effectiveness of the bagging. However, the calibrated estimates are only better than the desired estimates for smaller values of R^x (0-20%)

in all the ranges of B^x . This observation implies that the algorithm can only move closer to the desired ODs X^* for lower values of the R^x . This is because, firstly, increased randomness in the initial estimates will deteriorate the structure of the initial demand specification and the quality of the domain specification of the demand parameters. At high randomness values, the initial point and domain misguide the calibration algorithm to a local optimum which is even far from the starting point resulting in higher error. Secondly, high R^x is not translating to a higher error in sensor counts due to the cancellation of the random errors. Thus, gradients relying on the sensor counts cannot effectively guide the reach of the desired demand parameters. The conclusion is that desired OD parameters are only recoverable when R^x is small since, at higher values, the essential structure of the X^* in X^a starts to disappear. However, the Bagging approach effectively improves the weighted fitness of both the demand and count parameters. Based on these findings, in the black box simulation experiments i.e., scenario 2 and scenario 3, we set the randomness values as $R^x=20\%$. This randomness value is similar to those used in the existing literature (Antoniou et al., 2016).

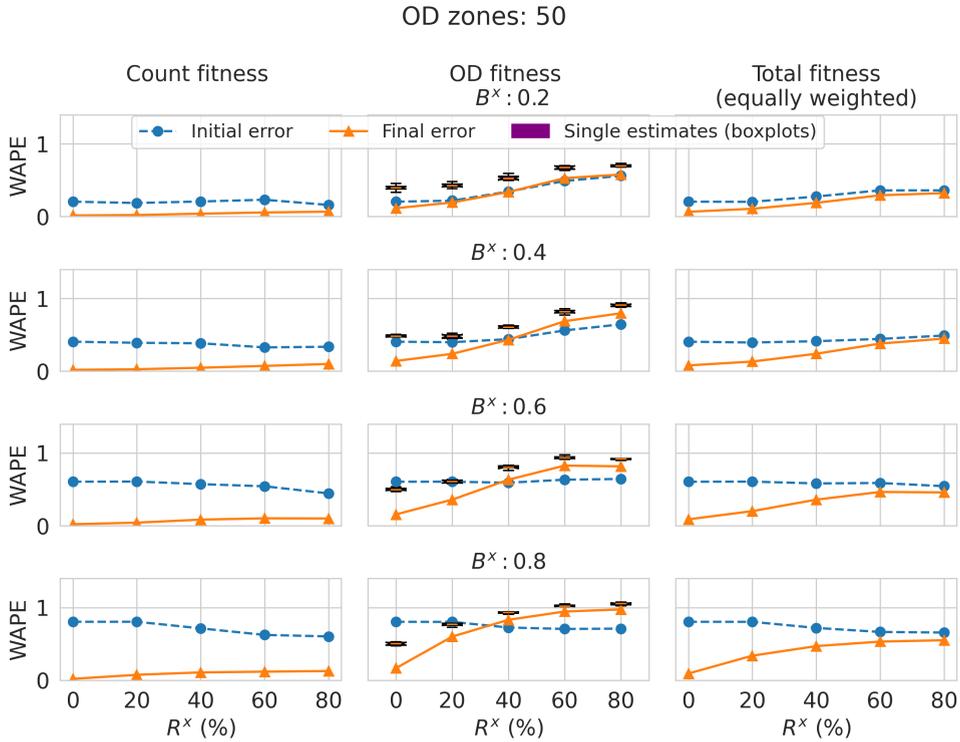


Figure 4: Scenario 1: Error at varying levels of B^x and R^x , for a Synthetic scenario with 50 OD zones. The *Final error* includes equally weighted sensor counts and OD demand estimates.

In Figure 5, we show the OD fitness error contours for single W-SPSA, SPA, and bagged estimates. Due to high dimensional optimization, fitness error is influenced by thousands of demand parameters. The plot shows the conditional error (because it depends on multiple parameters) region with the values of the pair of zones on X and Y-axes. The columns in this figure correspond to two levels of R^x : 30% and 90%, both at $B^x=0.6$. Fitness error increases with the increase in R^x . The single estimates are scattered in the region. However, the averaged estimates from SPA and bagging lie with the region of lower errors than the single W-SPSA estimates. Thus, bagging and SPA help reduce the variance in the estimates from single W-SPSA estimates.

We compare the performance of bagging and Stochastic Parameter Averaging (SPA) in Figure 6, where $B^x=0.6$ and $R^x=30\%$. Here we report bagged estimates from 20 W-SPSA runs, each for 100 iterations. Further, we also show the results of multiple SPA runs, each running for 2000 iterations. It is pointed out that function evaluations in bagging (with 20 different W-SPSA cold restarts, each running for 100 iterations) are equivalent to those in a single SPA run of 2000 iterations with warm restarts. Thus, the comparison between them is fair. The final count WAPE for individual W-SPSA estimates (column 1 in Figure 6) stops to reduce at 0.05 after a few iterations. On the other hand,

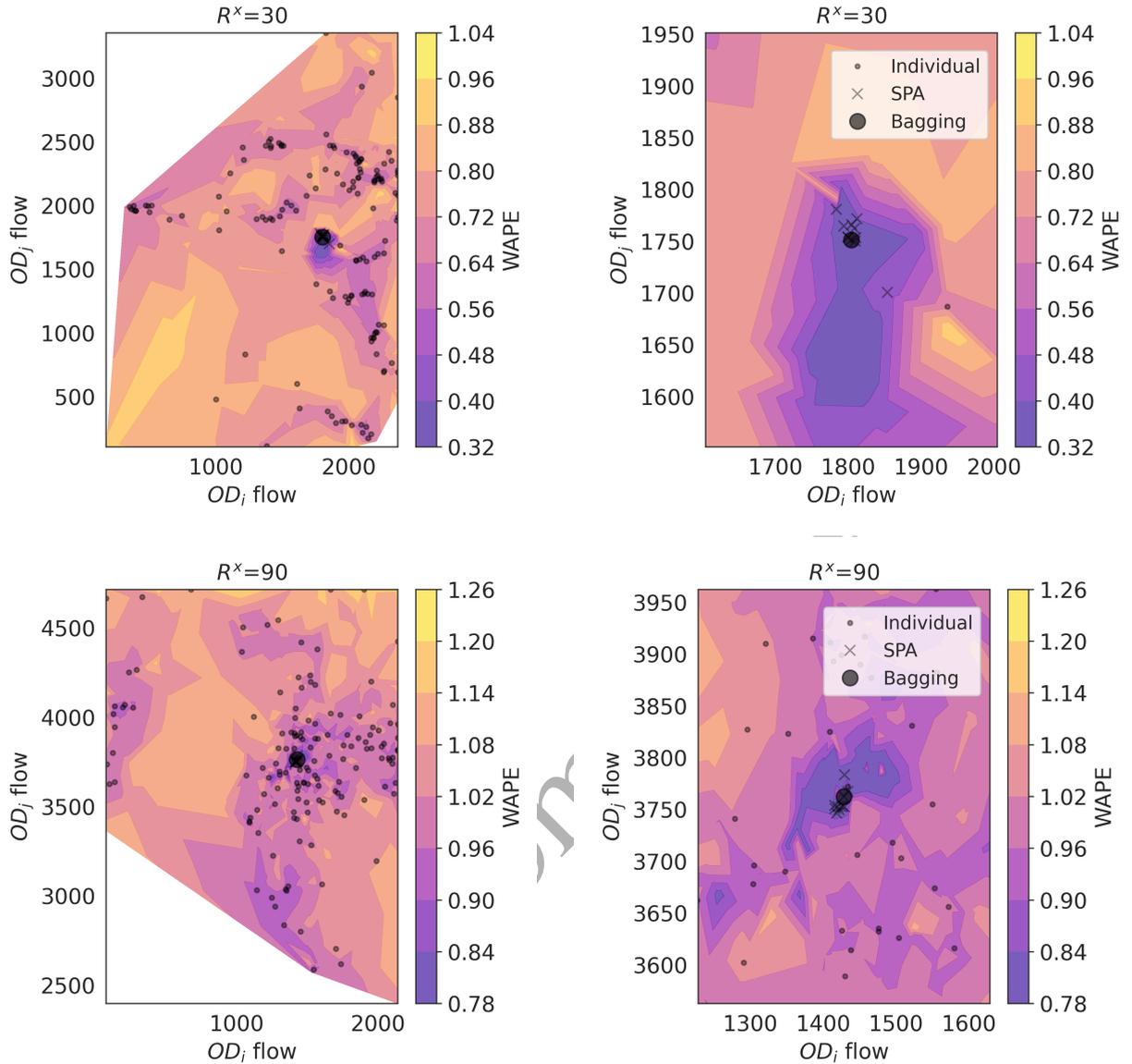


Figure 5: Contour plots showing the parameter values for selected pair of the zones with $B^x=0.6$, at different values of the R^x . It can be seen that bagged (●) or SPA (×) estimates lie in the lower error region as compared to the single SPSA iterate (•). The right column is the zoomed-in version of the plots in the left column. The plot shows the conditional error region with the selected two OD zone pairs on X and Y-axes

count fitness WAPE for SPA continues to reduce up to a value of more than 0.025. In the SPA loss curve, we see that each warm restart of the cyclic learning rate pushes the loss curve down faster than before the restart of the learning rate. Individual estimates achieve an OD fitness WAPE of 0.70, whereas individual SPA achieves a WAPE of about 0.55. We find that averaging helps improve the final W-SPSA solution, compared to the single solutions from each. Both bagging and SPA provide better OD estimates than the individual estimate from each W-SPSA run. However, the averaged estimates from bagging show superior performance with a WAPE of 0.38 compared to the averaged SPA estimates with a WAPE of 0.49. This implies that even though individual SPA estimates are more effective in fitting the counts and ODs than individual W-SPSA estimates, the averaged estimates of bagging are better than those of SPA. This could be because SPA prioritizes exploration around the initial local optima. If the initial local optimal is

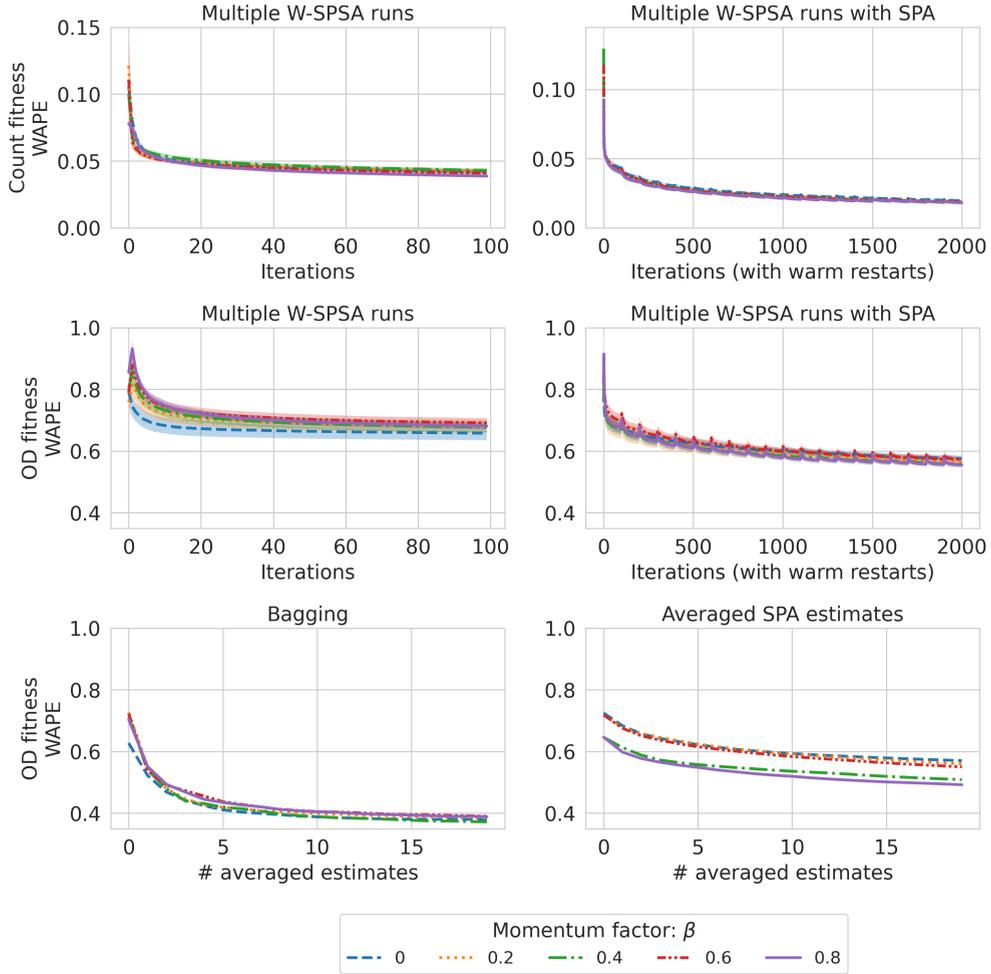


Figure 6: Scenario 1: OD and count fitness curves for bagging and stochastic parameter averaging with $B^x=0.6$ and $R^x=30\%$.

insufficient, SPA does not explore sufficiently due to over-fitting, and SPA averaging fails to reduce the variance. In the case of bagging, each estimate is obtained from exploration in a broader region. Thus, averaging the estimates has a superior result. The results of averaged estimates from bagging are not too sensitive to the momentum parameter β , as compared to those from the SPA. In bagging, five individual estimates reduce a significant part of the OD fitness error, whereas, for SPA, the error reduction is gradual. This implies that a small number of cold restarts as in bagging can give major benefits. Due to these reasons, we only used bagging or ensembling with cold restarts for the following experimentation.

We compare the performance of different components of our methodology for OD parameter fitness and sensor count fitness in Figure 7. In this case, we set $B^x = 0.8$, test the performance for values of N^x ranging from 10% to 200%, and show WAPE and RMSE. The approaches compared are *Bias correction (BC) using naive method*, *BC with weighted method*, *BC with W-SPSA*, *W-SPSA with bagging*, and *BC with W-SPSA and bagging*. Although a high randomness factor leads to higher corresponding errors, the problem becomes more challenging since the structure of the desired estimate is not identifiable from the initial matrix.

We find that the performance of the approaches depends on the R^x . All approaches with bias correction perform equally well at low randomness values. This is an interesting finding since it implies a simple and computationally inexpensive heuristic can achieve similar or better error performance as the W-SPSA optimization process for small randomness in the initial OD matrix. At $R^x > 40\%$, bagging performs better than the other approaches, specifically as seen from *W-SPSA with bagging*. This implies that for OD fitness, the bias-correction heuristic dominates at small

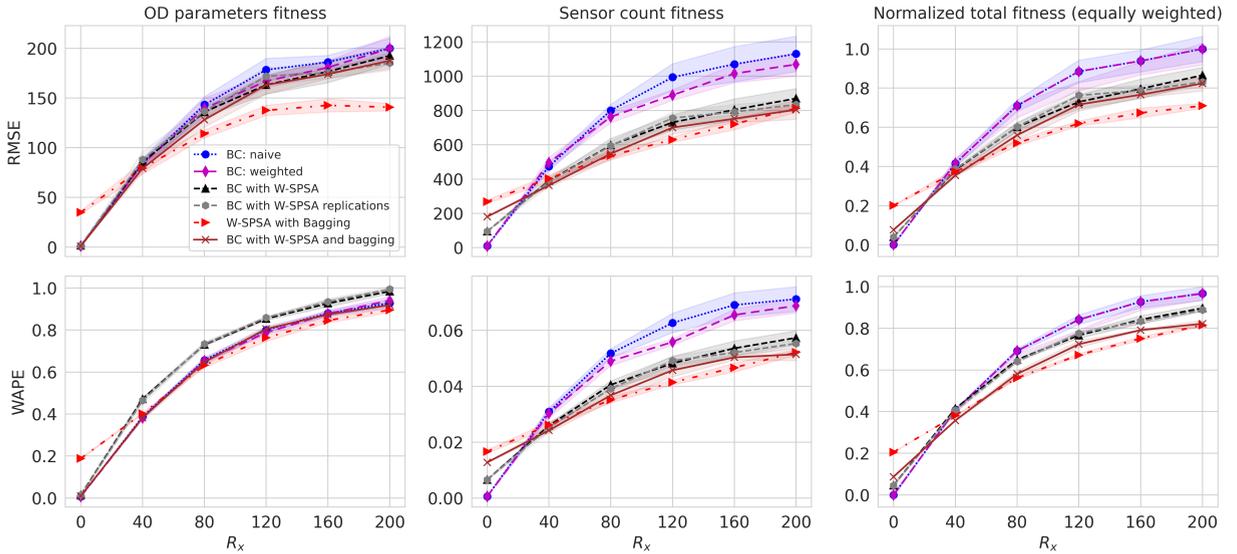


Figure 7: Scenario 1: OD and count fitness (RMSE and WAPE) sensitivity with the change in the randomness parameter, with different approaches using an analytical simulator ($B^x=0.8$)

randomness, whereas bagging dominates at high randomness. For high randomness, initial estimates are unreliable; thus, averaging multiple estimates helps provide better results. In the case of WAPE, OD fitness of *BC with W-SPSA without bagging* show high errors than just using *BC*. Intuitively, the SPSA model works better when the objective function has a clear descent direction. This is often the case when the objective function has a lower/ higher demand with respect to the true demand (Cantelmo et al., 2015). However, as the *BC* heuristic removes bias related to, e.g., overestimation or underestimation, the performance of W-SPSA may be affected.

Looking at the fitness for sensor counts, we find that W-SPSA outperforms simple heuristics in matching the sensor counts regarding both WAPE and RMSE. This is understandable since *BC* heuristics only adjust the OD parameters without ensuring consistency with the true sensor counts. Simple heuristics work equally well if the randomness in initial estimates is small ($20\% < R^x < 30\%$), meaning that initial estimates sufficiently capture the structure of the true estimates. The normalized total fitness shows that W-SPSA and bagging approaches achieve lower errors than the *BC* heuristics even in high randomness. Thus approaches using W-SPSA and bagging are best when ensuring the overall fitness of the counts and OD demand parameters. To speed up the convergence, *BC* can be used to adjust the initial values of the OD parameters, followed by W-SPSA with bagging to ensure consistency with the MOPs, such as counts.

6.3. Scenario 2: Munich scenario with SUMO simulator and synthetic data

We show the results of the calibration for the Munich scenario using the SUMO platform with synthetic counts and speeds in Table 3. The first set of results corresponds to $B^x = 0.6$ and a relatively smaller factor for randomness ($R^x = 20\%$) and uses only sensor counts or both sensor counts and link speeds in the objective function. We also add artificial randomness to the sensor counts to mirror data errors. We perform an ablation study by using one or more of the components of our methodology, namely W-SPSA (W), Bias Correction (BC), Automatic SPSA Tuning (A), and Bagging (B). The initial WAPE errors in count, speed, and OD are 0.42, 0.03, and 0.45, respectively. Similarly, the initial RMSE errors in count, speed, and OD are 288, 1.71, and 10.80, respectively. We define *baseline* as the scenario using sensor counts as MOP, with only W-SPSA, where count fitness WAPE is 0.14. The corresponding final speed and OD WAPE are 0.02 and 0.72, respectively. In this case, although counts and speeds fit better, the estimated OD is worse than the initial OD values. This is because in the objective function minimization, W-SPSA can converge to fit better to counts, but it lands in undesired local optima for the OD estimates, which is still away from the desired optima. Thus, individual estimates from W-SPSA have worse OD fitness due to induced randomness

Approach	Which MOPs in objective?		Count sensor noise	Final error value WAPE / RMSE		
	Count	Speed		Count	Speed	OD
Low noise ($B^x = 0.6$ & $N^x = 20$)						
Initial estimate	-	-	-	0.42 / 288.12	0.03 / 1.71	0.45 / 10.80
W ¹ (baseline)	Yes	No	0	0.14 / 89.34	0.02 / 0.85	0.72 / 19.75
BC	Yes	No	0	0.10 / 71.15	0.02 / 0.97	0.28 / 08.55
A-W-B	Yes	No	0	0.10 / 58.34	0.02 / 0.70	0.42 / 11.95
BC-A-W	Yes	No	0	0.11 / 72.96	0.02 / 1.00	0.63 / 18.81
BC-A-W-B	Yes	No	0	0.07 / 44.03	0.02 / 0.75	0.35 / 09.44
BC-A-W-B	Yes	Yes	0	0.11 / 81.75	0.02 / 0.89	0.41 / 11.35
BC-A-W-B	Yes	Yes	15	0.16 / 124.96	0.02 / 0.61	0.41 / 11.31
BC-A-W-B	Yes	Yes	30	0.28 / 230.30	0.02 / 0.72	0.46 / 14.41
BC-A-W-B	Yes	Yes	45	0.39 / 340.48	0.02 / 0.91	0.45 / 13.90
High noise ($B^x = 0.6$ & $N^x = 200$)						
BC	Yes	No	0	0.17 / 123.12	0.02 / 0.82	1.01 / 27.10
A-W-B	Yes	No	0	0.16 / 123.75	0.02 / 0.91	0.97 / 27.55
BC-A-W-B	Yes	No	0	0.14 / 95.15	0.02 / 1.06	1.03 / 30.26

¹ W: W-SPSA; BC: Bias-Correction; A: Automatic SPSA tuning; B: Bagging

Table 3

Scenario 2: Results (WAPE and RMSE) of the Munich scenario with synthetic data

in the parameters during the optimization path. When using only BC, the OD fitness, count fitness, and speed WAPE are 0.28, 0.10, and 0.02, respectively. When using W-SPSA with bagging (*A-W-B*), we obtain OD fitness of 0.42, whereas count and speed fitness are 0.10 and 0.02. Thus, we find that bagging helps to provide improved count and OD estimates over initial values as well as Baseline scenarios. In this case, we find speed and count fitness comparable to the *BC* approach. Using *BC-A-W* provides better results than the baseline in terms of improvement over count and OD fitness, but still, the estimated ODs are worse than the initial estimates in terms of WMAPE and RMSE. Adding Bagging helps to address this variance in the estimated OD parameters since the approaches *A-W-B* and *BC-A-W-B* have superior OD fitness than the baseline scenario. Only the latter approach outperforms the *BC* approach in terms of count fitness. This means that at small levels of randomness in the initial estimates, a simple heuristic such as *BC* can provide equal or better OD estimates than other approaches. However, we cannot simultaneously minimize fitness with respect to MOPs. This is why the combination of BC, W-SPSA, and Bagging helps to obtain the estimates while ensuring optimal fitness with respect to counts and speeds. For the given scenario, speed errors are low in all the cases and are not sensitive to the count errors/ approach used. This is possible because most of the network is uncongested. Therefore, they add little value to the calibration process.

When we add randomness to the sensor counts, we expect a reduction in fitness to the OD counts since the signal-to-noise ratio of the gradients from MOPs reduces. Therefore, for different sensor noise levels, we see a gradual reduction of OD fitness. Thus, the quality of sensor counts has important implications for the fitness of the OD parameters. Another finding is that in our experiments, using speeds in MOPs leads to higher errors in estimates as compared to using only counts in the MOPs. Since speed error is already low, they do not provide additional signals to the calibration process. W-SPSA essentially decomposes the original problem into multiple smaller sub-SPSA problems. By inclusion of speeds in the objective function, the number of MOPs increase, and due to the non-linear dependence between speeds and OD flows, the complexity of sub-SPSA problems also increase, leading to a drop in the accuracy of the estimates. However, speeds can provide additional context for better convergence in cases where the network is significantly congested. We suppose that the trade-off between additional context from speed data and complexity depends on the level and spread of congestion/ spill-back in the network and could be a matter of future research.

Then we set the OD randomness value to a high value ($N^x = 200\%$) to simulate situations where the initial demand estimates are of poor quality and, thus, the essential structure of the demand is lost. In the existing literature, such extreme scenarios are not considered and tested in OD estimation. We observe the adverse effect of using the

BC approach in these situations. This is because the BC approach is unreliable when the initial estimates have high random errors; thus, the bias correction is ineffective. Therefore in these cases, *A-W-B* gives the best fitness for OD parameters. Using *BC-A-W-B* provides the best count fitness in this case as well. However, the final estimates are still far from the desired values. When the initial estimates have high errors, there is little hope of recovering the desired estimates using the local search since the proposed methods will tend to converge to the local optima but far from the desired optima.

The effects of bagging on the calibrated OD estimates are shown in Figure 8. The two plots on top and bottom correspond to initial estimates with a good initial estimate (low randomness $R^x=20\%$) and poor initial estimates (high randomness $R^x=200\%$). Bagging can benefit both cases, as the OD fitness improves with the number of estimates used for averaging. We can see that averaging four individual estimates lead to most of the improvement in OD fitness. However, the final OD fitness errors are much lower than initial estimates with low random errors. The calibrated estimates in the case of bagging have lower variance, especially in case of low randomness, and is evident by calibrated estimates closer to the 45° line. Another interesting thing to note in Figure 8 is that even though OD parameters have a lot of scatter, counts have limited scatter. This implies that the variance in the OD parameters does not proportionally translate into variance in link counts since counts are the weighted sum of the OD flows. Thus, even if the ODs have significant random errors in case of poor estimates, the sensor counts will not have proportionately larger errors. Thus the optimization algorithm will struggle to converge to a local optima solution using only counts as MOP, which is undesirable. In case of poor estimates, the domain specification (l_x, u_x) also needs to be broad enough to include the desired solutions, which will further increase the complexity of the calibration and the possibility of more local optima. Thus, the quality of good initial estimates from auxiliary sources cannot be overstated in the case of OD estimation.

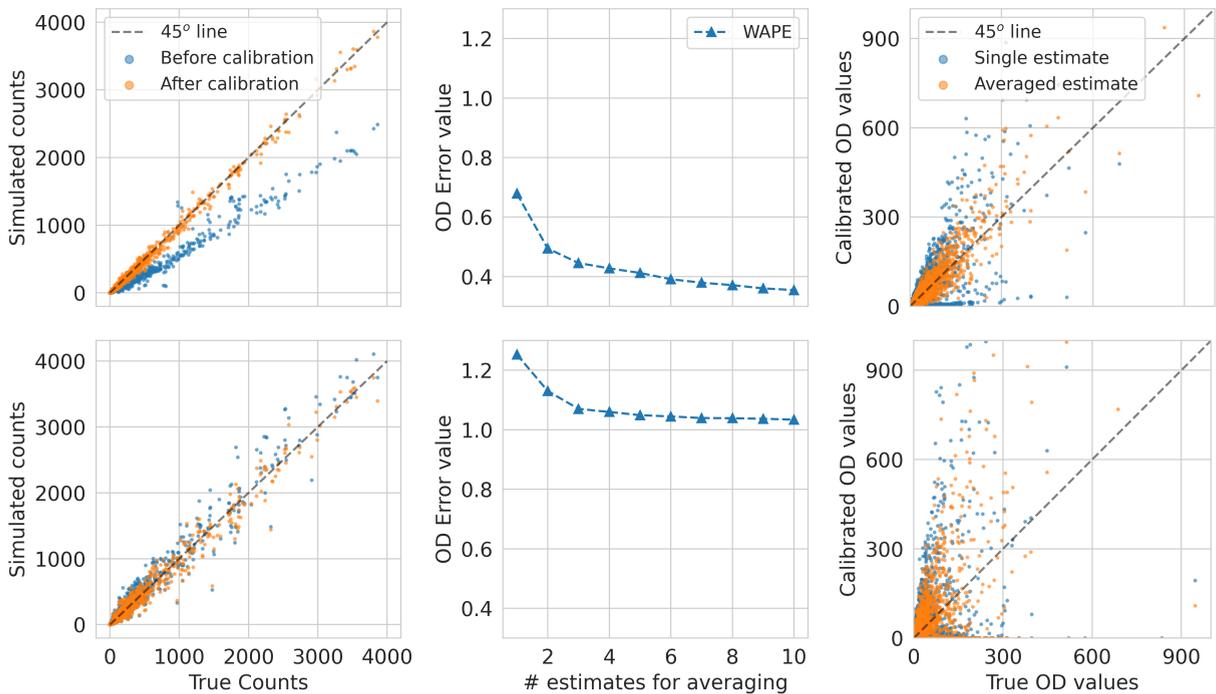


Figure 8: Scenario 2: Effects of bagging with initial estimates with (top) small randomness ($R^x = 20\%$) and (bottom) high randomness ($R^x = 200\%$)

6.4. Scenario 3: Munich scenario with SUMO and real-world data

This scenario requires a minimum of 400 function evaluations ($S = 2$, $E = 5$, $K = 10$, $B = 50$) or 5-6 days (reduced to 2-3 days if using parallelized W-SPSA and Bagging) to converge. However, these estimates can vary depending on the preciseness of the initial demand and supply parameters. Regressing the error with the supply parameters ($R^2=0.90$) shows that only *priority factor*, *meso-minor penalty*, *rerouting adaptation*, and *tls travel-time*

penalty are significant. We also visually inspect the error surface. Figure 9 shows the error surface with supply parameters. A *meso-minor penalty* of less than 10 gives optimal results. The optimal *tls travel-time penalty* is close to 1, and *rerouting adaptation* is less than 5. The optimal *priority factor* lies between 0.35 to 0.60, and *rerouting probability* lies between 0.40 to 0.50; however, lower values of the rerouting probability, such as close to 0.10 are also feasible, conditional on other parameters. Based on the results, we select values of flow penalty, travel-time penalty, and minor junction penalty are 0.57, 0.00, and 0.00, respectively. Rerouting probability, period, and adaptation interval are 0.10, 80, and 1, respectively. We see that multiple values of the combination of supply parameters give desired or good fitness of the sensor counts. This points to the fact that additional MOPs from other data sources, such as inter-zone travel times, queue lengths, trajectory data, and travel speeds, should be considered for the further calibration of these parameters.

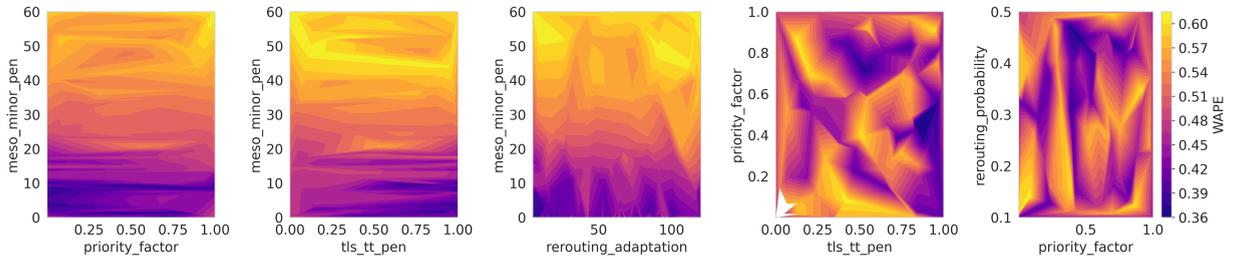


Figure 9: Scenario 3: Fitness or error surface with the supply parameters

Figure 10 shows the plot of simulated and observed link sensor counts during 0700-1000 at different stages of sequential demand and supply calibration. Before demand calibration, the scatter plot is not centered around a 45-degree line for the counts on *other* link types (trunk and primary links), which implies room for improvement. WAPE for *other* links ranges between 1.49 and 1.74. After demand calibration, WAPE for federal (motorway links) is in the range of 0.24-0.40 for a time interval of 0700-1000 hours. WAPE for other links (trunk and primary links) is in the 0.51-0.53 for the same time interval. Overall WAPE varies between 0.39-0.47, which is lower than the corresponding WAPE before calibration. Simulated counts for federal links during 0800-1000 hours are lower than the corresponding observed counts. After supply calibration, WAPE for federal (motorway) links ranges from 0.19-0.25 for 0700-1000 hours. WAPE for other links (trunk and primary links) ranges from 0.40-0.48 for the same time interval. The overall WAPE varies between 0.32-0.37. Calibration of supply parameters substantially reduces the overall error. The improved match for the federal links during the 0800-1000 is also evident.

We also show the hourly link volumes (Figure 11) on the network for the 0800-0900 hour, highlighting the comparison between the uncalibrated and final calibrated models. The difference between the distribution of the flows between the two cases is evident. In the uncalibrated model, there is lesser traffic on the links corresponding to the outer Autobahn ring road (German translation: Äußerer Ring), as well as the middle ring road (Mittlerer Ring), whereas the share of traffic on inner city links is higher. This points to lower impedance on inner roads, so a major share of the traffic selects the routes through these links for their trips. On the contrary, in the calibrated model, traffic distribution is consistent with the observed counts, with a major chunk of trips routed through the outer ring, middle ring roads, and major radial roads. In Figure 12, we compare the uncalibrated, calibrated, and observed link speeds in the network. The changes in the speeds between uncalibrated and calibrated models show that certain links (in red) in the former model were congested but not in the latter. Further, we see a reasonable match of link speeds between observed data and calibrated model.

7. Conclusion

This work presented an end-to-end sequential approach for demand (OD estimation) and supply calibration. Our approach has components automating certain aspects, such as SPSA and tuning of supply parameters. We achieved SPSA tuning using the Bayesian optimization algorithm and tackled bias and variance in the initial estimates by proposing methods for each. We proposed a bias correction heuristic to correct the initial bias and thus reduce the burden on the following optimization algorithm, i.e., W-SPSA. W-SPSA will stop improving the errors after most of the overall bias has been corrected, i.e., beyond the limit where noise starts to dominate. This happens due to the cancellation of

Automating traffic simulation model calibration

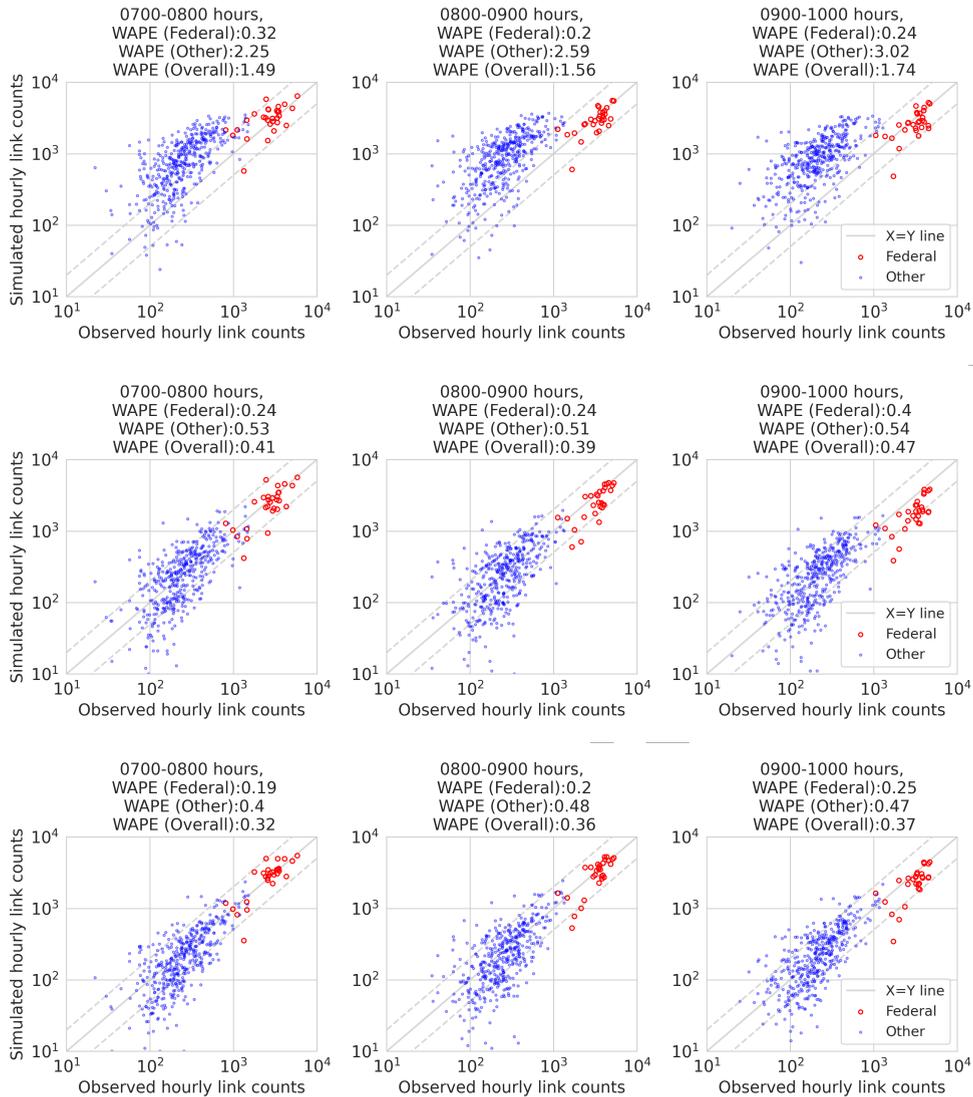


Figure 10: Scenario 3: Fitness of link sensor counts before demand calibration (top), after demand calibration (middle), and followed by supply calibration (bottom). Data corresponding to federal links (motorways) and other links (trunk, primary and secondary links) are highlighted in red and blue, respectively. The Centre line is 45°, or the $Y=X$ line, and the lower and upper dotted lines are at $Y=X/2$ and $Y=2X$, respectively.

the random gradients dominated by the noise. We applied ensembling with bagging and SPA, and obtained estimates that are precise compared to the estimates from W-SPSA without averaging. We show that averaging estimates can be better than individual estimates, subject to the quality of the initial solution. We find that BC-A-W-B provides the best fit of counts in both low and high-noise scenarios with simulation-based assignments. In low-noise scenarios, BC works well to fit ODs and counts (second to BC-A-W-B), but in high-noise scenarios, an approach with bagging provides a better fit. If the information from speed data is not conflicting with that from count data, then using them does not lead to additional benefits or even a reduction in accuracy. Further, in high randomness scenarios, count data is insufficient for reliable OD estimation.

Practically, the advantage of bagging is that it can be in parallel, and thus, with parallel compute nodes, it does not cause substantial time overhead. Our approach can help modelers to calibrate their simulation models with little manual effort. By releasing the codes, we also make a practical contribution to OD estimation; there is a large gap between

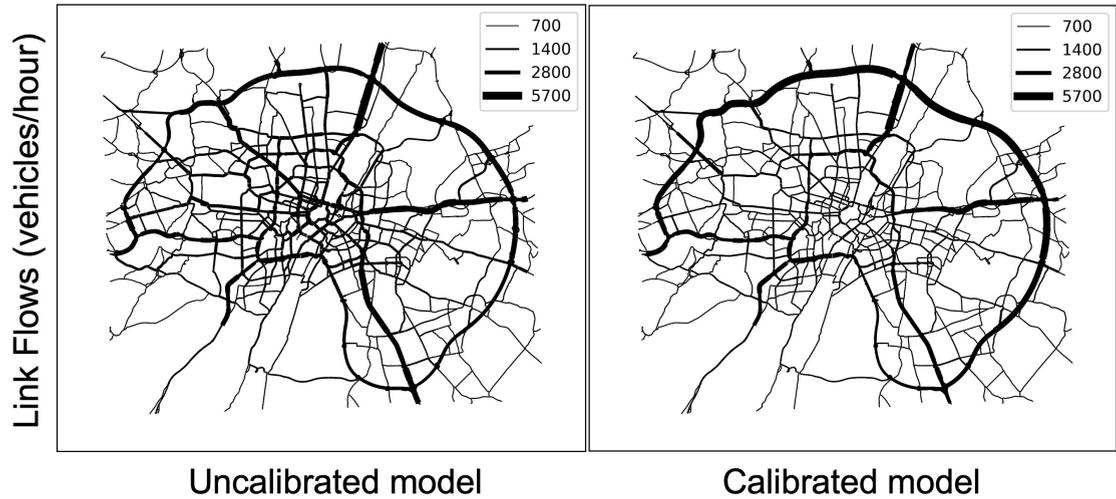


Figure 11: Scenario 3: Simulated link flows and speeds during 0800-0900 hours (left) before calibration and (right) after calibration.

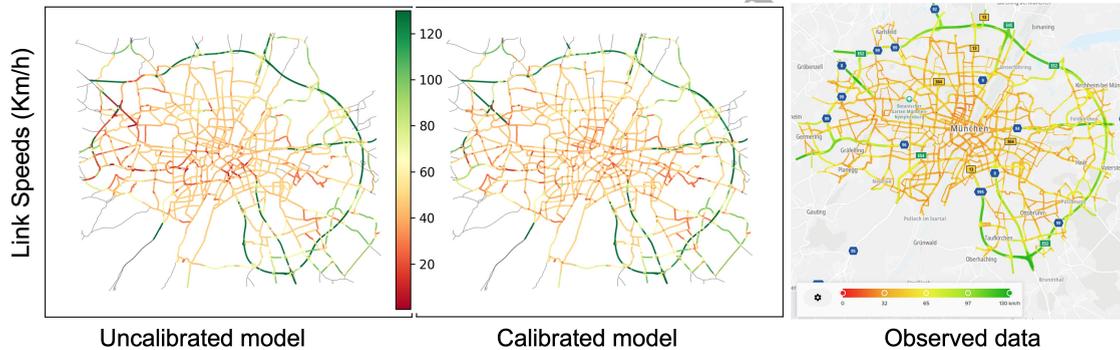


Figure 12: Scenario 3: Simulated link speeds during 0800-0900 hours (left) before calibration, (middle) after calibration, and (right) observed data (source: TomTom).

literature and open-source tools. An additional advantage of proposed ensembles is that they can be efficiently used without parallel computing, which can be useful in practice (e.g., the number of licenses for 'commercial software' often limits parallel computing in practice).

Future works should explore the transferability of the proposed approach and its derived conclusions to other real networks. Our proposed OD estimation framework can be further augmented with any auxiliary OD demand data sources in the objective function. Also, using additional data sources for MOPs will help reap additional benefits, especially in real scenarios where true or global parameters are unknown. In these cases, W-SPSA may need to be adapted according to the data source to reap benefits. For instance, the weight matrix based on the link assignment matrix may not be the best choice for non-linear variables such as speed and thus need further enhancements. Ensembling techniques such as bagging and SPA have proven effective in machine learning and thus should be explored for other simulation-based optimization problems. The location of sensors can influence the quality of the estimated ODs. This is related to the coverage or network observability the sensors provide. In our calibration experiments using synthetic or analytical simulators, we use multiple random variations of detector configurations for each run and thus help to tackle the variance due to such sensor location settings. For the experiments with SUMO simulators, doing this is computationally expensive, and we consider investigation of this aspect beyond the scope of this paper and a matter of future work. Methodological components such as BC are specific to traffic count data, and thus they cannot be applied when such data are unavailable for calibration. Future work can be done to apply Probe Vehicle data (Antoniou et al.,

2004) or Speed data for initial bias adjustment.

Experiments with the ensembling aspects, such as different types of gain coefficient restart techniques use of intermediate estimates during each cycle are also interesting avenues. Although we used a mesoscopic simulation model to ease the computation burden, the proposed methodological components are simulator agnostic. Theoretically, our approach can also be applied to micro-simulation models in future works. Ensemble methods should also be explored for application to calibrate parameters, even in car-following or lane-changing models. Still, there could be some practical challenges with micro-simulation, such as more number of parameters and their sensitivity to the measurements. Further, the developed framework should be tested for application to online calibration where the fluctuations in the demand and traffic flow are prominent and challenging to handle. A unique calibration parameter set is not guaranteed in stochastic simulations involving high-dimensional inputs. The possibility of a multiple-parameter set arises from the unobservable/ indeterminate system, wherein many solutions for given conditions are possible. However, some of these parameters can be practically reasonable in real-world scenarios due to the stochasticity of the system. Thus, having a single set of parameters but their distribution is insufficient. Here, multiple estimates during ensembling cycles can also be used to quantify the uncertainty in parameters.

8. Acknowledgement

We thank the SUMO developers for actively responding to our queries on the mailing list and GitHub. We also acknowledge the city of Munich for sharing the sensor count data. Finally, the authors gratefully acknowledge the computational and data resources provided by the Leibniz Supercomputing Centre (www.lrz.de). This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under Grant 415208373 for Project TraMPA-Transport Modelling using Publicly Available Data.

9. Declarations of interest

No potential conflict of interest was reported by the author(s).

CRedit authorship contribution statement

Vishal Mahajan: Conceptualization of this study, Data curation, Methodology, Software, Writing - original draft. **Guido Cantelmo:** Conceptualization of this study, Methodology, Writing - review & editing. **Constantinos Antoniou:** Conceptualization of this study, Methodology, Writing - review & editing.

References

- Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., et al., 2016. Towards a generic benchmarking platform for origin-destination flows estimation/updates algorithms: Design, demonstration and validation. *Transportation Research Part C: Emerging Technologies* 66, 79–98.
- Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2004. Incorporating automated vehicle identification data into origin-destination estimation. *Transportation Research Record* 1882, 37–44. URL: <https://doi.org/10.3141/1882-05>, doi:10.3141/1882-05, arXiv:<https://doi.org/10.3141/1882-05>.
- Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2005. Online calibration of traffic prediction models. *Transportation Research Record* 1934, 235–245. URL: <https://doi.org/10.1177/0361198105193400125>, doi:10.1177/0361198105193400125, arXiv:<https://doi.org/10.1177/0361198105193400125>.
- Antoniou, C., Lima Azevedo, C., Lu, L., Pereira, F., Ben-Akiva, M., 2015. W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Part C: Emerging Technologies* 59, 129–146. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X15001710>, doi:<https://doi.org/10.1016/j.trc.2015.04.030>. special Issue on International Symposium on Transportation and Traffic Theory.
- Balakrishna, R., Ben-Akiva, M., Koutsopoulos, H.N., 2007. Offline calibration of dynamic traffic assignment: Simultaneous demand-and-supply estimation. *Transportation Research Record* 2003, 50–58. URL: <https://doi.org/10.3141/2003-07>, doi:10.3141/2003-07, arXiv:<https://doi.org/10.3141/2003-07>.
- BAST: Bundesanstalt für Straßenwesen, 2023. Automatic permanent counting points: raw data. <https://www.bast.de/DE/Publikationen/Daten/Verkehrstechnik/DZ.html?nn=1954870>. Accessed: 2023-02-01.
- Ben-Akiva, M.E., Gao, S., Wei, Z., Wen, Y., 2012. A dynamic traffic assignment model for highly congested urban networks. *Transportation Research Part C: Emerging Technologies* 24, 62–82. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X12000186>, doi:<https://doi.org/10.1016/j.trc.2012.02.006>.
- Breiman, L., 1996. Bagging predictors. *Machine Learning* 24, 123–140. URL: <https://doi.org/10.1007/BF00058655>, doi:10.1007/BF00058655.

- Brochu, E., Cora, V.M., de Freitas, N., 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. URL: <https://arxiv.org/abs/1012.2599>, doi:10.48550/ARXIV.1012.2599.
- Buisson, C., Daamen, W., Punzo, V., Wagner, P., Montanino, M., Ciuffo, B., 2014. Chapter 4: Calibration and validation principles, in: Daamen et al. (2014). pp. 89–118. doi:10.1201/b17440.
- Bunge, M., 1963. A general black box theory. *Philosophy of Science* 30, 346–358. URL: <http://www.jstor.org/stable/186066>. full publication date: Oct., 1963.
- Cantelmo, G., Cipriani, E., Gemma, A., Nigro, M., 2014a. An adaptive bi-level gradient procedure for the estimation of dynamic traffic demand. *IEEE Transactions on Intelligent Transportation Systems* 15, 1348–1361.
- Cantelmo, G., Viti, F., Cipriani, E., Nigro, M., 2015. A two-steps dynamic demand estimation approach sequentially adjusting generations and distributions, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, IEEE. pp. 1477–1482.
- Cantelmo, G., Viti, F., Cipriani, E., Nigro, M., 2018. A utility-based dynamic demand estimation model that explicitly accounts for activity scheduling and duration. *Transportation Research Part A: Policy and Practice* 114, 303–320.
- Cantelmo, G., Viti, F., Tampère, C.M., Cipriani, E., Nigro, M., 2014b. Two-step approach for correction of seed matrix in dynamic demand estimation. *Transportation Research Record* 2466, 125–133.
- Cascetta, E., 2001. Modeling transportation systems: Preliminary concepts and application areas, in: *Transportation Systems Engineering: Theory and Methods*. Springer New York, NY, pp. 18–45. doi:10.1007/978-1-4757-6873-2.
- Cascetta, E., Inaudi, D., Marquis, G., 1993. Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science* 27, 363–373. URL: <https://doi.org/10.1287/trsc.27.4.363>, doi:10.1287/trsc.27.4.363, arXiv:<https://doi.org/10.1287/trsc.27.4.363>.
- Cascetta, E., Papola, A., Marzano, V., Simonelli, F., Vitiello, I., 2013. Quasi-dynamic estimation of O-D flows from traffic counts: Formulation, statistical validation and performance analysis on real data. *Transportation Research Part B: Methodological* 55, 171–187.
- Castiglione, J., Bradley, M., Gliebe, J., 2014. Activity-Based Travel Demand Models: A Primer. Technical Report. TRANSPORTATION RESEARCH BOARD. Washington, DC. doi:10.17226/22357.
- Chiu, Y.C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., Hicks, J., 2011. Dynamic traffic assignment: A primer. URL: <https://onlinepubs.trb.org/onlinepubs/circulars/ec153.pdf>.
- Ciuffo, B., Punzo, V., Montanino, M., 2014. Global sensitivity analysis techniques to simplify the calibration of traffic simulation models. methodology and application to the idm car-following model. *IET Intelligent Transport Systems* 8, 479–489. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2013.0064>, doi:<https://doi.org/10.1049/iet-its.2013.0064>, arXiv:<https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2013.0064>.
- Daamen, W., Buisson, C., Hoogendoorn, S.P. (Eds.), 2014. *Traffic Simulation and Data: Validation Methods and Applications*. CRC Press, Boca Raton, FL (U.S.). doi:10.1201/b17440.
- Dieterich, T.G., 2000. Ensemble methods in machine learning, in: *Multiple Classifier Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 1–15.
- Djukic, T., Lint, J.W.C.V., Hoogendoorn, S.P., 2012. Application of principal component analysis to predict dynamic origin–destination matrices. *Transportation Research Record* 2283, 81–89. URL: <https://doi.org/10.3141/2283-09>, doi:10.3141/2283-09, arXiv:<https://doi.org/10.3141/2283-09>.
- Frederix, R., Viti, F., Tampère, C.M., 2013. Dynamic origin–destination estimation in congested networks: theoretical findings and implications in practice. *Transportmetrica A: Transport Science* 9, 494–513.
- He, L., Ishibuchi, H., Trivedi, A., Wang, H., Nan, Y., Srinivasan, D., 2021. A survey of normalization methods in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 25, 1028–1048. doi:10.1109/TEVC.2021.3076514.
- Ho, M.C., Lim, J.M.Y., Chong, C.Y., Chua, K.K., Siah, A.K.L., 2023. High dimensional origin destination calibration using metamodel assisted simultaneous perturbation stochastic approximation. *IEEE Transactions on Intelligent Transportation Systems*, 1–10doi:10.1109/TITS.2023.3234615.
- Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopperoff, J.E., Weinberger, K.Q., 2017. Snapshot ensembles: Train 1, get m for free. URL: <https://arxiv.org/abs/1704.00109>, doi:10.48550/ARXIV.1704.00109.
- Izmailov, P., Podoprikin, D., Garipov, T., Vetrov, D., Wilson, A.G., 2018. Averaging weights leads to wider optima and better generalization. URL: <https://arxiv.org/abs/1803.05407>, doi:10.48550/ARXIV.1803.05407.
- Kolassa, S., Schütz, W., 2007. Advantages of the mad/mean ratio over the mape. *Foresight: The International Journal of Applied Forecasting*, 40–43URL: <https://EconPapers.repec.org/RePEc:for:ijafaa:y:2007:i:6:p:40-43>.
- Kostic, B., Gentile, G., Antoniou, C., 2017. Techniques for improving the effectiveness of the SPSA algorithm in dynamic demand calibration, in: 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 368–373. doi:10.1109/MTITS.2017.8005699.
- Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using sumo, in: The 21st IEEE International Conference on Intelligent Transportation Systems, IEEE. pp. 2575–2582. URL: <https://elib.dlr.de/127994/>.
- Loshchilov, I., Hutter, F., 2016. Sgdr: Stochastic gradient descent with warm restarts. URL: <https://arxiv.org/abs/1608.03983>, doi:10.48550/ARXIV.1608.03983.
- Lu, L., Xu, Y., Antoniou, C., Ben-Akiva, M., 2015. An enhanced SPSA algorithm for the calibration of dynamic traffic assignment models. *Transportation Research Part C: Emerging Technologies* 51, 149–166. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X14003295>, doi:<https://doi.org/10.1016/j.trc.2014.11.006>.
- Ma, T., Abdulhai, B., 2002. Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters. *Transportation Research Record* 1800, 6–15. URL: <https://doi.org/10.3141/1800-02>, doi:10.3141/1800-02, arXiv:<https://doi.org/10.3141/1800-02>.

- Mayer, A., 2017. Noisyopt: A python library for optimizing noisy functions. *Journal of Open Source Software* 2, 258. URL: <https://doi.org/10.21105/joss.00258>, doi:10.21105/joss.00258.
- Moeckel, R., Kuehnel, N., Llorca, C., Moreno, A.T., Rayaprolu, H., 2020. Agent-based simulation to improve policy sensitivity of trip-based models. *Journal of Advanced Transportation* 2020, 1902162. URL: <https://doi.org/10.1155/2020/1902162>, doi:10.1155/2020/1902162.
- Omrani, R., Kattan, L., 2012. Demand and supply calibration of dynamic traffic assignment models: Past efforts and future challenges. *Transportation Research Record* 2283, 100–112. URL: <https://doi.org/10.3141/2283-11>, doi:10.3141/2283-11, arXiv:<https://doi.org/10.3141/2283-11>.
- Omrani, R., Kattan, L., 2018. Concurrent estimation of origin-destination flows and calibration of microscopic traffic simulation parameters in a high-performance computing cluster. *Journal of Transportation Engineering, Part A: Systems* 144, 04017068. URL: <https://ascelibrary.org/doi/abs/10.1061/JTEPBS.0000093>, doi:10.1061/JTEPBS.0000093, arXiv:<https://ascelibrary.org/doi/pdf/10.1061/JTEPBS.0000093>.
- Osorio, C., 2019. Dynamic origin-destination matrix calibration for large-scale network simulators. *Transportation Research Part C: Emerging Technologies* 98, 186–206. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X18305357>, doi:<https://doi.org/10.1016/j.trc.2018.09.023>.
- Polyak, B.T., Juditsky, A.B., 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* 30, 838–855. URL: <https://doi.org/10.1137/0330046>, doi:10.1137/0330046, arXiv:<https://doi.org/10.1137/0330046>.
- Prakash, A.A., Seshadri, R., Antoniou, C., Pereira, F.C., Ben-Akiva, M.E., 2017. Reducing the dimension of online calibration in dynamic traffic assignment systems. *Transportation Research Record* 2667, 96–107. URL: <https://doi.org/10.3141/2667-10>, doi:10.3141/2667-10, arXiv:<https://doi.org/10.3141/2667-10>.
- Qurashi, M., Lu, Q.L., Cantelmo, G., Antoniou, C., 2022. Dynamic demand estimation on large scale networks using principal component analysis: The case of non-existent or irrelevant historical estimates. *Transportation Research Part C: Emerging Technologies* 136, 103504. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X21004903>, doi:<https://doi.org/10.1016/j.trc.2021.103504>.
- Qurashi, M., Ma, T., Chaniotakis, E., Antoniou, C., 2020. PC-SPSA: Employing dimensionality reduction to limit spsa search noise in dta model calibration. *IEEE Transactions on Intelligent Transportation Systems* 21, 1635–1645. doi:10.1109/TITS.2019.2915273.
- Robbins, H., Monro, S., 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* 22, 400–407. URL: <http://www.jstor.org/stable/2236626>.
- Ruder, S., 2016. An overview of gradient descent optimization algorithms. *CoRR abs/1609.04747*. URL: <http://arxiv.org/abs/1609.04747>, arXiv:1609.04747.
- Ruppert, D., 1988. Efficient estimations from a slowly convergent robbins-monro process. *Cornell University Operations Research and Industrial Engineering*.
- Spall, J., 1998a. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems* 34, 817–823. doi:10.1109/7.705889.
- Spall, J., Cristion, J., 1994. Nonlinear adaptive control using neural networks: estimation with a smoothed form of simultaneous perturbation gradient approximation, in: *Proceedings of 1994 American Control Conference - ACC '94*, pp. 2560–2564 vol.3. doi:10.1109/ACC.1994.735021.
- Spall, J.C., 1998b. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins Apl Technical Digest* 19, 482–492.
- Spall, J.C., 2003. Stochastic Approximation for Nonlinear Root-Finding. John Wiley & Sons, Ltd. chapter 4. pp. 95–125. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/0471722138.ch4>, doi:<https://doi.org/10.1002/0471722138.ch4>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471722138.ch4>.
- SUMO, 2023a. Automatic routing. URL: https://sumo.dlr.de/docs/Demand/Automatic_Routing.html. accessed on 29.03.2023.
- SUMO, 2023b. Meso. URL: <https://sumo.dlr.de/docs/Simulation/Meso.html#tls-penalty>. accessed on 29.03.2023.
- SUMO, 2023c. Routing. URL: https://sumo.dlr.de/docs/Simulation/Routing.html#routing_by_traveltime_and_edge_priority. accessed on 29.03.2023.
- Toledo, T., Kolechkina, T., Wagner, P., Ciuffo, B., Lima Azevedo, C., Marzano, V., Flötteröd, G., 2014. Network model calibration studies, in: Daamen et al. (2014). p. 22. doi:10.1201/b17440.
- Tympakianaki, A., Koutsopoulos, H.N., 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: Emerging Technologies* 55, 231–245. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X15000248>, doi:<https://doi.org/10.1016/j.trc.2015.01.016>.
- Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2018. Robust SPSA algorithms for dynamic od matrix estimation. *Procedia Computer Science* 130, 57–64. URL: <https://www.sciencedirect.com/science/article/pii/S1877050918303624>, doi:<https://doi.org/10.1016/j.procs.2018.04.012>. the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.