

## Research Article

# Network-Wide Calibration of Link Capacities for Dynamic Traffic Assignment Models

Guang Wei <sup>1</sup>, Clas Rydergren <sup>1</sup>, David Gundlegård <sup>1</sup>, Joakim Ekström <sup>1</sup>,  
 and Gunnar Flötteröd <sup>1,2</sup>

<sup>1</sup>Department of Science and Technology, Linköping University, Norrköping 60174, Sweden

<sup>2</sup>Department of Society, Environment and Transport, Swedish National Road and Transport Research Institute, Stockholm 11428, Sweden

Correspondence should be addressed to Guang Wei; [guang.wei@liu.se](mailto:guang.wei@liu.se)

Received 28 June 2024; Revised 22 February 2025; Accepted 11 June 2025

Academic Editor: Peter J. Jin

Copyright © 2025 Guang Wei et al. Journal of Advanced Transportation published by John Wiley & Sons Ltd. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Dynamic traffic assignment (DTA) models are used in many transportation planning and traffic management scenario analyses today. The aim of the DTA model is to reproduce the pattern of vehicular movements. DTA models require inputs in terms of demand and capacity of the road network and are very challenging to calibrate for large urban networks. In this paper, a new network-wide calibration method for link capacities in urban networks is proposed. The method takes link flow observations for a subset of the links in the network to estimate the link capacities. The proposed method relies on partial least squares (PLS) regression and is demonstrated to be feasible and efficient in an urban road network (Stockholm, Sweden) compared to the simultaneous perturbation stochastic approximation (SPSA) method. Performance analysis of the proposed method for different amounts of link flow observations shows that it performs favorably for the cases in which only a small percentage of link flow observations is given.

**Keywords:** calibration; dynamic traffic assignment; partial least squares regression; road capacity

## 1. Introduction

Dynamic traffic assignment (DTA) models are important tools for planning and managing large-scale urban networks. Calibration of DTA models of large urban networks includes several challenges. Accurately replicating real-world traffic conditions, especially in highly congested areas, requires sophisticated models that can handle complex interactions. Many characteristics of urban networks, such as complicated intersections, short links, and route choice are challenging to model, and are factors influencing the calibration [1, 2]. Further, large urban networks involve numerous variables and parameters, making the calibration process computationally demanding [3].

The precision of congestion and delay estimates of DTA models rely heavily on correctly estimated capacity

parameters. Link capacities, which describe the physical property of links, are assumed fixed in this paper if we ignore factors such as weather conditions. Link capacities in dynamic urban traffic networks are typically estimated locally for each link or assigned based on general guidelines [4]. Local estimation of link capacities requires assumptions on the fundamental diagram of each link as well as flow measurements for several traffic regimes with densities both higher and lower than the critical density. In this paper, we aim to utilize the network structure in combination with link flow measurements to perform network-wide calibration of link capacities.

Most research regarding link capacity calibration focuses on the local level instead of the network level. Local calibration is done by calibrating fundamental diagrams for individual links, or individual link types [5–7]. Other related

research deals with calibration of demand, that is, origin-destination (OD) calibration, and capacity calibration simultaneously. Here, the OD matrix acts as parameters and flows are outputs. Capacities appear in the constraint part of an optimization problem which minimizes the difference between the predicted link flows obtained when the OD matrix is assigned to the network and observed link flows [8, 9]. There is very limited work focusing on quantitative analysis of capacities in urban networks as well as pure capacity calibration at the network-wide level [10].

The capacity calibration problem is very challenging, not only because of the large number of link capacities to be calibrated, but also due to the complex relation between capacities and observed traffic flows. There are many acceptable capacity values which will result in identical traffic flow output. The capacity of a given link cannot be smaller than any observed flow on this link; however, the highest observed traffic flows are likely not the real capacities of the links. One example is a link acting as a bottleneck that will equivalently reduce the flows on its downstream links within a certain time period, which means we are unable to get useful information to calibrate the capacities of those downstream links.

In this paper, we aim to find an efficient capacity calibration method which can be applied to large-scale urban transportation networks. We investigate how network-wide link capacities influence the resulting flows by using a small toy network, present a novel calibration method, and evaluate the performance of the results from this method on a large-scale network model of Stockholm, Sweden.

The proposed method, based on partial least square (PLS) regression, is evaluated for the case where MATSim, an open-source framework for implementing large-scale agent-based transport simulations, is used for assigning the travel demand to the network. The evaluation is made both with respect to the quality of the resulting capacity estimates and the computational efficiency. It is evaluated by comparing the results from the proposed method with those from the method of simultaneous perturbation stochastic approximation (SPSA).

The main contribution of this paper is a novel, network-wide, calibration method for urban network link capacities. Further, the relationship between flows and capacities in the context of urban network capacity calibration is analyzed, which could be helpful for future network-wide approaches for capacity calibration.

In the next section, we go through previous work, put forward relevant earlier published calibration methods and introduce PLS regression. Section 3 presents the method developed in this paper, which contains the following major steps: trial points generation, simulation, PLS regression and capacity updates. Section 4 describes the simulation setup for evaluation of the method and the results are presented and analyzed in Section 5. The last section concludes the paper and suggests directions for further research.

## 2. Literature Review

In the field of traffic model calibration, most of the research focuses on demand (OD) calibration or a combination of demand and supply calibration (i.e., joint calibration).

The joint calibration of DTA models can be categorized into two groups: (1) iterative demand–supply calibration approaches and (2) simultaneous demand–supply calibration approaches [11]. For iterative calibration methods, the O–D flows and route choices are calibrated first, and then the driver behavior parameters are calibrated. These two steps are iterated until a convergence criterion is satisfied [12–14]. For simultaneous demand–supply calibration, Balakrishna [9] proposes a formulation of an optimization problem which can jointly estimate both demand (OD flow and route choice) and supply (speed–density diagram and segment capacity) parameters. Even though capacities are included in this work, they only appear in the constraint part in formulating optimization, acting as upper bounds for flows.

In terms of pure supply calibration, most of the work concentrates on speed–density relationship calibration for individual links, or individual link types [5–7]. The available literature on network-level capacity calibration is very limited. In Lin et al. [10]; a Dantzig–Wolfe decomposition-based heuristic for capacity calibration is presented and it has also been stressed that there are many acceptable capacity values which will result in identical traffic flow output, which makes capacity calibration very challenging.

There is a lot of work concentrating on improving the efficiency and accuracy of methods applied to OD calibration problems. These problems are relevant for capacity calibration since these two types of calibration problems share many similarities.

A frequently encountered calibration method includes numerical estimation of a full Jacobian matrix [15], aiming at finding the local linear approximation between the input and output variables. The issue is that high dimensionality (e.g., number of links) in networks introduces great complexity. This classical calibration method, consisting of series of iterations in which estimation of local Jacobian matrix is computed, often faces problems with computational efficiency.

One alternative to estimating the full Jacobian matrix is the SPSA method, a method that has shown improvement in computational efficiency. It simplifies multivariate optimization problems by approximating the gradient with only a small number of measurements per iteration in which all variables vary randomly [16]. Although the SPSA method is computationally efficient, the performance regarding convergence rate and calibration accuracy deteriorates greatly when the problem dimension increases. According to previous research, the calibration errors stop decreasing at relatively high values [17]. In Lu et al. [18]; different values of algorithm parameters are tested, and adaptive step sizes are evaluated, but with limited effect in decreasing the calibration error. Some researchers investigate modifications and variations of the SPSA method to achieve higher efficiency and better robustness, such as weighted SPSA [17, 19, 20] and cluster-wise SPSA [21], which have better calibration performance compared to the traditional SPSA method.

In Zhang et al. [22] and Chong and Osorio [23]; a simulation-based optimization algorithm is presented which provides a fundamental structure of the calibration

method in this paper. In this method, a simulator and a sampling strategy for collecting trial points are used to update a regression-based model constructing an approximation between inputs and outputs. This regression model is further utilized in an optimization problem minimizing the difference between observed and predicted flows [22]. There has been other work focuses on improving the efficiency of sensitivity analysis based on this method. In Osorio and Bierlaire [24]; a simulation-based optimization (SO) method which improves the efficiency of complex stochastic urban traffic simulators is presented: Trial points from previous iterations are saved and a criterion is formulated to determine the acceptance of these points in a later iteration. The weight for each trail point is formed on the inverse distance weight function [25]. This trial points selection strategy is similar to what we will use in this research.

The method proposed in this paper implements a dimensionality reduction method (PLS regression) in capacity calibration problems to reduce the computational cost. PLS regression has been broadly used in chemometrics, for example, Godoy et al. [26] and Geladi and Kowalski [27] but has to the best of our knowledge not yet been used in the area of traffic model calibration.

Compared to principal components regression (PCR), which has been used in traffic calibration problems, PLS regression has the following advantages:

1. Since loading vectors (which can approximately be regarded as principal components) are considered in a format of pairs, only the diagonal elements in the regression matrix need to be calculated. This will be illustrated in a more detailed way in the latter chapters.
2. In order to achieve the same level of approximation, fewer components are needed in PLS regression than in PCR [28].
3. PLS regression can be the optimal compromise between ordinary least squares (OLS) and PCR for cases with noise existence [29].

### 3. Capacity Calibration

The hypothesis in this paper is that the network structure in combination with multiple link flow observations can be used to improve link capacity estimates. From the literature review, the most popular methods for capacity estimation are based on calibration of fundamental diagrams for individual links. However, for large urban networks, it can be challenging to collect enough data to support local calibration for all links. Furthermore, due to network bottlenecks, many of the links are always in an undersaturated state, which makes it hard to estimate the link capacity locally.

**3.1. Model Formulation.** In this paper, we consider a mathematical model in which  $m$  input variables (capacities),  $x_1, \dots, x_m$ , and  $m$  output variables (flows),  $y_1, \dots, y_m$ , are considered. Here,  $m$  denotes the total number of links in the network. The input vector and output vector are defined as

$$\mathbf{x} = (x_1 x_2 \dots x_m)^T, \quad (1)$$

$$\mathbf{y} = (y_1 y_2 \dots y_m)^T, \quad (2)$$

where  $T$  represents transpose.

In MATSim, which is the simulator used for loading the demand to the network, a network file contains information on attributes of all links in the network, among which the capacity parameter is one of the attributes. The capacity value of a given link determines the maximum number of vehicles which can leave this link per unit of time (1 h). In other words, the capacity we are investigating is the outflow capacity.

It should be stressed that in a real situation, the intersection capacity plays a major role for congestion. However, in MATSim, intersection capacity does not exist as a parameter. Instead, an intersection is a node connecting the upstream and downstream links, deciding the vehicle sequence from different competing upstream links to a certain downstream link based on internal randomness embedded in the MATSim simulator [30]. This formation of vehicle sequence leaving a link is the main noise (randomness) existing in the MATSim simulator. When it comes to the intersection capacity, it is absorbed in the outflow capacity which we are investigating. In short, when we consider the outflow capacity in MATSim, we are exactly investigating the essential intersection capacity in reality.

In the experimental scenarios, we investigate the flows of the first rush hour (from 6 to 7 a.m.) of a specific day, before which the demand is set to zero in the simulator. Within this specific hour, the simulator loads the demand, letting all agents join the network as soon as possible, and each of the agents will travel along its chosen route through the network. A certain number of agents will travel through a given link  $i$  in this one-hour period and this is the link flow output.

The flow of a given link is theoretically determined by capacities of all links, which implies the following relationship in the simulator:

$$y_i = \tilde{f}_i(\mathbf{x}), \quad \forall i = 1, \dots, m, \quad (3)$$

where  $\tilde{f}_i$  represents the capacity-to-flow mapping for link  $i$ . In the investigated scenarios, travel demand and route choice are predetermined and kept fixed. This is an over-ideal assumption, but it can be expected that future large scale mobility data enabling route choice observations (e.g., from GPS probe data) and OD observations (e.g., from mobile network data) can potentially be used to make good estimates of demand and route choice, which makes our assumptions acceptable.

Since we do not know the explicit form of  $\tilde{f}_i$ , the calibration problem is a black-box inverse problem. We need to find an analytical (in our case, linear) approximation of  $\tilde{f}_i(\mathbf{x})$ , which can be denoted as  $\hat{f}_i(\mathbf{x})$ , at the current estimate. Local linear approximation implies that the method requires multiple iterations where the current capacity estimate is updated in each iteration.

There are different objective functions used in calibration problems found in the literature, such as the least square error (LSE) [31], maximum likelihood [32] and entropy maximization and information minimization [33]. One of the most intuitive choices is LSE, in which the mean squared error between observed values of the output (link flow) variables and predicted values of the output variables are minimized:

$$\min_{\mathbf{x}} \sum_{i=1}^{m'} \left( \tilde{y}_i - \tilde{f}_i(\mathbf{x}) \right)^2, \quad (4)$$

where  $\tilde{y}_i$  is the observed flow value on link  $i$  in a specific hour range in a day and  $m'$  is the number of links on which we have flow observations. In most realistic situations, we have  $m' \ll m$ .

In short, the capacity calibration problem is solved through a minimization problem in which the squared Euclidean distance between the observed output vector and the predicted one obtained from the estimated capacities is minimized.

**3.2. Calibration Procedure.** Figure 1 outlines the four main steps of the proposed calibration method: trial points generation, MATSim simulation, PLS regression and capacity update. The method is iterative, with each iteration comprising the four steps illustrated in the figure. Table 1 lists the most important symbols used in this paper and their meaning.

**3.3. Generation and Evaluation of Trial Points.** A network with  $m$  links and a current best capacity estimate vector are given. A trial capacity vector is obtained by uniformly varying each capacity value in a range of  $1 \pm \delta$  multiplied by its current value. Each element in this trial vector is obtained independently. This procedure is repeated until  $n$  trial capacity vectors are generated.

**3.4. MATSim Simulation.** Each capacity vector is evaluated in the road traffic simulator MATSim and the corresponding network flows are obtained.

Omitting an iteration index, the resulting  $n$  (capacity, flow) vector tuples are denoted by  $(\mathbf{x}_r, \mathbf{y}_r)$ , with  $r$  as the replication index within one algorithm iteration. Both capacity and flow vectors are mean-centered as immediate postprocessing [27]. The mean-centering operation prevents the capacity estimate from moving far away from the initial capacities if no new information on the relationship between capacities and flows is found.

**3.5. Joint Linear Approximation and Dimensionality Reduction.** From the previous step, assume a set of mean-centered input/output (capacity/flow) data tuples  $(\mathbf{x}_r, \mathbf{y}_r)$ ,  $r = 1, \dots, n$ , to be given, with both input and output being  $m$ -dimensional vectors. We are interested in estimating a linear model relationship between independent (input) variables  $\mathbf{x}$  and dependent (output) variables  $\mathbf{y}$ . We

further assume that  $n$ , the number of trial points used in PLS regression, is relatively small compared to  $m$ , the dimensionality of the model's in- and output space.

To arrive at an identifiable model, we reduce the dimensionality of both input and output spaces from dimension  $m$  to dimension  $a$ . The low-dimensional representation of the input space is spanned by *loading vectors*  $\mathbf{p}_i, i = 1, \dots, a$ , and every input vector  $\mathbf{x}_r$  is represented as a linear combination of these loading vectors:

$$\mathbf{x}_r = \sum_{i=1}^a t_{ri} \mathbf{p}_i + \mathbf{f}_r, \quad (5)$$

where the *score*  $t_{ri}$  represents the contribution of the  $i$ -th loading vector to  $\mathbf{x}_r$  and  $\mathbf{f}_r$  absorbs the approximation error in input space. Symmetrically, the output space is spanned by loading vectors  $\mathbf{q}_i, i = 1, \dots, a$ :

$$\mathbf{y}_r = \sum_{i=1}^a u_{ri} \mathbf{q}_i + \mathbf{g}_r, \quad (6)$$

with  $u_{ri}$  and  $\mathbf{g}_r$  being specified symmetrically to  $t_{ri}$  and  $\mathbf{f}_r$ . Given the loading vectors, the input and output vectors are hence encoded by the *score vectors*

$$\mathbf{t}_i = (t_{1i} \dots t_{ni})^T, \quad (7)$$

$$\mathbf{u}_i = (u_{1i} \dots u_{ni})^T, \quad (8)$$

with  $i = 1, \dots, a$ . Instead of estimating a regression model coupling  $\mathbf{x}$  and  $\mathbf{y}$ , we estimate one regression model for each  $i = 1, \dots, a$  by OLS. For the  $i$ -th model, its single regression coefficient  $b_i$  is given by

$$b_i = \arg \min_{b \in \mathbb{R}} \|\mathbf{u}_i - b \mathbf{t}_i\|^2. \quad (9)$$

For given loading vectors, this model is used for prediction by (i) representing an input vector  $\mathbf{x}_{n+1}$  in terms of its input scores  $t_{(n+1)i}, i = 1, \dots, a$ , (ii) using the a regression coefficients of (9) to compute the corresponding output scores  $u_{(n+1)i} = b_i t_{(n+1)i}, i = 1, \dots, a$ , and (iii) approximating the output signal from (6) with a zero residual vector  $\mathbf{g}$ .

The PLS algorithm estimates simultaneously the loading vectors  $\mathbf{p}_i, \mathbf{q}_i$ , the corresponding score representation of a set of data tuples  $(\mathbf{x}_r, \mathbf{y}_r), r = 1, \dots, n$ , and the low-dimensional regression models (9). To simplify notation, in- and output vectors are stacked in the following matrices:

$$\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_n)^T, \quad (10)$$

$$\mathbf{Y} = (\mathbf{y}_1 \dots \mathbf{y}_n)^T. \quad (11)$$

Based on this, the PLS regression algorithm from Geladi and Kowalski [27], can be given in Algorithm 1, see also Wei et al., [34].

The PLS regression can be seen as a method that reduces the problem dimensionality by enforcing an OLS solution that is located in a low-dimensional subspace that is constructed along directions of large variability in the explanatory variables [35].

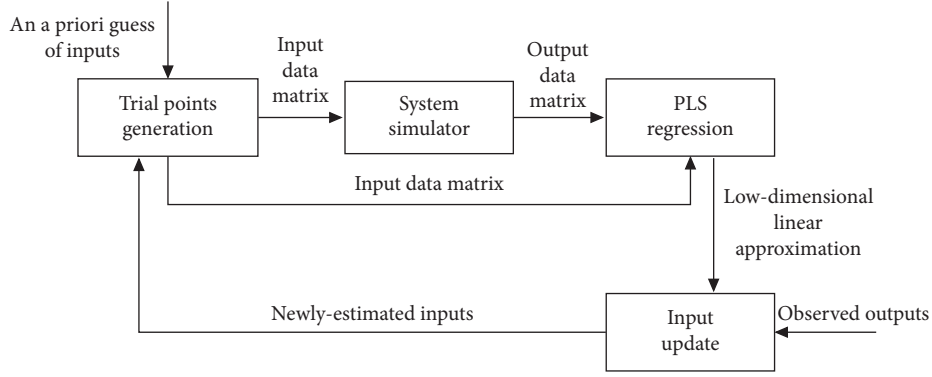


FIGURE 1: Flowchart of the proposed method.

TABLE 1: Main symbols used in the proposed calibration method.

Symbol	Meaning
$m$	Dimension of input variables (capacities) and output variables (flows), i.e., number of links
$m'$	Number of links for which we have synthetic flow observations
$m''$	Number of links which have a non-zero true flow value
$n$	Number of trial points used in the PLS regression
$\delta$	Variation coefficient in the step <i>Generation and evaluation of trial points</i>
$\mathbf{p}_i$	$i$ -th loading vector in input space, size: $m \times 1$
$\mathbf{t}_i$	$i$ -th score vector in input space, size: $n \times 1$
$\mathbf{q}_i$	$i$ -th loading vector in output space, size: $m \times 1$
$\mathbf{u}_i$	$i$ -th score vector in output space, size: $n \times 1$
$a$	The number of loading vector pairs used, i.e., dimension after dimensionality reduction operation
$b_i$	The regression coefficient for $i$ -th loading vector pair (of $\mathbf{u}_i$ on $\mathbf{t}_i$ ) in PLS regression
$\mathbf{s}$	Low dimensional representation of the estimated capacity vector, size: $a \times 1$

Notation: “ $\leftarrow$ ” means a variable assignment from right to left

1. Initialize:
  - a.  $\mathbf{X}_1 \leftarrow \mathbf{X}$ ,
  - b.  $\mathbf{Y}_1 \leftarrow \mathbf{Y}$ .
2. For  $i = 1, \dots, a$ :
  - a. Set  $\mathbf{u}_i$  to an arbitrary column of  $\mathbf{Y}_i$ .
  - b. “X block”:
    - i.  $\mathbf{w}_i \leftarrow \mathbf{X}_i \mathbf{u}_i / \|\mathbf{u}_i\|^2$ ,
    - ii.  $\mathbf{w}_i \leftarrow \mathbf{w}_i / \|\mathbf{w}_i\|$ ,
    - iii.  $\mathbf{t}_i \leftarrow \mathbf{X}_i \mathbf{w}_i$ .
  - c. “Y block”:
    - i.  $\mathbf{q}_i \leftarrow \mathbf{Y}_i \mathbf{t}_i / \|\mathbf{t}_i\|^2$ ,
    - ii.  $\mathbf{q}_i \leftarrow \mathbf{q}_i / \|\mathbf{q}_i\|$ ,
    - iii.  $\mathbf{u}_i \leftarrow \mathbf{Y}_i \mathbf{q}_i$ .
  - d. Update of loadings and scores:
    - i.  $\mathbf{p}_i \leftarrow \mathbf{X}_i \mathbf{t}_i / \|\mathbf{t}_i\|^2$ ,
    - ii.  $\mathbf{t}_i \leftarrow \mathbf{t}_i / \|\mathbf{p}_i\|$ ,
    - iii.  $\mathbf{p}_i \leftarrow \mathbf{p}_i / \|\mathbf{p}_i\|$ .
  - e. Regression:  $b_i = \mathbf{u}_i^T \mathbf{t}_i / \|\mathbf{t}_i\|^2$ .
  - f. Calculation of residuals:
    - i.  $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$ ,
    - ii.  $\mathbf{Y}_{i+1} = \mathbf{Y}_i - b_i \mathbf{t}_i \mathbf{q}_i^T$ .

ALGORITHM 1: PLS regression.

Since the original  $m$  input variables are reduced to  $a$  new variables through PLS regression, the method does not require all the  $m$  output variables to have an observed value to get a unique solution in the optimization problem.

**3.6. Capacity Update.** Denote the mean-centered true network flows as  $\tilde{y}_c = (\tilde{y}_j)$ . The updated mean-centered capacities  $\hat{x}_c$  are then obtained in two steps. First, the following optimization problem is solved:

$$\min_{s=(s_i)} \sum_{j=1}^{m'} \left( \tilde{y}_j - \sum_{i=1}^a b_i s_i q_{ij} \right)^2, \quad (12)$$

where  $q_{ij}$  is the  $j$ -th element in the loading vector  $q_i$ . The solution  $s = (s_1 s_2 \dots s_a)^T$  of this optimization problem contains the scores for all the loading vectors  $p_i, i = 1, \dots, a$ , and  $s$  has a dimension of  $a$ .

Next, the corresponding mean-centered capacity vector is constructed according to

$$\hat{x}_c = \sum_{i=1}^a s_i p_i. \quad (13)$$

Then, the estimate  $\hat{x}$ , without mean-centering, can be recovered from  $\hat{x}_c$  based on the average value for each input variable from trial point data. To avoid oscillations, the currently best capacity estimate is updated by computing a convex combination of the previous estimate and  $\hat{x}$ , with the weight on  $\hat{x}$  being specified further below.

It should be emphasized that in Section 3.1, *Model formulation*, and optimization expression, we use variables  $m$  and  $m'$  for flow dimension, respectively. The reason is when we deal with dimensionality reduction, we consider all the link flows, which could be obtained through MATSim in simulation. While in the *Capacity update*, we assume a scenario that we only have a limited number of links on which we know the flow, which is more realistic. Since this paper aims to provide a very first idea on capacity calibration, we make  $m' = m$  in most numerical experiments.

**3.7. Weight Settings.** In step Capacity update, a weight  $\alpha_k$  (where  $k$  refers to the current iteration number) on  $\hat{x}$  needs to be set to guarantee that the optimal solution can be reached, and oscillations are avoided. Moreover, when trial points are generated, there also exists a weight  $\beta_k$ , which makes the range coefficient of variation  $\delta$  change after each iteration (i.e.,  $\delta = \delta_0 \beta_k$ , where  $\delta_0$  is the fixed initial variation coefficient).  $\alpha_k$  and  $\beta_k$  need to satisfy a series of conditions [16]. These conditions guarantee that the optimal can be reached (or be sufficiently close to) after a certain number of iterations.

$$\alpha_k > 0, \beta_k > 0, \quad \forall k, \quad (14)$$

$$\alpha_k \longrightarrow 0, \beta_k \longrightarrow 0, \quad \text{as } k \longrightarrow \infty, \quad (15)$$

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad (16)$$

$$\sum_{k=1}^{\infty} \left( \frac{\alpha_k}{\beta_k} \right)^2 < \infty. \quad (17)$$

Based on these conditions,  $\alpha_k = 1/k$  and  $\beta_k = (1/k)^{(1/3)}$  are set in this work. They remain unchanged in all the following experiments. Eventually, the final estimated capacities  $\hat{x}$  in  $k$ -th iteration is set according to

$$\hat{x} \longleftarrow \hat{x}_p + \frac{1}{k} (\hat{x} - \hat{x}_p), \quad (18)$$

where  $\hat{x}_p$  is the final estimated capacities in the previous iteration and “ $\longleftarrow$ ” means a variable assignment from right to left.

## 4. Simulation Setup

The calibration method developed in Section 3 is evaluated in a MATSim simulation environment [36] for a toy network with two links and a large-scale urban network for Stockholm, Sweden. (Another 5-link toy model which illustrates loading vectors can be found in Wei [37]). For each network, demand and route choice are fixed. The route choice is generated from the simulation with the initial capacity guess. MATSim was chosen since we believe it provides a reasonable compromise between modelling complexity, flexibility, and computational efficiency for large networks.

Before starting a calibration method run, a synthetic true capacity vector is given. For the toy network, the synthetic true capacities are set manually to help us understand the mechanism of capacities in influencing network flows better. For the Stockholm network, a synthetic true capacity vector is generated by randomly varying each given link capacity in the range [0.85, 1.15] and then multiplying it by the initial guess of this link capacity in MATSim. For example, if a given link has an initial guess of capacity value of 1000, then its synthetic true capacity is generated between 850 and 1150, with a uniform distribution probability. It should be emphasized that the synthetic true capacity vector will not be used in the calibration method. It is only used in the evaluation of the performance of the method after calibration is done.

The MATSim network assignment package is used to assign the flows to routes and to compute network link flows. The simulation runs are made with a demand for the 6-7 a.m. period on a specific day.

For both networks, the method is run for 20 iterations (see Figure 1), this number is selected since the calibration result becomes stable after 20 iterations in most experiments, which can be regarded as a compromise between efficiency and accuracy. In step generation and evaluation of trial points, the initial variation parameter  $\delta_0 = 0.1$ .

In addition, in step *Joint linear approximation and dimensionality reduction*, a sampling strategy is implemented intending for further improving the efficiency of the proposed calibration method: One does not only create a new set of trial points in every iteration, but also recycles all trial points from earlier iterations since introducing new trial points requires rerunning of the simulator to get the

corresponding link flows, which significantly increases the overall computational time. In the first iteration,  $n = 101$  trial points are generated in the simulator and used in the PLS regression. From the second iteration, 11 trial points are generated, and these newly generated trial points are added to the complete pool of trial points from all the previous iterations. Correspondingly, in the PLS regression from 2-nd iteration, the 101 trial points (out of 112) that are closest to the current capacity estimate are used. These values are picked after multiple trials, and it leads to satisfying efficiency and accuracy (which is shown in the result section). In terms of the initial experiments, the loading vector pair number  $a$  is set to 2 and 20 for the toy network and the Stockholm network, respectively. It is also assumed that the link flows are measured for all links, that is,  $m' = m$ , for the initial experiments.

**4.1. Toy Network.** The topology of the investigated toy network is shown in Figure 2. Each link has the same length of 1 km.

In this network, the true capacities of the two links are manually set to be  $\bar{x} = (1000, 500)$  and the initial capacity guess is  $x^{(0)} = (800, 800)$ . The purpose is to illustrate the results of the method for a small network with one bottleneck link (link 2) and a nonbottleneck link (link 1). The travel demand is 758 at 6 a.m., from the origin (O) to the destination (D) shown in Figure 2.

**4.2. Stockholm Network.** The Stockholm network has 22 547 links, with the topological network graph visualized in Figure 3.

In the Stockholm network, initial capacity values and travel demand are taken from a model developed by the Swedish Transport Administration. The route choice file is obtained by running the simulator with initial capacity values and the route choice remains unchanged during the calibration process. This range parameter is different and picked independently for each link.

**4.3. Performance Metrics.** The following metrics are used for evaluation of the calibration methods:

1. The performance of different configurations of the proposed method can be evaluated in terms of the squared error between true (to the calibration method unknown) capacities  $\bar{x}$  and their estimated counterpart  $\hat{x}$ :

$$e_{\text{capacity}} = \frac{1}{m} \|\bar{x} - \hat{x}\|^2, \quad (19)$$

where  $\|\cdot\|$  represents the Euclidean norm.

2. To map the estimated capacities to link flows through the simulator and then compute the difference between these predicted link flows and the true ones:

$$e_{\text{flow}} = \frac{1}{m'} \|\bar{y} - \tilde{f}(\hat{x})\|^2. \quad (20)$$

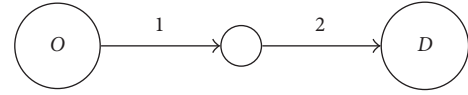


FIGURE 2: Topology of the 2-link toy network.



FIGURE 3: Topology of Stockholm network.

Moreover, errors based on coefficient of determination ( $R^2$ ) is defined as

$$R_{\text{capacity}}^2 = 1 - \frac{\|\bar{x} - \hat{x}\|^2}{\sum_{i=1}^m (\tilde{x}_i - \bar{x})^2}, \quad (21)$$

$$R_{\text{flow}}^2 = 1 - \frac{\|\bar{y} - \tilde{f}(\hat{x})\|^2}{\sum_{i=1}^{m'} (\tilde{y}_i - \bar{y})^2}, \quad (22)$$

where  $\bar{x}$  and  $\bar{y}$  denote the mean values of synthetic true capacities and flows, respectively:

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m \tilde{x}_i, \quad (23)$$

$$\bar{y} = \frac{1}{m'} \sum_{i=1}^{m'} \tilde{y}_i. \quad (24)$$

Equations (19) and (20) are based on mean squared error (MSE). Error metrics derived from mean absolute percentage error (MAPE) are defined as:

$$e_{\text{capacity}}^{\text{MAPE}} = \frac{1}{m} \sum_{i=1}^m \left| \frac{\tilde{x}_i - \hat{x}_i}{\tilde{x}_i} \right|, \quad (25)$$

$$e_{\text{flow}}^{\text{MAPE}} = \frac{1}{m''} \sum_{i=1}^{m''} \left| \frac{\tilde{y}_i - \tilde{f}_i(\hat{\mathbf{x}})}{\tilde{y}_i} \right|, \quad \forall \tilde{y}_i \neq 0, \quad (26)$$

where  $m''$  is the total number of links which have a non-zero synthetic true link flow.

## 5. Results

**5.1. Results for Toy Network.** For the toy network, we investigate flows and capacities on link 1 and link 2. Again, the synthetic true capacity values are  $(\tilde{x}_1, \tilde{x}_2) = (1000, 500)$ , for which MATSim produces corresponding synthetic observed flow  $(\tilde{y}_1, \tilde{y}_2)$ . The flow error function is here defined as

$$f_{\text{toy}}(x_1, x_2) = \frac{1}{2} \left( \tilde{y}_1 - \tilde{f}_1(x_1, x_2) \right)^2 + \frac{1}{2} \left( \tilde{y}_2 - \tilde{f}_2(x_1, x_2) \right)^2, \quad (27)$$

which has the same form as equation (20). Figure 4 shows the contour plot of  $f_{\text{toy}}$  values with respect to different  $x_1$  and  $x_2$  values. Further, the figure shows the iterative path of estimation of capacities  $(\hat{x}_1, \hat{x}_2)$  through 20 iterations.

From the contour graph, we can see that we have a region of low  $f_{\text{toy}}$  values in a rectangular area  $650 < x_1 < 1500, 470 < x_2 < 550$ . It indicates that in order to get a small  $f_{\text{toy}}$  value,  $x_2$  is limited to a small range, while the range for  $x_1$  is significantly larger, caused by link 2 being the bottleneck link. The iterative estimation of  $(\hat{x}_1, \hat{x}_2)$ , starting from  $(800, 800)$ , is illustrated as the red curve in the contour graph. The bottleneck link (link 2) achieves a better calibration result than the upstream link (link 1). The final estimated capacities are  $(\hat{x}_1, \hat{x}_2) = (736, 474)$ , which can be compared with the true capacities  $(\tilde{x}_1, \tilde{x}_2) = (1000, 500)$ . The worse calibration result for  $\hat{x}_1$  is caused by the objective function being flat in the  $x_1$  dimension for  $x_2$  values around the bottleneck capacity. This is an inherent characteristic of the network-wide capacity calibration problem.

It should be noted that the method is applied when the demand starts to be loaded into the network. Under this circumstance, the newly joined agents can travel through each link until the number of agents leaving the link reaches its capacity. In other words, the flow of each link is only determined by its own capacity, not the capacity of other links, and the travel demand on this link under this phase. These link flows provide information on where the bottlenecks are located. However, if the bottleneck is saturated, other links are experiencing either congestion or low flow due to bottlenecks. The flows of these links are bounded above by capacity of the bottleneck, and it is not possible to deduce which links are bottleneck links.

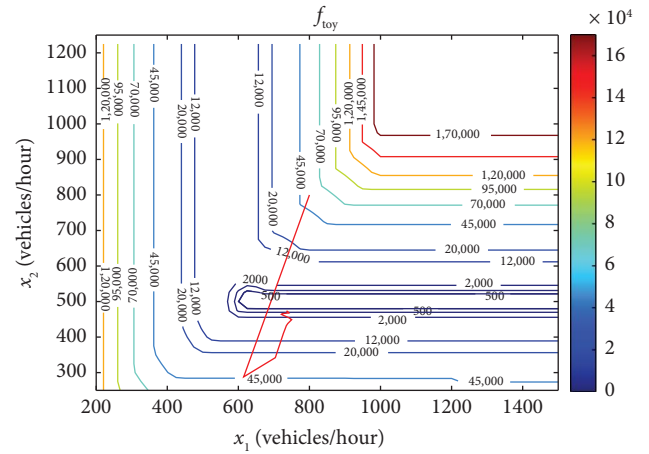


FIGURE 4: Flow error function  $f_{\text{toy}}$  contour plot for different capacities of each link, with estimation of capacities through iterations, for the toy network.

## 5.2. Results for Stockholm network

**5.2.1. Experiment 1—Default Parameter Settings.** Based on the method and experimental scenario introduced before, we construct Experiment 1 with the parameter settings given in Section 4 for the Stockholm network.

Figure 5 shows the MSE capacity error  $e_{\text{capacity}}$  and MSE flow error  $e_{\text{flow}}$  versus iteration numbers, where the error at iteration number 0 represents the error between initial guess and true values of capacities and flows, respectively. From Figure 5, both errors decrease as the iteration number increases. The fact that neither  $e_{\text{capacity}}$  nor  $e_{\text{flow}}$  goes to 0 is due to that for capacities, there exists many-to-one functional relationship when they are mapped onto flows, and for the flows, randomness exists in modeling of intersections.

Figures 6 and 7 show the initial capacities and estimated capacities for all links in Experiment 1, respectively, and they are compared to the synthetic true capacities, (for initial values of capacities, there is a certain number of discrete values for all links, which comes from a very generalized classification of links from the data provider). It is visible that the estimated capacities are closer to the true ones when compared to the distance between initial capacities and true capacities. Similarly, Figures 8 and 9 show the initial flows and predicted flows for all links in Experiment 1, respectively, and they are compared to the observed flows. There is a clear improvement in reducing the difference between simulated flows and observed flows after calibration.

We also test the performance of an SPSA based calibration method [38] with 700 iterations, in which the step size parameters are selected through trial and error. The calibration result through iterations is shown in Figure 10.

Table 2 gives the initial error and the error after calibration for three error metrics presented in Section 4, where *initial* means the error before calibration, *proposed* represents our proposed method in this paper and *SPSA* represents the SPSA method. The results indicate that the

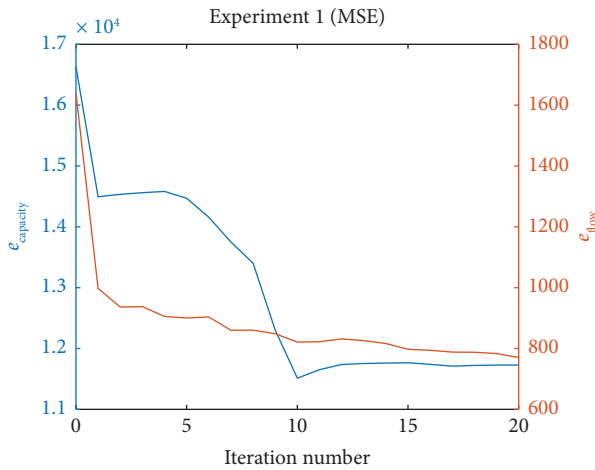


FIGURE 5: Capacity error and flow error versus iteration number in Experiment 1 (MSE).

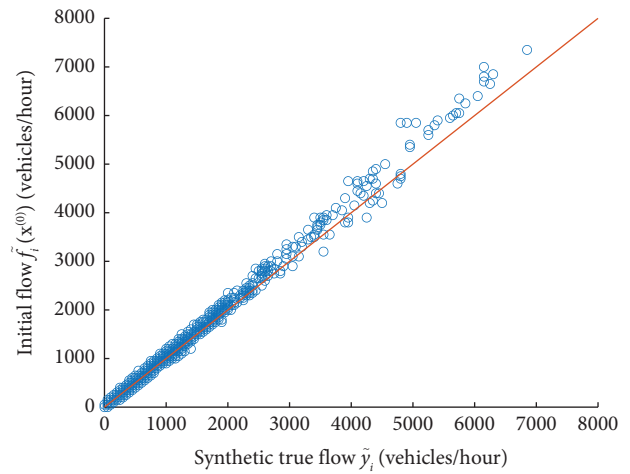


FIGURE 8: Initial flows versus observed flows for all links in Experiment 1 (MSE).

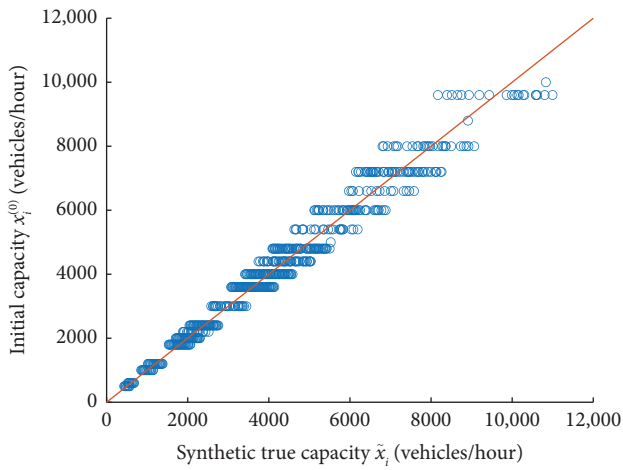


FIGURE 6: Initial capacities versus true capacities for all links in Experiment 1 (MSE).

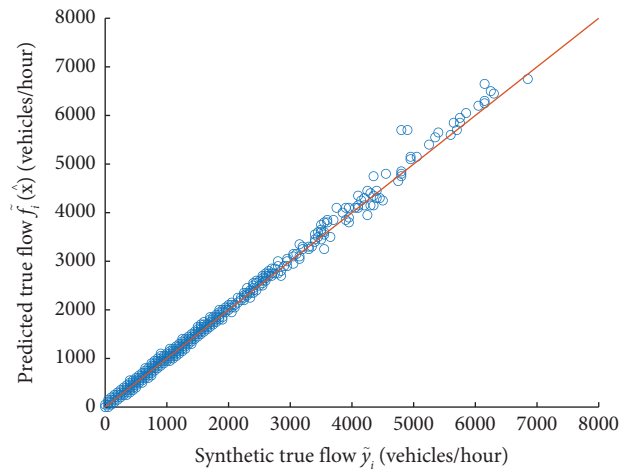


FIGURE 9: Predicted flows versus observed flows for all links in Experiment 1 (MSE).

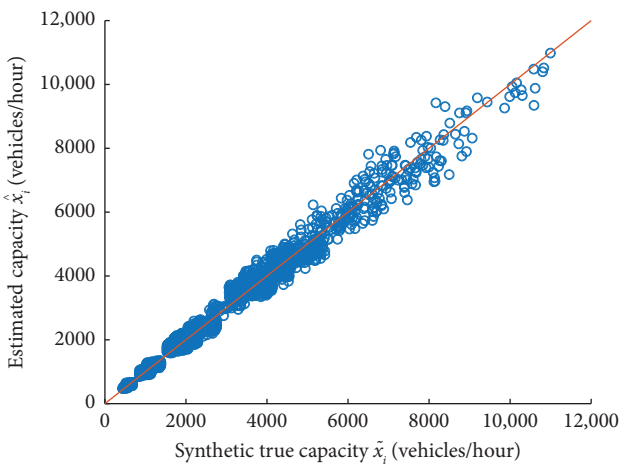


FIGURE 7: Estimated capacities versus true capacities for all links in Experiment 1 (MSE).

proposed method performs better than the SPSA method in terms of calibration accuracy. The  $R^2$  metric value does not change much after calibration since it is very close to 1 before capacities are calibrated. For the result from the SPSA method, even though the flow error decreases, we observe an increase in capacity error for metrics after calibration. Due to the many-to-one mapping from capacities to flows, it is possible that estimated capacities move further away from the true synthetic capacities when compared with initial capacities. It should be stressed that the proposed method requires about 8 h of running time on a PC with a RAM of 16 GB, while the time for running the counterpart SPSA method is roughly twice as long.

5.2.2. *Experiment 2—Influence on Initial Variation Coefficient  $\delta_0$  in Trial Points Generation.* In Experiment 2, we investigate the influence of initial variation coefficient  $\delta_0$  in

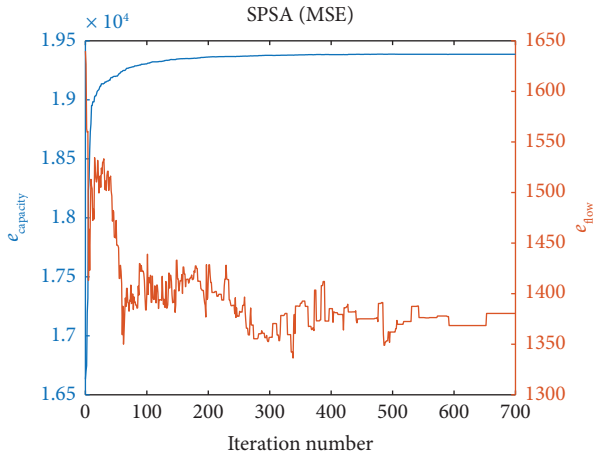


FIGURE 10: Capacity error and flow error versus iteration number for SPSA counterpart (MSE).

TABLE 2: Capacity error and flow error before and after calibration, for different methods and error metrics (Experiment 1).

		$e_{\text{capacity}}$	$e_{\text{flow}}$
MSE	Initial	$1.66 \times 10^4$	1640
	Proposed	$1.17 \times 10^4$	770
	SPSA	$1.93 \times 10^4$	1380
		$R^2_{\text{capacity}}$	$R^2_{\text{flow}}$
$R^2$	Initial	0.9822	0.9901
	Proposed	0.9875	0.9957
	SPSA	0.9793	0.9923
		$e_{\text{capacity}}^{\text{MAPE}}$	$e_{\text{flow}}^{\text{MAPE}}$
MAPE	Initial	0.076	0.045
	Proposed	0.061	0.034
	SPSA	0.098	0.042

trial points generation ( $\delta_0 = 0.05, 0.10, 0.15, 0.20, 0.25$ ). Figures 11 and 12 show the capacity error  $e_{\text{capacity}}$  and flow error  $e_{\text{flow}}$  (both are in MSE metric) versus iteration numbers in Experiment 2. From the graphs, it can be observed that  $\delta_0 = 0.15$  case has the best calibration result. It indicates that the ideal  $\delta_0$  value should be neither too large (approximation will be inaccurate) nor too small (noise influence would be large).

**5.2.3. Experiment 3—Influence on Initial Variation Coefficient  $\delta_0$  in Trial Points Generation.** In Experiment 3, we investigate the influence of the number of loading vector pairs  $a$  in PLS regression ( $a = 20, 30, 40, 50$ ). Figures 13 and 14 show the errors for different numbers of loading vector pairs used in PLS regression. More loading vector pairs being implemented indicate better results, but it should be noted that the introduction of more loading vectors leads to an increase in the running time. In this experiment, the running time for  $a = 50$  is around 15 h.

Experiments 2 and 3 suggest that with the proper combination of  $\delta_0$  and  $a$ , for example,  $\delta_0 = 0.15$  and  $a = 50$ , calibration can achieve even higher accuracy without

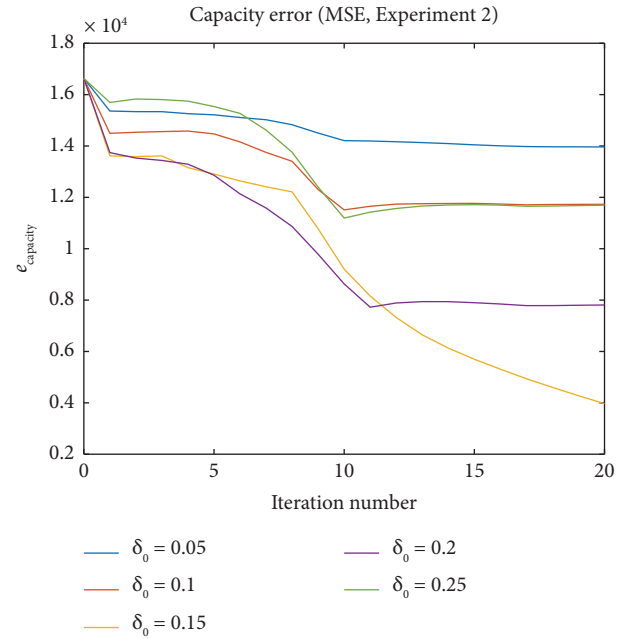


FIGURE 11:  $e_{\text{capacity}}$  versus iteration number in Experiment 2 (different initial variation coefficients).

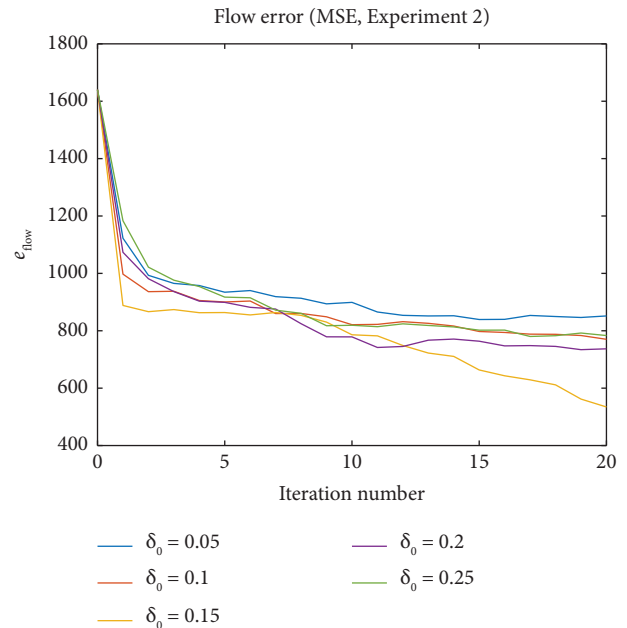


FIGURE 12:  $e_{\text{flow}}$  versus iteration number in Experiment 2 (different initial variation coefficients).

significantly compromising computational efficiency. The result is shown in Figures 15 and 16, with the blue and red curves representing the best results for different  $\delta_0$  and  $a$  values in Experiment 2 and 3, respectively.

**5.2.4. Experiment 4—Nonfully Known Flow Measurement Case.** In real urban networks, it is not likely that sensor-based flow measurements are available on all links in the

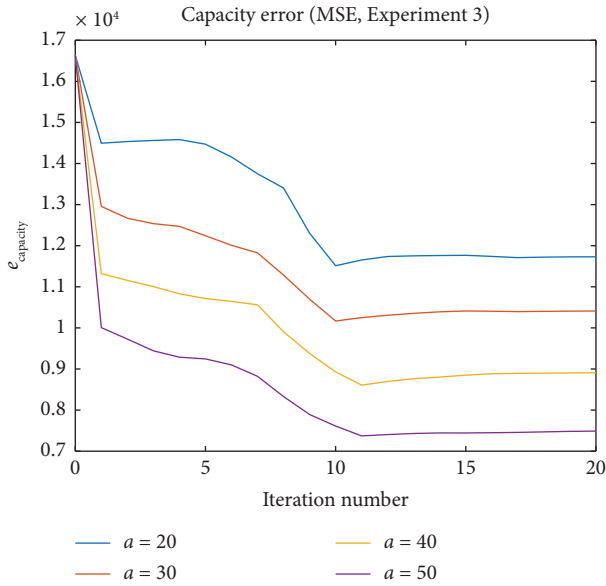


FIGURE 13:  $e_{\text{capacity}}$  versus iteration number in Experiment 3 (different numbers of loading vector pairs).

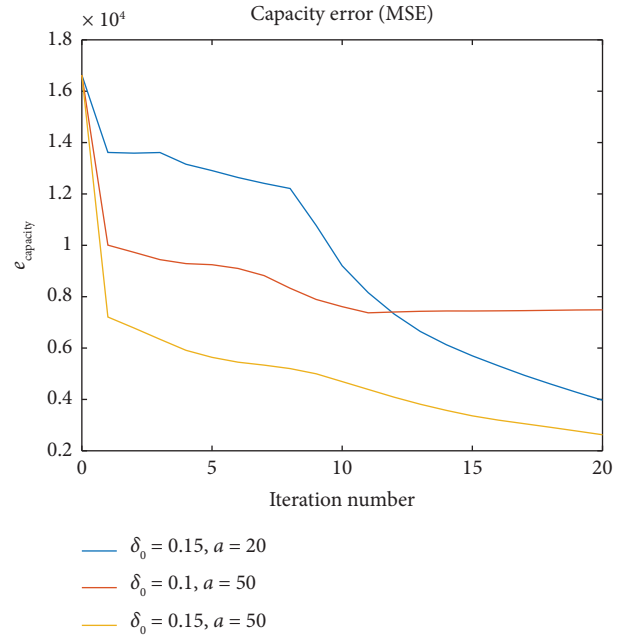


FIGURE 15:  $e_{\text{capacity}}$  versus iteration number for combining best parameter settings from Experiment 2 and 3.

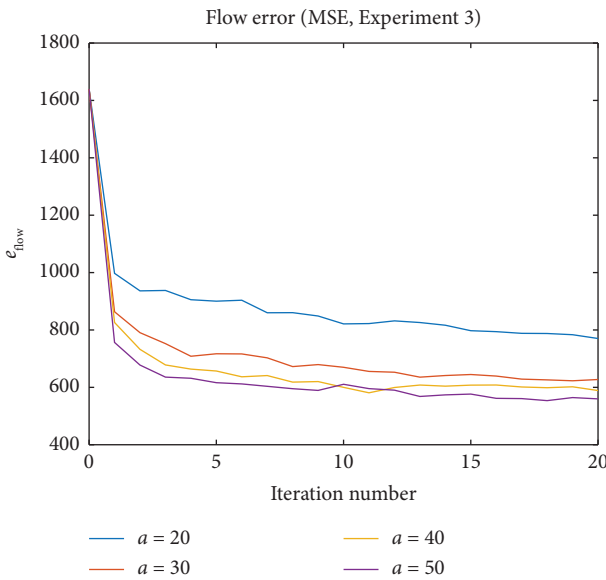


FIGURE 14:  $e_{\text{flow}}$  versus iteration number in Experiment 3 (different numbers of loading vector pairs).

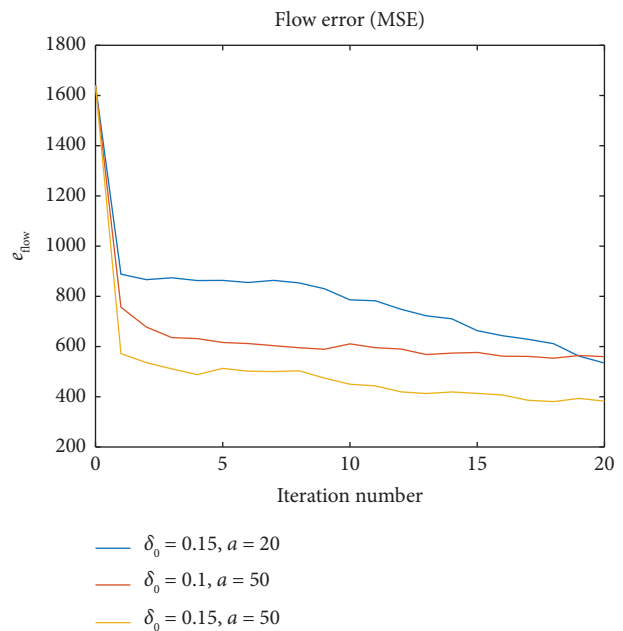


FIGURE 16:  $e_{\text{flow}}$  versus iteration number for combining best parameter settings from Experiment 2 and 3.

investigated network. In Experiment 4, the measured flows are known only for a fixed percentage of all links and these links are picked randomly. We investigate the effect of the proportion of link flow observations in the network ( $m'/m = 100\%, 75\%, 50\%, 25\%, 1\%$ ). It should be stressed again that it only influences the objective function part in this method, and both MATSim and PLS regression still treat the output as  $m$ -dimensional. Figures 17 and 18 show the errors for different proportions.

One can notice that the method performs well in terms of decreasing the flow error when  $m'/m < 100\%$ . For  $m'/m = 25\%$ , which means we only have link flow

observations on 25% of the links, the performance is similar to the case of 100%. The 1% case performs worse compared to the 100% case regarding flow error, but the error still decreases compared to the initial one. In terms of capacity error, the final error is also smaller than the initial one for all percentage values in this Experiment 1. It is evident from this experiment that the method does not seem sensitive to  $m'$ , unless the fraction is very small. One interesting aspect of

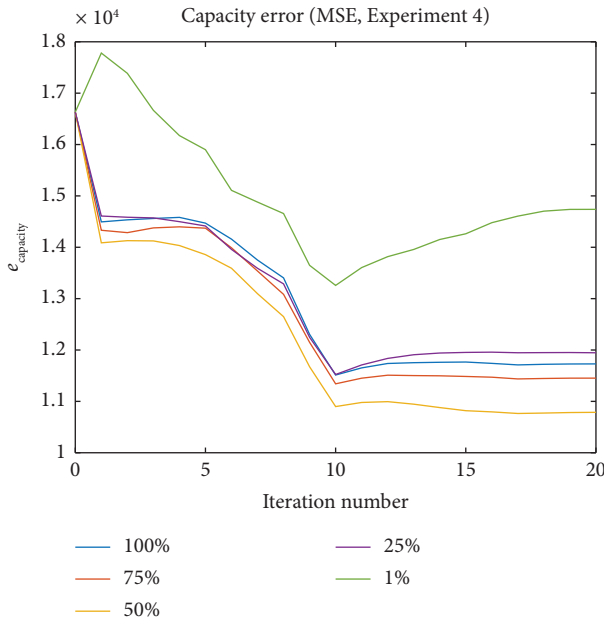


FIGURE 17:  $e_{\text{capacity}}$  versus iteration number in Experiment 4 (different subset sizes of known link flow observations).

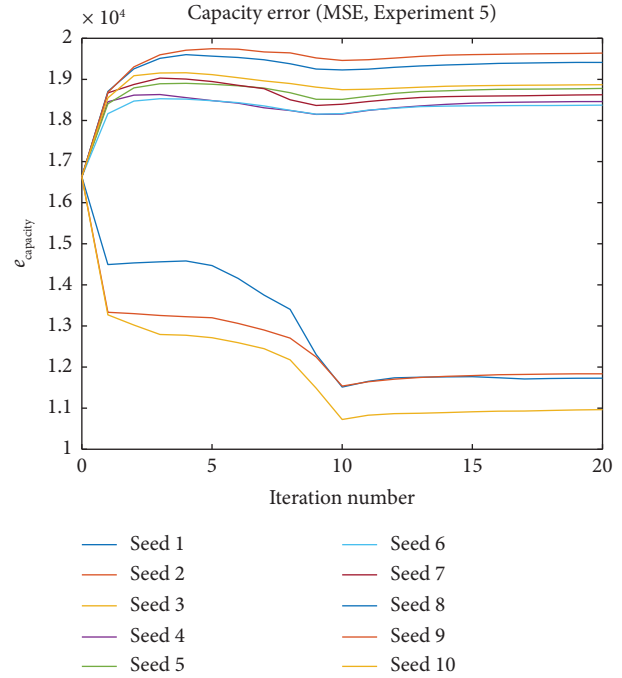


FIGURE 19:  $e_{\text{capacity}}$  versus iteration number in Experiment 5 (different random seeds in generation of trial points).

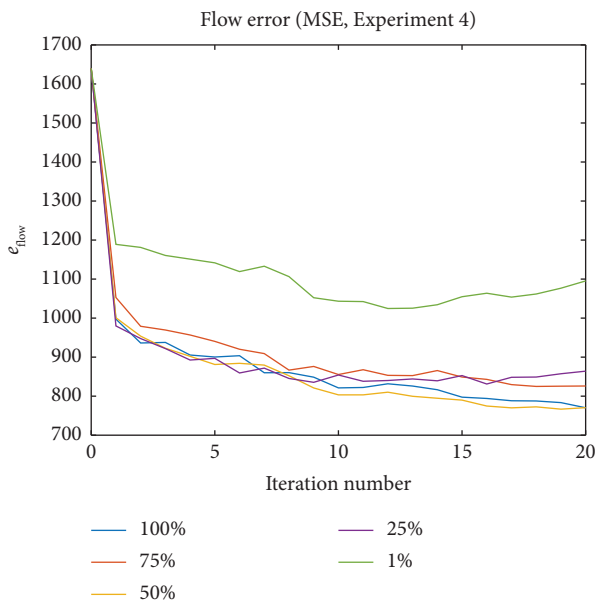


FIGURE 18:  $e_{\text{flow}}$  versus iteration number in Experiment 4 (different subset sizes of known link flow observations).

this experiment is that the 50% case has better calibration result than 75% case, which is worth further investigation.

**5.2.5. Experiment 5—Influence of Different Random Seeds in Generation of Trial Points.** From Section 3, where we illustrate the whole proposed method, the generated trial points will influence the result of calibration. To test reliability of the proposed method, we can do multiple experiments to test the influence of random seeds in trial points generation. The generated trial points will influence

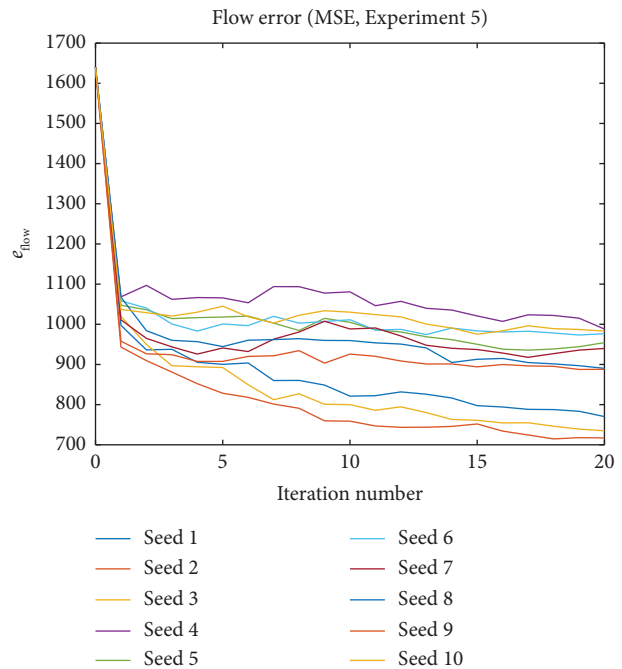


FIGURE 20:  $e_{\text{flow}}$  versus iteration number in Experiment 5 (different random seeds in generation of trial points).

the result of calibration. If different random seed settings are implemented, the calibration result will differ. In Experiment 5, we redo the work in Experiment 1, but with multiple different random seeds for generating trial points (10 seeds in total).

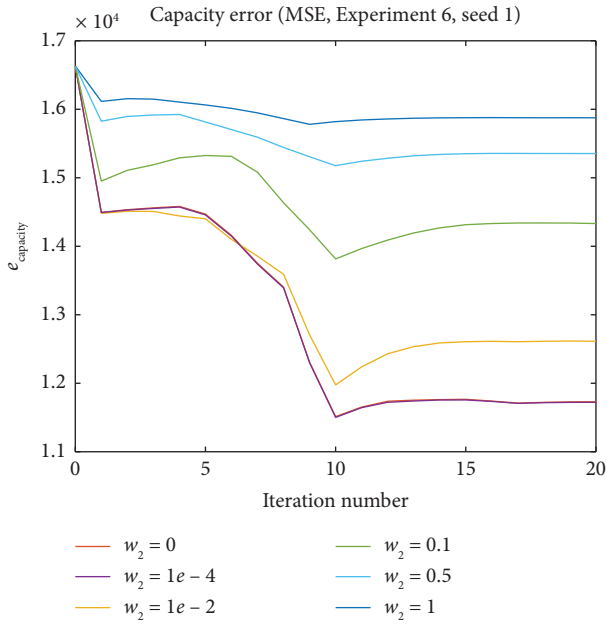


FIGURE 21:  $e_{\text{capacity}}$  versus iteration number in Experiment 6 (different  $w_2$  values, seed 1).

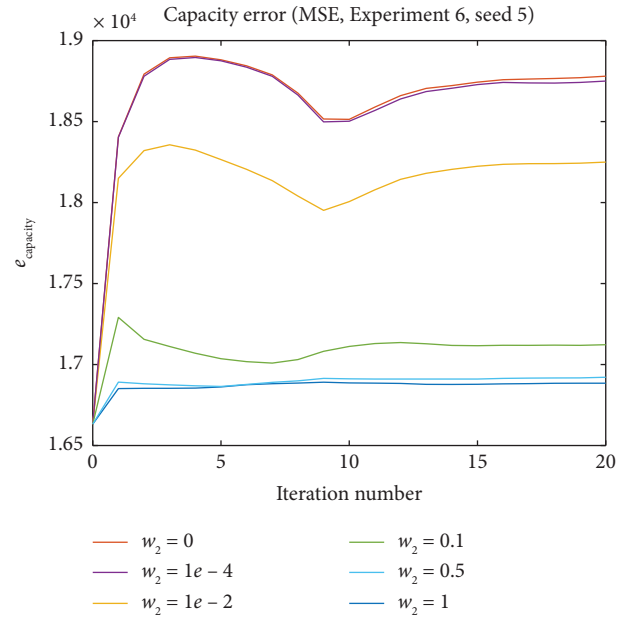


FIGURE 23:  $e_{\text{capacity}}$  versus iteration number in Experiment 6 (different  $w_2$  values, seed 5).

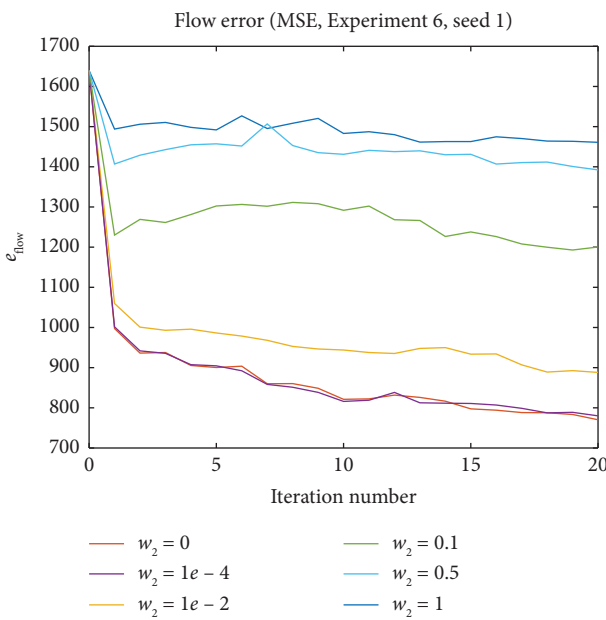


FIGURE 22:  $e_{\text{flow}}$  versus iteration number in Experiment 6 (different  $w_2$  values, seed 1).

Figures 19 and 20 show both capacity and flow errors for different random seeds. The method performs well in terms of decreasing the flow error for all 10 seeds. However, for the capacity error, results for some seeds are even worse when compared to the initial error. It is because in the objective function, we don't constrain the capacities to be close to their

initial values. Moreover, it is known that the map between capacities and flows are many to one, the solver aims at finding capacity values which make the simulated flows similar to the observed ones without really taking care of the estimated capacities values. This phenomenon leads to Experiment 6.

**5.2.6. Experiment 6—Influence of Adding Capacity Closeness Term in Optimization.** In this experiment, we reformulate the expression of the optimization, including the terms making the capacity estimate be close to the initial guess. Note that, for simple reading, the formulation is given in a form without applying dimensionality reduction. In the experiment, it is transformed to a corresponding low-dimensional representation:

$$\min_{\mathbf{x}} \sum_{i=1}^{m'} \left( \tilde{y}_i - \tilde{f}_i(\mathbf{x}) \right)^2 + w_2 \left\| \mathbf{x} - \mathbf{x}^{(0)} \right\|^2, \quad (28)$$

where  $\mathbf{x}^{(0)} = (x_1^{(0)} \dots x_m^{(0)})^T$  represents the initial values of capacities of all links. For the previous five experiments, it is equivalent to this formulation with weight  $w_2 = 0$ . Here, we test the calibration result for different  $w_2$  values as well as for two different seeds (seed 1 and seed 5 from Experiment 5, which represent a 'good' and a 'bad' calibration results in terms of capacity error, respectively). The error graphs are shown in Figures 21, 22, 23, 24.

Intuitively, and as can be seen from the graphs, larger  $w_2$  values give capacity estimations that is closer to the initial capacity, which on the other hand reduces the error

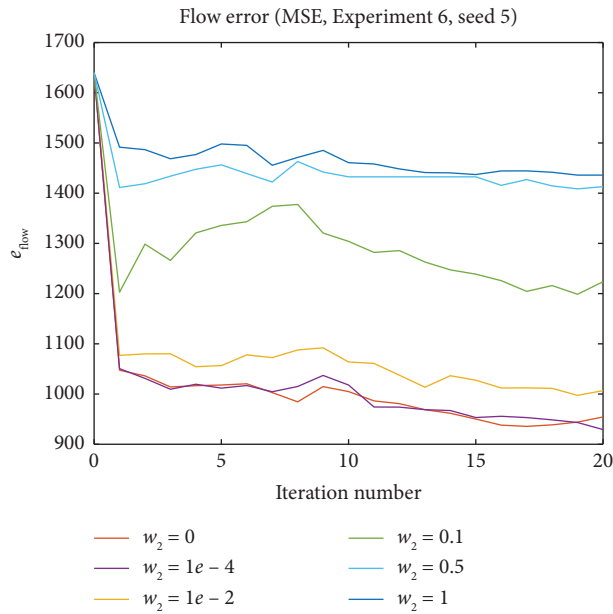


FIGURE 24:  $e_{\text{flow}}$  versus iteration number in Experiment 6 (different  $w_2$  values, seed 5).

reduction in flow. For *seed 1*, this means  $w_2 = 0$  gives the best result for both errors since the existence of capacity error term in optimization will impede the calibration performance. For *seed 5*, higher  $w_2$  values make the capacity error more stable, while on the other hand, the improvement on flow error is smaller.

## 6. Conclusion

Capacity calibration is a very challenging as well as an important problem in traffic modeling and network-wide approaches for capacity calibration are relatively unexplored in the literature. In this paper, we have highlighted the challenges in network-wide calibration.

Furthermore, we propose a novel method for network-wide capacity calibration of DTA models by implementing PLS regression to achieve dimensionality reduction. In our approach, the modified objective function can reduce the effect of these challenges and give a more stable result in the capacity dimension. We evaluate it by using simulation on a toy network as well as a large-scale urban network with promising results for the selected large-scale network in terms of both accuracy and efficiency when compared to the widely used SPSA method.

The assumption related to fixed and known OD demand is a major simplification of the problem and needs to be addressed in future work. However, new large-scale mobility data, like GPS probe data and mobile network data, with direct observations of route choice and OD demand, may support these assumptions in the future and can also be interesting to incorporate in the estimation method as future work.

Due to the theoretical foundation of the proposed method, one can expect it can be further implemented in OD calibration problems, which share a very similar structure.

Future work includes analysis of the method across more networks, incorporating speed data for bottleneck detection, integrating network-wide and local approaches for capacity calibration, and utilizing different simulators as well as real-world data.

## Data Availability Statement

The data that support the findings of this study are available from Swedish Transport Administration. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the authors with the permission of Swedish Transport Administration.

## Disclosure

An earlier draft of this manuscript was presented in 10th Symposium of the European Association for Research in Transportation, hEART 2022, Leuven, Belgium, June 1–3, 2022, and was also presented as the main author's Licentiate thesis in Linköping University, 2022. (Link: <https://liu.diva-portal.org/smash/get/diva2:1689092/FULLTEXT02.pdf>).

## Conflicts of Interest

The authors declare no conflicts of interest.

## Funding

This study was funded by the Trafikverket, 2018/134731 2021/22404.

## Acknowledgments

This work was funded by the Swedish Transport Administration (TRV 2018/134731 and TRV 2021/22404).

Software License Information: MATLAB R2024a: 663068.

## References

- [1] Z. Wei, "Critical Enhancements of a Dynamic Traffic Assignment Model for Highly Congested, Complex Urban Network," (Massachusetts Institute of Technology, 2010), Doctoral dissertation.
- [2] M. E. Ben-Akiva, S. Gao, Z. Wei, and Y. Wen, "A Dynamic Traffic Assignment Model for Highly Congested Urban Networks," *Transportation Research Part C: Emerging Technologies* 24 (2012): 62–82, <https://doi.org/10.1016/j.trc.2012.02.006>.
- [3] S. Shafiei, Z. Gu, and M. Saberi, "Calibration and Validation of a Simulation-Based Dynamic Traffic Assignment Model for a Large-Scale Congested Network," *Simulation Modelling Practice and Theory* 86 (2018): 169–186, <https://doi.org/10.1016/j.simpat.2018.04.006>.
- [4] Transportation Research Board and National Academies of Sciences Engineering and Medicine, in *Highway Capacity Manual 7th Edition: A Guide for Multimodal Mobility Analysis* (The National Academies Press, 2022).
- [5] S. Chiappone, O. Giuffrè, A. Granà, R. Mauro, and A. Sferlazza, "Traffic Simulation Models Calibration Using Speed–Density Relationship: An Automated Procedure Based

- on Genetic Algorithm,” *Expert Systems With Applications* 44 (2016): 147–155, <https://doi.org/10.1016/j.eswa.2015.09.024>.
- [6] G. Dervisoglu, G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya, “Automatic Calibration of the Fundamental Diagram and Empirical Observations on Capacity,” *Transportation Research Board 88th Annual Meeting* 15 (2009): 31–59.
- [7] R. Zhong, C. Chen, A. H. Chow, T. Pan, F. Yuan, and Z. He, “Automatic Calibration of Fundamental Diagram for First-Order Macroscopic Freeway Traffic Models,” *Journal of Advanced Transportation* 50, no. 3 (2016): 363–385, <https://doi.org/10.1002/atr.1334>.
- [8] K. K. Kundé, “Calibration of Mesoscopic Traffic Simulation Models for Dynamic Traffic Assignment,” (Massachusetts Institute of Technology, 2002), Ph.D. thesis.
- [9] R. Balakrishna, “Off-Line Calibration of Dynamic Traffic Assignment Models,” (Massachusetts Institute of Technology, 2006), Ph.D. thesis.
- [10] D.-Y. Lin, V. Valsaraj, and S. T. Waller, “A Dantzig-Wolfe Decomposition-Based Heuristic for Off-Line Capacity Calibration of Dynamic Traffic Assignment,” *Computer-Aided Civil and Infrastructure Engineering* 26, no. 1 (2011): 1–15.
- [11] R. Omrani and L. Kattan, “Demand and Supply Calibration of Dynamic Traffic Assignment Models: Past Efforts and Future Challenges,” *Transportation Research Record: Journal of the Transportation Research Board* 2283, no. 1 (2012): 100–112, <https://doi.org/10.3141/2283-11>.
- [12] R. Balakrishna, H. N. Koutsopoulos, and M. Ben-Akiva, “Calibration and Validation of Dynamic Traffic Assignment Systems,” in *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic Theory* (College Park: University of Maryland, 2005).
- [13] A. Gupta, “Observability of Origin-Destination Matrices for Dynamic Traffic Assignment,” (Massachusetts Institute of Technology, 2005), Ph.D. thesis.
- [14] C. Yu-Sen, H. J. Van Zuylen, and L. Rex, “Developing a Large-Scale Urban Decision Support System,” *IFAC Proceedings Volumes* 39, no. 12 (2006): 216–221, <https://doi.org/10.3182/20060829-3-nl-2908.00038>.
- [15] E. Cascetta, “Estimation of Trip Matrices From Traffic Counts and Survey Data: A Generalized Least Squares Estimator,” *Transportation Research Part B: Methodological* 18, no. 4-5 (1984): 289–299, [https://doi.org/10.1016/0191-2615\(84\)90012-2](https://doi.org/10.1016/0191-2615(84)90012-2).
- [16] J. Spall, “Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation,” *IEEE Transactions on Automatic Control* 37, no. 3 (1992): 332–341, <https://doi.org/10.1109/9.119632>.
- [17] C. Antoniou, C. L. Azevedo, L. Lu, F. Pereira, and M. Ben-Akiva, “W-SPSA in Practice: Approximation of Weight Matrices and Calibration of Traffic Simulation Models,” *Transportation Research Procedia* 7 (2015): 233–253, <https://doi.org/10.1016/j.trpro.2015.06.013>.
- [18] L. Lu, “W-SPSA: An Efficient Stochastic Approximation Algorithm for the Off-Line Calibration of Dynamic Traffic Assignment Models,” (Massachusetts Institute of Technology, 2013), Ph.D. thesis.
- [19] L. Lu, Y. Xu, C. Antoniou, and M. Ben-Akiva, “An Enhanced SPSA Algorithm for the Calibration of Dynamic Traffic Assignment Models,” *Transportation Research Part C: Emerging Technologies* 51 (2015): 149–166, <https://doi.org/10.1016/j.trc.2014.11.006>.
- [20] S. Oh, R. Seshadri, C. L. Azevedo, and M. Ben-Akiva, “Demand Calibration of Multimodal Microscopic Traffic Simulation Using Weighted Discrete SPSA,” *Transportation Research Record: Journal of the Transportation Research Board* 2673, no. 5 (2019): 503–514, <https://doi.org/10.1177/0361198119842107>.
- [21] A. Tympakianaki, H. Koutsopoulos, and E. Jenelius, “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 (2015): 231–245, <https://doi.org/10.1016/j.trc.2015.01.016>.
- [22] C. Zhang, C. Osorio, and G. Flötteröd, “Efficient Calibration Techniques for Large-Scale Traffic Simulators,” *Transportation Research Part B: Methodological* 97 (2017): 214–239, <https://doi.org/10.1016/j.trb.2016.12.005>.
- [23] L. Chong and C. Osorio, “A Simulation-Based Optimization Algorithm for Dynamic Large-Scale Urban Transportation Problems,” *Transportation Science* 52, no. 3 (2018): 637–656, <https://doi.org/10.1287/trsc.2016.0717>.
- [24] C. Osorio and M. Bierlaire, “A Simulation-Based Optimization Framework for Urban Transportation Problems,” *Operations Research* 61, no. 6 (2013): 1333–1345, <https://doi.org/10.1287/opre.2013.1226>.
- [25] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally Weighted Learning. Lazy Learning” (1997), 11–73.
- [26] J. L. Godoy, J. R. Vega, and J. L. Marchetti, “Relationships Between PCA and PLS-Regression,” *Chemometrics and Intelligent Laboratory Systems* 130 (2014): 182–191, <https://doi.org/10.1016/j.chemolab.2013.11.008>.
- [27] P. Geladi and B. R. Kowalski, “Partial Least-Squares Regression: A Tutorial,” *Analytica Chimica Acta* 185 (1986): 1–17, [https://doi.org/10.1016/0003-2670\(86\)80028-9](https://doi.org/10.1016/0003-2670(86)80028-9).
- [28] I. S. Helland, “On the Structure of Partial Least Squares Regression,” *Communications in Statistics—Simulation and Computation* 17, no. 2 (1988): 581–607, <https://doi.org/10.1080/03610918808812681>.
- [29] M. Stone and R. J. Brooks, “Continuum Regression: Cross-Validated Sequentially Constructed Prediction Embracing Ordinary Least Squares, Partial Least Squares and Principal Components Regression,” *Journal of the Royal Statistical Society—Series B: Statistical Methodology* 52, no. 2 (1990): 237–258, <https://doi.org/10.1111/j.2517-6161.1990.tb01786.x>.
- [30] G. Flötteröd, “Queueing Representation of Kinematic Waves,” *The Multi-Agent Transport Simulation MATSim* (Ubiquity Press, 2016), 347–352.
- [31] E. Cascetta, D. Inaudi, and G. Marquis, “Dynamic Estimators of Origin-Destination Matrices Using Traffic Counts,” *Transportation Science* 27, no. 4 (1993): 363–373, <https://doi.org/10.1287/trsc.27.4.363>.
- [32] H. Spiess, “A Maximum Likelihood Model for Estimating Origin-Destination Matrices,” *Transportation Research Part B: Methodological* 21, no. 5 (1987): 395–412, [https://doi.org/10.1016/0191-2615\(87\)90037-3](https://doi.org/10.1016/0191-2615(87)90037-3).
- [33] H. J. Van Zuylen and L. G. Willumsen, “The Most Likely Trip Matrix Estimated From Traffic Counts,” *Transportation Research Part B: Methodological* 14, no. 3 (1980): 281–293, [https://doi.org/10.1016/0191-2615\(80\)90008-9](https://doi.org/10.1016/0191-2615(80)90008-9).
- [34] G. Wei, J. Ekström, and G. Flötteröd, “Calibration of Urban Road Network Capacities,” in *hEART 2022: 10th Symposium of the European Association for Research in Transportation* (2022).

- [35] I. E. Frank and J. H. Friedman, "A Statistical View of Some Chemometrics Regression Tools," *Technometrics* 35, no. 2 (1993): 109–135, <https://doi.org/10.2307/1269656>.
- [36] A. Horni, K. Nagel, and K. W. Axhausen, *The Multi-Agent Transport Simulation MATSim* (Ubiquity Press, 2016).
- [37] G. Wei, "Calibration of Urban Network Capacities," (Linköping University Electronic Press, 2022), Licentiate dissertation.
- [38] R. Balakrishna, M. Ben-Akiva, and H. N. Koutsopoulos, "Offline Calibration of Dynamic Traffic Assignment: Simultaneous Demand-and-Supply Estimation," *Transportation Research Record: Journal of the Transportation Research Board* 2003, no. 1 (2007): 50–58, <https://doi.org/10.3141/2003-07>.