Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 7 (2015) 233 – 253

Transportation Research Procedia

www.elsevier.com/locate/procedia

21st International Symposium on Transportation and Traffic Theory

W–SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models

Constantinos Antoniou [a], Carlos Lima Azevedo [b], Lu Lu [c], Francisco Pereira [b,*], Moshe Ben-Akiva [d]

[a] *National Technical University of Athens, GR-15780, Zografou, Greece*
[b] *Singapore-MIT Alliance for Research and Technology (SMART), Singapore 138602, Singapore*
[c] *Google Inc., Mountain View, CA, 94043, USA*
[d] *Massachusetts Institute of Technology (MIT), Cambridge, MA 02139-4307, USA*

Abstract

The development and calibration of complex traffic models demands parsimonious techniques, because such models often involve hundreds of thousands of unknown parameters. The Weighted Simultaneous Perturbation Stochastic Approximation (W–SPSA) algorithm has been proven more efficient than its predecessor SPSA (Spall, 1998), particularly in situations where the correlation structure of the variables is not homogeneous. This is crucial in traffic simulation models where effectively some variables (e.g. readings from certain sensors) are strongly correlated, both in time and space, with some other variables (e.g. certain OD flows). In situations with reasonably sized traffic networks, the difference is relevant considering computational constraints. However, W–SPSA relies on determining a proper weight matrix (**W**) that represents those correlations, and such a process has been so far an open problem, and only heuristic approaches to obtain it have been considered.

This paper presents W–SPSA in a formally comprehensive way, where effectively SPSA becomes an instance of W–SPSA, and explores alternative approaches for determining the matrix **W**. We demonstrate that, relying on a few simplifications that marginally affect the final solution, we can obtain **W** matrices that considerably outperform SPSA. We analyse the performance of our proposed algorithm in two applications in motorway networks in Singapore and Portugal, using a dynamic traffic assignment model and a microscopic traffic simulator, respectively.

Selection and peer-review under responsibility of Kobe University

Keywords: calibration algorithms, dynamic traffic assignment, microscopic traffic simulation, large–scale applications, optimisation, heuristics

1. Introduction

Due to the well known complexity of transportation systems in our cities, together with their fundamental role in terms of environment, quality of life and economic growth, research in analysis and prediction of traffic phenomena is gaining a growing importance. This has been even more notable with the recent sensing and data processing innovations of varying nature (e.g. telecom, smart cards), globally referred to as ”big data”. We do have more data, more computing power and higher recognition of the importance of understanding traffic in our cities.

* Corresponding author. Tel.: +65-6601-1548 ; fax: +65-6684-2218.
E-mail address: camara@mit.edu

Selection and peer-review under responsibility of Kobe University
doi:10.1016/j.trpro.2015.06.013

However, the problem is still very complex as it quickly reaches high dimensionality with large networks, multiple measurements, several traffic control systems, and high and heterogeneous demand patterns. An approach to deal with this complexity is by using simulation models. In this case, the origin–destination (OD) flows (the demand) are assigned to the system by moving vehicles on the network (the supply). This approach can capture emergent behaviour (e.g. congestion) that is often hard to predict analytically. To run properly, simulation models expect, therefore, both supply and demand inputs and parameters. The size and type of such parameter set depends on the simulation scenario and on the simulator itself. One may need to define, for example, OD matrices and route choice model parameters for the demand and speed/density relationship functions or driving behaviour model parameters for the supply.

The essential challenge then becomes the calibration of all the supply and demand parameters in order to reflect the real phenomena. Different requirements are expected for dynamic traffic assignment models (DTA) (e.g. Ben-Akiva et al., 2010a) and for microscopic traffic simulation (e.g. Yang and Koutsopoulos, 1996). For example, DTA models usually utilise mesoscopic demand and supply simulator components, that employ a mix of microscopic and macroscopic models to capture the decision of the travellers and the movement of vehicles throughout the network. They consider the (often thousands or tens of thousands) OD flows in the network as inputs that need to be calibrated. Similarly, in the supply side, segment output capacities are among the parameters that need to be calibrated, and these are easily in the order of thousands. Microscopic traffic simulator models also require OD flows as inputs, but on the supply side they require a much smaller number of parameters to be calibrated (used in the individual models, such as car–following, merging, lane–changing) (Toledo et al., 2007).

Overall, a traffic model may contain hundreds of thousands of parameters, and, in a complex network with a large population, the simulation itself is not computationally negligible. Moreover, due to the (generally unknown) nature of the search space, this becomes a complex optimization problem. Given the available data (e.g. traffic volumes, densities and speeds from conventional counters, but also travel times or route–choice fractions), the optimization problem consists of estimating the parameters that minimize the difference between sensed values and simulated values. Of course, the computational costs forbid brute force solutions and the lack of a precise analytical model frustrates the use of deterministic methods. We need a methodology that is parsimonious with the simulation runs, yet capable of making an efficient search in a stochastic fashion.

The Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm (Spall, 1998) was designed to address these issues. Briefly, at each iteration, it generates a pair of new vectors to inspect (i.e. run the simulation for), where each individual vector element, or parameter, is determined by a perturbation with respect to the original value. The particularity is that all parameters are perturbed simultaneously in a stochastic, pair-wise symmetric fashion: the new pair of values of parameter i will be $a_i \pm p_i$, being a_i the original value of parameter i and p_i the perturbation. The gradient is then calculated taking into account the respective simulation results.

The characteristics of SPSA allow for another functionality, introduced by Balakrishna (2006), that is to simultaneous calibrate all supply and demand parameters together, as opposed to have them calibrated separately (and possibly iteratively). Balakrishna (2006) have shown that simultaneous approaches outperform the traditional iterative framework when applied to the calibration of DTA models. SPSA and its variations have since been applied extensively in the field of traffic simulation model calibration. Balakrishna et al. (2007) apply SPSA for the simultaneous calibration of the demand and supply parameters and inputs to the microscopic traffic simulation model MITSIMLab (Yang and Koutsopoulos, 1996) using the network of Lower Westchester County, NY, to demonstrate the feasibility, application, and benefits of the proposed methodology. Ma et al. (2007) compare the performance of SPSA against a genetic algorithm (GA) and a trial-and-error iterative adjustment algorithm (IA) for the calibration of a microscopic simulation model in a northern California network and conclude that SPSA can achieve the same level of accuracy as the other two with a significantly shorter running time. Vaze et al. (2009) present a framework for the joint calibration of demand and supply model parameters of DTA models using multiple sources of traffic information. The calibration problem has been formulated as a stochastic optimization framework and SPSA was found to outperform competing algorithms, based on results using both counts and travel time measurements obtained from automated vehicle identification systems on a synthetic network and the network of Lower Westchester County, NY.

Huang et al. (2010) applied SPSA for the calibration of dynamic emission models. This research uses a microscopic traffic simulator and the aggregate estimation ARTEMIS as a standard reference. Lee and Ozbay (2008) propose a Bayesian calibration methodology and applied a modified SPSA algorithm to solve the calibration problem of a cell transmission based macroscopic traffic model. In this formulation, the probability distributions of model parame-

ters are considered instead of their point values. Paz et al. (2012) calibrate all the parameters in CORSIM models simultaneously using SPSA and demonstrate its effectiveness.

In the first case study presented in this research, using a mesoscopic DTA model for the entire expressway system in Singapore, it was found that, although SPSA maintained its computational efficiency, its performance in terms of convergence rate and long run accuracy deteriorated significantly, as the problem scale increased. The errors stopped to decrease at relatively high values. Different values of algorithm parameters were tested and adaptive step sizes (a_k) were implemented. However, no significant improvement was made (Lu, 2014). This led us to believe that SPSA itself has fundamental limitations, when applied to very large scale, noisy problems without analytical representation and with correlated parameters and measurements, as identified in Lu et al. (2015); Cipriani et al. (2011); Cantelmo et al. (2014). One of those limitations refers to the agnostic perspective on the correlation structure between the variables involved and the observations. It is assumed that they are all equally co-dependent, but in practice it is rarely the case.

To complicate matters further, if we consider also the temporal dimension, changing the value of an OD flow at time t_0 may have little influence to the network measurements after a while (e.g. time t_{30}), and will have absolutely no influence to preceding measurements (e.g. t_{-30}). Therefore, in a traffic simulation off–line calibration problem with a large scale network and a large number of intervals, a large number of uncorrelated measurements may introduce an excessive amount of disturbing noise, which makes it very hard to estimate the actual influence from the perturbation of each of the parameter value to determine a good direction and amplitude to move.

This gradient approximation error raises performance issues for SPSA, when a large-scale system has sparse correlations between parameters and measurements, such as the traffic system. Successful attempts have been made to modify or extend the existing SPSA algorithm to improve its performance on DTA calibration, such as incorporating transition equations of OD flows in the objective function (Balakrishna and Koutsopoulos, 2008) and using asymmetric estimation and adopting polynomial interpolation to the algorithm step size (Cipriani et al., 2011). Cantelmo et al. (2014) also compare the latter extension to its second order and perform a sensitivity analysis of the parameters for the algorithm. However, they all targeted only the estimation of OD flows. More importantly, the serious concern of gradient approximation error could not be addressed with these modifications. To overcome this problem, knowledge about the existing correlations in the system should be incorporated in a sophisticated way into the joint demand–supply calibration framework to reduce the gradient approximation error. Such knowledge is lost in applications of the original SPSA algorithm, when time–dependent, location–specific error terms are transformed into a single scalar. Besides SPSA extensions, other gradient approximation methods, which consider (implicitly) correlations, have also been proposed, such as the work of Frederix et al. (2011, 2013).

The proposal of Weighted SPSA (W–SPSA) from Lu et al. (2015) aims precisely to overcome these limitations by relying on a weight matrix, $\mathbf{W}$, that represents the appropriate correlation structure. While the authors demonstrated the concept and successfully compared it with SPSA in several settings, the treatment of the essential ingredient (i.e. the $\mathbf{W}$ matrix) was not systematic, which considerably limits the applicability of the algorithm.

In this paper, we propose the full framework for W–SPSA, with a complete formulation (effectively a generalization of SPSA) and with a sensible methodology for obtaining a reliable $\mathbf{W}$ matrix. To demonstrate the flexibility and generality of the approach, we have applied it to two applications: (i) the simultaneous calibration of all input and parameters of the DynaMIT framework (Ben-Akiva et al., 2001, 2010a), a DTA model that serves for real-time and planning purposes, for a large–scale problem in Singapore, and (ii) the calibration of the demand inputs and supply parameters of MITSIMLab (Yang and Koutsopoulos, 1996; Ben-Akiva et al., 2010b), a microscopic traffic simulation, for a very demanding safety–related application, requiring thousands of separate calibration replications.

The applicability of W–SPSA goes much beyond traffic simulation calibration. It applies to any circumstance where there is a complex correlation structure in a model that has no known analytical formulation and that is computationally costly. Thus, it fits into the stochastic optimization realm which has wide well–known applications in engineering.

2. Methodology

2.1. General problem formulation

Let the time period of interest be divided into intervals $\mathcal{H} = \{1, 2, ..., H\}$. The off–line calibration problem is formulated using the following notation:

- $\boldsymbol{x}$: Time-dependent model parameters, e.g., OD flows, $\boldsymbol{x} = \{\boldsymbol{x_h}\}, \forall h \in \mathcal{H}$
- $\boldsymbol{\beta}$: Other model parameters, e.g., supply model parameters
- $\boldsymbol{M^o}$: Observed time-dependent sensor measurements, $\boldsymbol{M^o} = \{\boldsymbol{M^o_h}\}$
- $\boldsymbol{M^s}$: Simulated time-dependent sensor measurements, $\boldsymbol{M^s} = \{\boldsymbol{M^s_h}\}$
- $\boldsymbol{x^a}$: *A priori* time-dependent parameter values, $\boldsymbol{x^a} = \{\boldsymbol{x^a_h}\}$
- $\boldsymbol{\beta^a}$: *A priori* values of other model parameters
- $\boldsymbol{G}$: Road network and other fixed supply parameters, $\boldsymbol{G} = \{\boldsymbol{G_h}\}$

The off–line calibration problem is formulated as an optimization problem to minimize an objective function over the parameter space:

$$\underset{\boldsymbol{x},\boldsymbol{\beta}}{\text{minimize}} \quad z(\boldsymbol{M^o}, \boldsymbol{M^s}, \boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{x^a}, \boldsymbol{\beta^a}) \tag{1}$$

which can be operationalized as follows:

$$\underset{\boldsymbol{x},\boldsymbol{\beta}}{\text{minimize}} \sum_{h=1}^{H} [z_1(\boldsymbol{M^o_h}, \boldsymbol{M^s_h}) + z_2(\boldsymbol{x_h}, \boldsymbol{x^a_h})] + z_3(\boldsymbol{\beta}, \boldsymbol{\beta^a}) \tag{2}$$

$$\text{s.t.:} \qquad \boldsymbol{M^s_h} = f(\boldsymbol{x_1}, ..., \boldsymbol{x_h}; \boldsymbol{\beta}; \boldsymbol{G_1}, ..., \boldsymbol{G_h}) \tag{3}$$

$$\boldsymbol{l_{x_h}} \leq \boldsymbol{x_h} \leq \boldsymbol{u_{x_h}}, \quad \boldsymbol{l_\beta} \leq \boldsymbol{\beta} \leq \boldsymbol{u_\beta} \tag{4}$$

Equation 2 is the objective function for the minimization problem, where z_1 measures the goodness–of–fit between observed sensor measurements and simulated sensor measurements, and z_2 and z_3 compare estimated values with *a priori* values (z_1, z_2 and z_3 depend on the chosen approach). Equation 3 represents the relationship between simulated measurement values and model inputs, where f is the traffic simulation model. Equations 4 specify the boundaries for the estimated parameter values. Under the generalized least squares framework, Equation 2 becomes:

$$\underset{\boldsymbol{x},\boldsymbol{\beta}}{\text{minimize}} \sum_{h=1}^{H} [\boldsymbol{\epsilon}^\top_{\boldsymbol{M_h}} \boldsymbol{\Omega}^{-1}_{\boldsymbol{M_h}} \boldsymbol{\epsilon}_{\boldsymbol{M_h}} + \boldsymbol{\epsilon}^\top_{\boldsymbol{x_h}} \boldsymbol{\Omega}^{-1}_{\boldsymbol{x_h}} \boldsymbol{e}_{\boldsymbol{x_h}}] + \boldsymbol{\epsilon}^\top_{\boldsymbol{\beta}} \boldsymbol{\Omega}^{-1}_{\boldsymbol{\beta}} \boldsymbol{\epsilon}_{\boldsymbol{\beta}} \tag{5}$$

where $\boldsymbol{\epsilon}_{\boldsymbol{M_h}} = \boldsymbol{M^o_h} - \boldsymbol{M^s_h}$, $\boldsymbol{\epsilon}_{\boldsymbol{x_h}} = \boldsymbol{x_h} - \boldsymbol{x^a_h}$, $\boldsymbol{\epsilon}_{\boldsymbol{\beta}} = \boldsymbol{\beta} - \boldsymbol{\beta^a}$ and $\boldsymbol{\Omega}_{\boldsymbol{M_h}}$, $\boldsymbol{\Omega}_{\boldsymbol{x_h}}$, $\boldsymbol{\Omega}_{\boldsymbol{\beta}}$ are variance–covariance matrices.

The subscripts of intervals (i.e., h) highlight the time dependent feature of the traffic simulation model calibration problem. For a more straightforward illustration of the solution algorithm, we further generalize the framework by removing the interval subscripts, and defining it as:

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_H} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_P \end{bmatrix}, \boldsymbol{\epsilon_M} = \begin{bmatrix} M^o_1 - M^s_1 \\ \vdots \\ M^o_D - M^s_D \end{bmatrix} = \begin{bmatrix} \epsilon_{M1} \\ \vdots \\ \epsilon_{MD} \end{bmatrix} = F_M(\boldsymbol{\theta}; \boldsymbol{M^o}; \boldsymbol{G}), \boldsymbol{\epsilon_\theta} = \begin{bmatrix} \theta_1 - \theta^a_1 \\ \vdots \\ \theta_P - \theta^a_P \end{bmatrix} = \begin{bmatrix} \epsilon_{\theta 1} \\ \vdots \\ \epsilon_{\theta P} \end{bmatrix} = F_\theta(\boldsymbol{\theta}; \boldsymbol{\theta^a}; \boldsymbol{G}) \tag{6}$$

$\boldsymbol{\epsilon_M}$ is a vector of deviations between observed measurement values and simulated measurement values. F_M is a function that maps $\boldsymbol{\theta}$, $\boldsymbol{M^o}$ and the road network $\boldsymbol{G}$ to $\boldsymbol{\epsilon_M}$. $\boldsymbol{\epsilon_\theta}$ is a vector of deviations between *a priori* parameter values and estimated parameter values. F_θ is the function that maps the vector of the *a priori* parameters $\boldsymbol{\theta^a}$, $\boldsymbol{\theta}$ and $\boldsymbol{G}$ to $\boldsymbol{\epsilon_\theta}$. The length of $\boldsymbol{\theta}$ and $\boldsymbol{\epsilon_\theta}$, P, equals the total number of parameters to calibrate (note that one time–dependent OD flow in two different intervals is considered as two different parameters). The length of $\boldsymbol{\epsilon_M}$, D, equals the total number of measurements.

The further generalized problem formulation is:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad z(\boldsymbol{\theta}) \Rightarrow \underset{\boldsymbol{\theta}}{\text{minimize}} \quad \boldsymbol{\epsilon}^\top_{\boldsymbol{M}} \boldsymbol{\Omega}^{-1}_{\boldsymbol{M}} \boldsymbol{\epsilon_M} + \boldsymbol{\epsilon}^\top_{\boldsymbol{\theta}} \boldsymbol{\Omega}^{-1}_{\boldsymbol{\theta}} \boldsymbol{\epsilon_\theta} \tag{7}$$

subject to:

$$\boldsymbol{\epsilon_M} = F_M(\boldsymbol{\theta}; \boldsymbol{M^o}; \boldsymbol{G}), \qquad \boldsymbol{\epsilon_\theta} = F_\theta(\boldsymbol{\theta}; \boldsymbol{\theta^a}; \boldsymbol{G}), \qquad \boldsymbol{l_\theta} \leq \boldsymbol{\theta} \leq \boldsymbol{u_\theta} \tag{8}$$

To make the objective function in Equation 7 identical to the objective function in Equation 5, $\boldsymbol{\Omega_M}$ should be a block diagonal matrix with $\boldsymbol{\Omega_{M_h}}$ as its diagonal elements. The same applies to $\boldsymbol{\Omega_\theta}$, which is a block diagonal matrix with $\boldsymbol{\Omega_{x_h}}$ and $\boldsymbol{\Omega_\beta}$ at its diagonal. However, in a more general formulation that considers all the correlations across different time intervals and different types of parameters, $\boldsymbol{\Omega_M}$ and $\boldsymbol{\Omega_\theta}$ are general variance–covariance matrices. The structure of these matrices can be simplified in a number of ways, such as diagonal matrices (i.e. considering only the variances) or even identity matrices (i.e. assuming constant variance).

2.2. The SPSA algorithm

Stochastic approximation (SA) methods are a family of iterative stochastic optimization algorithms used in error function minimization when the objective function has no known analytical form and can only be estimated with noisy observations. The general iterative form of SA is:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{k+1}} = \hat{\boldsymbol{\theta}}_{\boldsymbol{k}} - a_k \hat{\boldsymbol{g}}_{\boldsymbol{k}}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) \tag{9}$$

where $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}$ is the estimate of the decision vector in the k_{th} iteration of the algorithm and $\hat{\boldsymbol{g}}_{\boldsymbol{k}}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$ is the estimated gradient at $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}$. a_k is a usually small number that gets smaller as k becomes larger, known as the k_{th} step size in a gain sequence:

$$a_k = \frac{a}{(A + k + a)^\alpha} \tag{10}$$

where a, α and A are algorithm parameters. Different approaches have been proposed to approximate the gradient. In the finite–difference (FD) schemes, the gradient is estimated by perturbing the parameters in the decision vector one at a time, evaluating the objective function values and computing the gradient as:

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) = \frac{z(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} + c_k \boldsymbol{e_i}) - z(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} - c_k \boldsymbol{e_i})}{2c_k} \tag{11}$$

where $\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$ is the i_{th} element in the gradient vector $\hat{\boldsymbol{g}}_{\boldsymbol{k}}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$, and $\boldsymbol{e_i}$ is a vector with 1 at i_{th} location and 0 elsewhere. Each parameter is perturbed with an amplitude of c_k to two opposite directions, with c_k defined as:

$$c_k = \frac{c}{(k + 1)^\gamma} \tag{12}$$

where c and γ are algorithm parameters.

This approach is capable of estimating high quality gradient vectors in non–analytical problems with noisy observations. It is, however, not computationally efficient for large–scale problems. The number of objective function evaluations within one algorithm iteration is $2P$, where P is the total number of parameters in the decision vector. In simulation–based models, one objective function evaluation involves the running of the simulation from beginning to end, which may take minutes. A mid–sized model often consists of thousands of model parameters to calibrate, which may lead to days of run time just for one iteration of the algorithm. Therefore, applying FDSA in off–line calibration problems of large–scale traffic simulation models is often infeasible in terms of computational overhead.

Spall (1992, 1998) proposes an innovative solution to this problem: simultaneous perturbation stochastic approximation (SPSA). SPSA efficiently estimates the gradient by perturbing all the parameters in the decision vector $\boldsymbol{\theta}$ simultaneously and the approximation of gradient needs only two function evaluations regardless of the number of parameters:

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) = \frac{z(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} + c_k \boldsymbol{\Delta_k}) - z(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} - c_k \boldsymbol{\Delta_k})}{2c_k \Delta_{ki}} \tag{13}$$

where $\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$ is the i_{th} element in the gradient vector $\hat{\boldsymbol{g}}_{\boldsymbol{k}}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}})$. c_k is the perturbation amplitude (same as in the FD approach). $\boldsymbol{\Delta}_{\boldsymbol{k}}$ is a random perturbation vector, generated through a Bernoulli process (or using other appropriate distributions) with values of +1 and -1 with equal probabilities.

SPSA provides a huge saving of computational time due to its constant number of perturbations for the gradient approximation. In terms of convergence performance, Spall (1998) argues that SPSA follows a path that is expected to deviate only slightly from that of FDSA. In other words, SPSA performs as good as FDSA, while having a P–fold time saving. With no or little noise, FDSA is expected to follow the true descent to the optimal. SPSA may have approximated gradients that differ from the true gradients, but they are almost unbiased. With larger noise, neither FDSA and SPSA follow the deepest decent directions, but SPSA stays in a path close to the optimal. The detailed step–by–step SPSA workflow is described below:

1. Set the current step $k = 0$, so that the initial values in the decision vector $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} = \hat{\boldsymbol{\theta}}_{\boldsymbol{0}}$ (usually the historical values or the most recent calibrated values). Decide the values of the algorithm parameters α, γ, a, A, and c.
2. Evaluate the initial objective function value z_0 by running the simulator with $\hat{\boldsymbol{\theta}}_{\boldsymbol{0}}$ as the input parameters.
3. Update $k = k + 1$. Calculate a_k and c_k based on Equations 10 and 12.
4. Generate the independent random perturbation vector $\boldsymbol{\Delta}_{\boldsymbol{k}}$.
5. Evaluate the objective function values at two points: $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} + c_k\boldsymbol{\Delta}_{\boldsymbol{k}}$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} - c_k\boldsymbol{\Delta}_{\boldsymbol{k}}$. Parameter boundaries are imposed before the objective function evaluation.
6. Approximate the gradient vector using Equation 13.
7. Calculate $\hat{\boldsymbol{\theta}}_{\boldsymbol{k+1}}$ using Equation 9.
8. If converged, stop the process. If not, return to step 3.

2.3. Shortcomings of SPSA in the calibration of large–scale traffic simulation models

For a better illustration of the shortcomings of SPSA and motivation of our improvement idea, we assume the inverted variance–covariance matrix Ω_M^{-1} is the identity matrix and Ω_θ^{-1} is a matrix with all zeros. This is equivalent to the situation where we don't consider the deviations between estimated parameter values and their *a priori* values. At the same time, we consider the deviations between simulated and observed measurement values in an ordinary least squares (OLS) way. The analysis and methodology can be easily extended to the general formulation in Equation 7.

The simplified objective function is:

$$z(\boldsymbol{\theta}) = \sum_{j=1}^{D} \epsilon_{Mj}^2 \tag{14}$$

We denote

$$\boldsymbol{\epsilon}_{\boldsymbol{Mk}}^{+} = F_M(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} + c_k\boldsymbol{\Delta}_{\boldsymbol{k}}; \boldsymbol{M}^{\boldsymbol{o}}; G) \tag{15}$$

$$\boldsymbol{\epsilon}_{\boldsymbol{Mk}}^{-} = F_M(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}} - c_k\boldsymbol{\Delta}_{\boldsymbol{k}}; \boldsymbol{M}^{\boldsymbol{o}}; G) \tag{16}$$

which are the deviation vectors obtained from the two perturbations. ϵ_{Mkj}^{+} is the j_{th} element in $\boldsymbol{\epsilon}_{\boldsymbol{Mk}}^{+}$ and ϵ_{Mkj}^{-} is the j_{th} element in $\boldsymbol{\epsilon}_{\boldsymbol{Mk}}^{-}$. Based on our simplified objective function in Equation 14, Equation 13 can be rewritten as

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) = \frac{\sum_{j=1}^{D}[(\epsilon_{Mkj}^{+})^2 - (\epsilon_{Mkj}^{-})^2]}{2c_k\Delta_{ki}} \tag{17}$$

The reason for the performance deterioration of SPSA was found to be related to a gradient approximation error that increased rapidly with the problem scale. The source of this error was not the stochasticity in the simulation models, or the inconsistency in the observed data due to measurement error, but the way SPSA estimates gradients. In each iteration, the gradient estimation process essentially tries to find a direction and amplitude for each parameter value in the decision vector to move. This is achieved by comparing the influence to the system caused by perturbing each of the parameter value in two opposite directions. Given our formulation in Equation 14, in SPSA the influence caused

by perturbing the value of a specific parameter is determined by a scalar: the sum of all the distances between model outputs and corresponding observed measurements. As all the parameters are perturbed at the same time, the change in a measurement value may or may not be caused by this specific parameter. This may not be a major issue in systems where each parameter is highly correlated to most of the measurements, because, in that case, the change in each parameter is responsible for the change of almost every measurement value.

However, in a real–world traffic system, correlations between model parameters and measurements are often sparse. In the spatial dimension, most of the model parameters tend to have a relatively local effect to the system. For example, by changing the OD flow of one OD pair, only traffic volume measurements along the paths between this OD pair will be influence directly, while nearby measurements may be indirectly affected. As the distance between the sensor and the paths increases, this influence will decrease. In the temporal dimension, changing the value of an OD flow at 7:30am may have little influence to the network measurements after e.g. 8:30am, and will have absolutely no influence to the measurements before 7:30am. Therefore, in the off–line calibration problem with a large scale network and a large number of intervals, a large number of uncorrelated measurements introduce a great amount of disturbing noise.

Assume the length of an interval is 15 minute and the simulation period consists of 20 intervals. Assume, further, that the longest travel time in this network is 30 minutes, i.e. two intervals, and that 6 sensors provide time–dependent count data. Therefore, we have in total 120 measurements (20 intervals of 6 measurements). Consider an OD flow i for a specific interval, i.e. the i_{th} element in $\boldsymbol{\theta}$. At the k_{th} SPSA iteration, the element in the gradient vector for this OD flow is approximated as:

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) = \frac{\sum_{j=1}^{120}[(\epsilon_{Mkj}^{+})^2 - (\epsilon_{Mkj}^{-})^2]}{2c_k\Delta_{ki}} \tag{18}$$

The numerator tries to approximate the influence on measurements caused by changing this specific OD flow, where the term $(\epsilon_{Mkj}^{+})^2 - (\epsilon_{Mkj}^{-})^2$ is decided by the change in the j_{th} simulated measurement. However, only counts from few sensors in the network are affected by the change in this OD flow. The changes in other sensors are primarily caused by other parameters and introduce disturbing noise. At the same time, the changes of measurement values from intervals before this interval are totally irrelevant to this change of the OD flow. Based on our longest trip assumption, the changes of measurement values in later intervals are also largely irrelevant to this OD flow change. Therefore, in the numerator we sum up values from 120 measurements but most of them are irrelevant to the change of the current OD flow and a great amount of noise is thus introduced to the gradient estimation. In other words, the number used to compute the approximated gradient consists primarily of noise.

In real world traffic simulation applications, in order to provide useful information for planners and travelers, the scale of network and the number of intervals to consider (usually across an entire day) is much larger, and therefore the gradient approximation problem will be more serious. To solve this problem, a weighted gradient approximation method and the corresponding solution algorithm, W–SPSA, is proposed in the next section.

2.4. The Weighted SPSA algorithm

A weighted gradient approximation method is introduced to exclude the negative influence of irrelevant measurements in the gradient approximation process of SPSA. In this approach, measurements are considered in a weighted manner, based on their relevance to a parameter:

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) = \frac{\sum_{j=1}^{D} w_{ji}[(\epsilon_{Mkj}^{+})^2 - (\epsilon_{Mkj}^{-})^2]}{2c_k\Delta_{ki}} = \frac{1}{2c_k\Delta_{ki}}\boldsymbol{W_i}^{\top}\begin{bmatrix}(\epsilon_{Mk1}^{+})^2 - (\epsilon_{Mk1}^{-})^2\\ \vdots \\ (\epsilon_{MkD}^{+})^2 - (\epsilon_{MkD}^{-})^2\end{bmatrix} \tag{19}$$

where w_{ji} is the element at the j_{th} row and i_{th} column of a $D \times P$ weight matrix. $\boldsymbol{W_i}$ is the i_{th} column of the matrix. As introduced in the previous subsection, D is the number of deviations (measurements plus historical parameter values) and P is the number of parameters.

$$\boldsymbol{W_i}^{\top} = \begin{bmatrix} w_{1i} \dots w_{ji} \dots w_{Di}\end{bmatrix} \tag{20}$$

For a change in parameter θ_i, w_{ij} represents the relative magnitude of change in measurement j, compared to other measurements. By putting more weights on relevant measurements and less/no weights on less relevant measurements/irrelevant measurements, we can effectively reduce the estimation error and therefore provide a better gradient approximation.

Under the generalised least squares (GLS) formulation in Equation 7, Equation 19 becomes

$$\hat{g}_{ki}(\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}) = \frac{E_M + E_\theta}{2c_k\Delta_{ki}} \tag{21}$$

where

$$\begin{aligned} E_M =&[(\boldsymbol{\epsilon}^{+}_{\boldsymbol{Mk}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}})]\boldsymbol{\Omega}^{-1}_{\boldsymbol{M}}[(\boldsymbol{\epsilon}^{+}_{\boldsymbol{Mk}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}})]^\top - \\ &[(\boldsymbol{\epsilon}^{-}_{\boldsymbol{Mk}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}})]\boldsymbol{\Omega}^{-1}_{\boldsymbol{M}}[(\boldsymbol{\epsilon}^{-}_{\boldsymbol{Mk}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}})]^\top \end{aligned} \tag{22}$$

$$\begin{aligned} E_\theta =&[(\boldsymbol{\epsilon}^{+}_{\boldsymbol{\theta k}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{\theta i}})]\boldsymbol{\Omega}^{-1}_{\boldsymbol{\theta}}[(\boldsymbol{\epsilon}^{+}_{\boldsymbol{\theta k}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{\theta i}})]^\top - \\ &[(\boldsymbol{\epsilon}^{-}_{\boldsymbol{\theta k}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{\theta i}})]\boldsymbol{\Omega}^{-1}_{\boldsymbol{\theta}}[(\boldsymbol{\epsilon}^{-}_{\boldsymbol{\theta k}})^\top \boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{\theta i}})]^\top \end{aligned} \tag{23}$$

where $\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}}$ is the element-wise square root of the i_{th} column of the weight matrix for measurements $\boldsymbol{W_M}$. $\boldsymbol{diag}(\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}})$ is the diagonal matrix built from $\boldsymbol{W}^{\circ\frac{1}{2}}_{\boldsymbol{Mi}}$. $\boldsymbol{W_\theta}$ is the weight matrix for historical values. Typically $\boldsymbol{W_\theta}$ is an identity matrix, because changing the value of a parameter only influences the deviation between this specific parameter's estimated value and its historical values. The definitions of $\boldsymbol{\epsilon}^{+}_{\boldsymbol{\theta k}}$ and $\boldsymbol{\epsilon}^{-}_{\boldsymbol{\theta k}}$ are similar to those of $\boldsymbol{\epsilon}^{+}_{\boldsymbol{Mk}}$ and $\boldsymbol{\epsilon}^{-}_{\boldsymbol{Mk}}$.

These relevance measurements can be obtained from the adjusted Jacobian matrix $\boldsymbol{J}$, where each element is replaced by the absolute value from the traditional Jacobian matrix. The absolute value is used because we are only interested in the magnitude.

$$\boldsymbol{J} = \begin{bmatrix} |\frac{\partial\epsilon_1}{\partial\theta_1}| & \dots & |\frac{\partial\epsilon_1}{\partial\theta_i}| & \dots & |\frac{\partial\epsilon_1}{\partial\theta_P}| \\ \vdots & & \vdots & & \vdots \\ |\frac{\partial\epsilon_j}{\partial\theta_1}| & \dots & |\frac{\partial\epsilon_j}{\partial\theta_i}| & \dots & |\frac{\partial\epsilon_j}{\partial\theta_P}| \\ \vdots & & \vdots & & \vdots \\ |\frac{\partial\epsilon_D}{\partial\theta_1}| & \dots & |\frac{\partial\epsilon_D}{\partial\theta_i}| & \dots & |\frac{\partial\epsilon_D}{\partial\theta_P}| \end{bmatrix} \tag{24}$$

The weight matrix $\boldsymbol{W}$ is a local approximation of the Jacobian matrix at a specific point $\boldsymbol{\theta_k}$. Due to lack of closed-form relationship between parameters and measurements, a local linear approximation of $\frac{\partial\epsilon_j}{\partial\theta_i}$ is commonly used.

$$\boldsymbol{W}|_{\boldsymbol{\theta_k}} = \hat{\boldsymbol{J}}|_{\boldsymbol{\theta_k}} \tag{25}$$

Weighted SPSA, or *W–SPSA*, is the proposed solution algorithm for the off–line calibration problem that applies the weighted gradient estimation in the stochastic path searching process. Typically, off–line calibration problems are highly non–linear and the Jacobian matrix changes with current parameter values. Therefore, the weight matrix should be re–estimated during the calibration process along with the change of current estimated parameter values. The frequency of this process depends on the specific problem, the stage of the calibration process, and the amount of available computational power. The results from the previous off–line calibration stage are commonly used as the input of weight matrix re–estimation in next iterations. However, this re–estimation process is not necessary under a special case of W–SPSA: SPSA (the weight matrix is an all-one matrix). The next section discusses different approaches to estimate the weight matrix and their pros and cons.

3. Estimation of weight matrices

The accurate estimation of weight matrices is the key to successfully applying W–SPSA for the off–line calibration of traffic estimation models. The ideal weight matrix estimation approach should be easy to implement, efficient to

execute, and able to provide accurate estimation for different types of parameters and measurements. Naturally, there is no silver bullet for this problem, as each case has its own complexity, data and domain constraints. In this paper, we propose four fundamental alternatives to systematically estimate the weight matrix:

- Analytical derivation
- Simulation–based approximation
- Numerical approximation
- Heuristics–based approximation

In practice, however, such alternatives end up combined or adapted to the problem at hand in one way or the other, so we propose two additional methods, namely hybrid and composite.

3.1. Analytical derivation

If known analytical relationships between parameters and measurements exist, it is possible to estimate the weights based on an approach called analytical derivation. Network knowledge, including network topology, path choice set, equilibrium link travel time, and route choice model, is usually required as input to this approach. However, the complete forms of these relationships are typically unknown and the required inputs change with the model parameter values. Even when these relations can be known or assumed, the problem can easily become intractable, due to their complexity. Therefore, an approximated linear relationship is commonly used with the current estimated parameter values.

For example, to calculate the weights between OD flows and link traffic flows (usually measured by loop detectors in the form of sensor counts), dynamic assignment matrices are commonly used. The assignment matrix stores the way in which each OD flow is assigned to different links (sensors), and theoretically, it can be analytically calculated using network topology, path choice set, current route choice model and equilibrium travel times (Ashok and Ben-Akiva, 2002). However, it is recognised that the complexity of the problem at hand can quickly lead to intractable situations (Ashok and Ben-Akiva, 2002).

The problem becomes even harder when considering the relationship between parameters with less direct impact, such as the impact of e.g. capacities or behavioural model parameters to travel times or counts in specific locations in the network. For such types of parameter–measurement correlations (e.g. capacities vs. counts), similar ideas can be applied, but more sophisticated analytical derivations may be required to capture the usually indirect and non–linear relationships.

Whenever tractable, the analytical approximation approach may have the advantage of computational efficiency, as no simulation run is required for the weight matrix estimation. Analytical derivation may also be able to provide very accurate weights, if the relationships between parameters and measurements are relatively straightforward and linear.

3.2. Simulation–based approximation

The analytical derivation method relies on the direct observation of parameter-measurement relationships, and the more observable the system is, the less complex the problem may become. When the needed dynamic observations in the network are unobservable, a simulator can help uncover useful relationships. For the case of OD flows, simulation can be used to compute a matrix approximation. In such a case, the assignment matrix is endogenous to the simulation model itself and cannot be analytically estimated. Ashok and Ben-Akiva (2000) developed such an approach using an iterative computation of the assignment matrix. For a given OD matrix, a DTA simulation model is used to compute an estimated assignment matrix. This simulated assignment matrix would then be used to update the OD flows and the existing measurements estimates. The entire process may be initiated using the one–step predicted OD flows generated at the end of the previous interval or using a seed OD matrix. Due to the stochasticity of the simulation model, for each perturbation of the W-SPSA where the weight matrix has to be estimated, the simulation has to be replicated R times (Antoniou et al., 2014). The simulated measurements are then used to estimate the deviations used in the estimation of a local linear approximation of the weight matrix.

Figure 1 illustrates the general flow of this approach. P is the total number of parameters, D is the total number of measurements and R is the total number of replications of the simulation run at each iteration of W–SPSA. M_j^{S+}

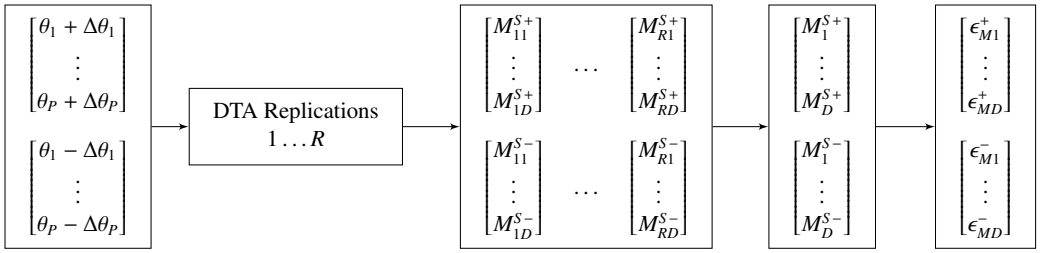

Fig. 1: Simulation-based approximation logic

(M_j^{S-}) is the combined simulated measurement of the j_{th} measurements M_{rj}^{S+} (M_{rj}^{S-}) for all replications R, using the parameter values after the "+"("-") perturbation.

The advantage of such a model is that the unobserved measurements are estimated by a traffic assignment model, which obviates the need for the analytical derivation of complex relationships typically present in large networks, where the link between path choice sets, the route choice model and the available measurements has a complex level of interaction; or when the assignment measurements are unobservable.

3.3. Numerical approximation

Taking advantage of the use of simulation, another improved approach to obtain a weight matrix without going through the analytical derivation is to use the simulator to approximate the Jacobian matrix at the current point through numerical experiments. We call this approach a numerical approximation and the logic of a single numerical experiment is also similar to Figure 1. However, in this approach several experiments $n = 1...N$ at each W–SPSA iteration are carried out. In one experiment, all the parameters are perturbed at the same time near the current value to two opposite directions, but $\Delta\theta_{ni}$ is randomly generated and independent across different experiments (it typically has a pre–defined amplitude and a randomly generated sign). The changes of simulated measurement values under each of the two perturbed parameter vectors are evaluated by running the simulation model, again R times. Each element in the Jacobian matrix is approximated as the change in a measurement value divided by the change in a parameter. After running N independent experiments, the estimated weights are obtained by averaging the result from each experiment:

$$w_{ij} = \frac{\partial \epsilon_{M_j}}{\partial \theta_i} \simeq \frac{\sum_{n=1}^{N} \frac{\epsilon_{M_n^S j}}{\Delta\theta_{ni}}}{N} \tag{26}$$

where M_{nj}^S is the simulated measurement value of the d^{th} measurement in the n^{th} experiment. The estimation result is expected to improve with the increase of N and to be unbiased.

Under this framework, different types of parameters and measurements are treated in a same manner. At the same time, no explicit analytical network knowledge is required in the estimation. Therefore, this approach covers all different kinds of parameter–measurement pairs using a single, and simple, method. However, the capability of obtaining accurate results depends highly on a big enough number of experiments and each experiment requires two simulation runs, which leads to intensive computational overhead. Fortunately, this approach can be easily parallelized, as the experiments are completely independent. If enough computers are available, a simple distribution algorithm should suffice to speed up this process significantly.

3.4. Heuristics–based approximation

Domain knowledge and intuition can also play a fundamental role in estimating the **W** matrix. An obvious example is the use of an assignment matrix (for example, obtained via simulation, as discussed above) to represent the correlations between ODs and sensors. Other examples could relate supply parameters with measurements that are spatially correlated (e.g. capacity parameters for some parameters with flow data from nearby sensors).

The general principle of the heuristics–based approximation is to use domain knowledge to map together parameters and measurements that are logically related. In the presence of ambiguity, a conservative attitude is to assume correlation (e.g. $w_{ij} = 1$).

3.5. Hybrid method

To better take advantage of the pros of each estimation method, a hybrid weight matrix can be built and adjusted during the calibration process:

$$\boldsymbol{W_h} = \lambda_a \boldsymbol{W_a} + \lambda_n \boldsymbol{W_n} + \lambda_o \boldsymbol{W_o} \tag{27}$$

where $\boldsymbol{W_h}$ is the hybrid weight matrix, which equals the weighted sum of a weight matrix estimated using an analytical approach ($\boldsymbol{W_a}$), a weight matrix estimated from numerical approach ($\boldsymbol{W_n}$) and an all–one weight matrix ($\boldsymbol{W_o}$). The weights λ depend on the relative confidence on each part and can change in different stages of the calibration process.

The estimation of the weights λ could be made empirically (e.g. by trial and error) or through a systematic process that searches for an acceptable combination (e.g. meta–heuristics). The complexity of this specific problem is not considerably high, as only two parameters are involved in practice (the third one is implicit, e.g. $\lambda_a = 1 - \lambda_n - \lambda_o$). A simple method could comprise a grid–search approach, relying on running a set of W–SPSA runs based on a discrete set of values.

In empirical applications, high quality weight matrices may not be easy to obtain through applying any single approach. This hybrid method provides some insights about possible ways to use different weight matrix estimation methods at the same time. The best way to estimate weight matrices is problem–specific and requires a good understanding of the characteristics of the problem, in–depth analysis of the parameters and measurements, and some engineering judgement.

3.6. Composite method

Another way is to generate a composite weight matrix, by dividing the weight matrix into different parts and using the most appropriate estimation approach for each part. For example, for the weights of OD flow–sensor counts, the analytical approach can be used. For the weights of supply parameters–sensor counts, a numerical approach can be used, because the analytical relationship is very hard to derive. For the weights of route choice parameters–sensor counts, an all–one matrix may be sufficient, because the route choice parameters influence the whole network across the entire simulation period and the relationship between these parameters and traffic measurements are hard to capture, even using the numerical approach.

Of course, this method has the drawback that it forces independence between the several sections of the matrix (or, conversely, assumption of full correlation by assuming $w_{ij} = 1$ throughout). The composite method comes naturally in decomposable problems (e.g. demand parameters vs. supply parameters). The first case study below provides a practical application of this method.

3.7. SPSA as a special case

A special case of the weight matrix is the "all–one" matrix, where all the weights are set to 1. In other words, for each parameter, all the measurements are supposed to have the same degree of relevance to this parameter. With this weight matrix, the W–SPSA algorithm becomes the original SPSA algorithm. Therefore, SPSA can be viewed as a special case of W–SPSA. Conversely, W–SPSA is a generalisation of SPSA.

This approach has the advantage of not relying on the estimation of a weight matrix. It is the simplest method to implement and most computationally efficient as no weight matrix estimation or update is required. However, as discussed in previous sections, when the network scale is large, the correlations between parameters and measurements are extremely sparse, using the "all–relevant" assumption results in a highly noisy gradient estimation process, which leads to extremely slow rate of objective function value reduction. Therefore, this approach is only considered to be applied when the weight matrix is very hard to obtain or at the final stage of the calibration process when the goodness–of–fit has already been improved significantly through other methods.

4. Case Study I: Mesoscopic DTA

4.1. Model and parameters

In this case study, we apply W–SPSA to DynaMIT ("Dynamic network assignment for the Management of Information to Travelers", Ben-Akiva et al. (2010a)) in the real–world setting of the Singapore expressway network. DynaMIT is a simulation–based DTA system with mesoscopic demand and supply simulators, each comprising microscopic and macroscopic sub–models. The demand simulator is capable of accurately modeling the time–dependent OD flows, and using detailed microscopic behavioural models for capturing pre–trip decisions and en–route choices of individual drivers given real time information. The supply models simulate the traffic dynamics, the formation, spill–back and dissipation of queues and the influence of incidents. The sophisticated interactions between demand and supply allow the system to estimate and predict network traffic conditions in a realistic manner (Ben-Akiva et al., 2010a).

On the demand side, a route choice model and a set of time-dependent OD flows need to be calibrated. DynaMIT models route choice through a Path-size Logit model (Ramming, 2001), which in our case will demand only one parameter, travel time, to be calibrated (others will remain with the initial values, obtained from field data). On the other side, the total number of time-dependent OD flows to calibrate equals the number of OD pairs in the network times the number of time intervals in the simulation. Reliable historical OD flows as initial values of the estimated OD flows are crucial for the success of calibration. However, in this case study, such OD flows are not available. The initial values are decided arbitrarily yet based on local transportation engineers' intuition.

The supply is based on two models, each one relying on its set of parameters: the moving part in a segment, which is modeled by a speed–density function; and the queuing model at the end of a segment, which is decided by segment capacities. The initial values for the former are obtained by fitting field data while for the latter, we rely on the *Highway Capacity Manual*, slightly adjusted according to local field data.

For more details about the features, framework and implementation of DynaMIT, we redirect the reader to Ben-Akiva et al. (2010a).

4.2. Network and data collection

The entire road network in Singapore is shown in Figure 2. In this case study, the entire expressway system is extracted and modeled in DynaMIT, including expressway links and on–ramps and off–ramps that connect the expressway system to local roads (shown in bold in Figure 2).

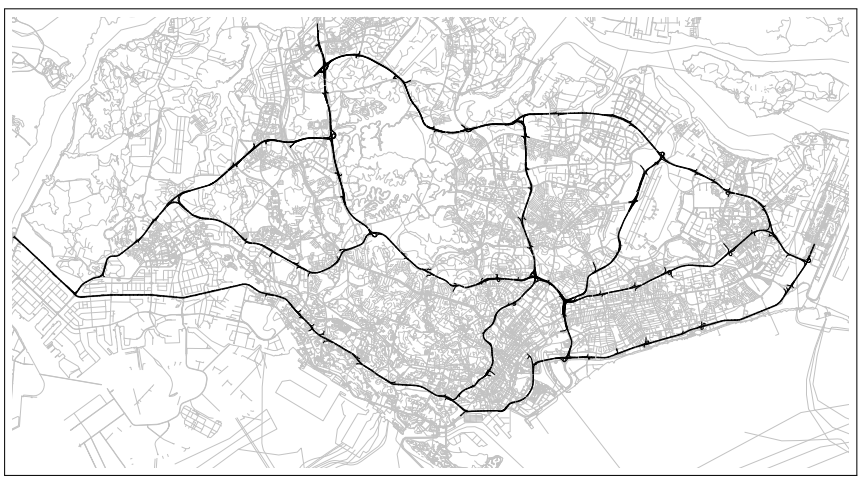

Fig. 2: Singapore expressways network

The network has accurate and detailed representation of length, geometry and lanes of each segment. There are in total 831 nodes connected by 1040 links in the network, each link made up of several segments based on the geometry, making a total of 3388 segments. 4106 OD pairs are chosen among the 831×830 possible node combinations to make sure each origin is an on–ramp, where vehicles enter the expressway system from local roads, and each destination is an off–ramp, where vehicles depart the network. Other heuristic rules are used to eliminate unreasonable long or detour trips between on-ramps and off–ramps. The simulation time period is from 5am to 10am, or 60 intervals with the length of each interval being 5 minute. The first two hours (24 intervals) are used for network warm–up.

The scale of the calibration problem is therefore:

- 1 route choice model parameter (travel time)
- $4,106 \times 36 = 147,816$ time dependent OD flows
- $3,388 \times 6 = 20,328$ speed–density function parameters
- 3,388 segment capacities

It is noted that this is a very different application from the one presented by Lu et al. (2015). First, Lu et al. (2015) considered 15–minute intervals, while in this research we consider 5–minute intervals (shorter intervals result in more volatile traffic measurements and hence a more difficult calibration problem). Furthermore, Lu et al. (2015) only used W–SPSA to calibrate the demand (albeit for an entire day, i.e. a larger number of parameters), while in this research we calibrate all demand and supply parameters simultaneously.

The Land Transport Authority of Singapore (LTA) provides a real time speed and flow data feed for DynaMIT, sent every 5 minutes. Flow data is provided from EMAS system (Expressway Monitoring and Advisory System). The flow data is obtained from about 338 fixed cameras which are mounted on street lamps at distances of approximately 500 to 1000 meters. Speed data is provided for each segment and is derived from probe vehicles equipped with GPS. The exact method for estimating the speed is proprietary and the details are not known to us.

Understandably, the quality of data influences the final achievable accuracy of the calibration process. Inconsistent data will make it harder to fit. More importantly, fitting to inaccurate, or even erroneous data is meaningless in terms of applying the calibrated model to estimate and predict real world traffic conditions (Wei, 2010). As in any other setting, there are factors that affect data quality (e.g. weather and lighting conditions, sensor de-calibration, malfunctioning parts) so we investigated the consistency of flow and speed throughout the dataset and eliminated sensors that demonstrated not to be fully trustworthy through time (e.g. consecutive impossible values), reducing our dataset to 216 sensors overall. For a detailed account of this process, we redirect the reader to Lu (2014). There were, however, *acceptable* inconsistencies, namely due to traffic incidents. In such cases, the flow/speed inconsistency had a temporal mark and the sensor was not removed. However, it does create difficulties for the calibration process so to reduce the influence of incidents and unrecognized malfunctioning sensors, the field traffic data was averaged across the 31 days in August 2011. The averaged data is not "true" data for any specific day, instead it reflects the averaged trends and patterns of the network, which is exactly what the off-line calibration aims to capture.

4.3. Weight matrix calculation

In this case study, the weight matrix is created with a *composite method*, where we combine one simulation with two heuristic approaches. The weights between OD flows and sensor counts, segment speeds are approximated based on network topology, latest estimated route choice model and time–dependent link travel times. Given an OD pair and a time interval, among all the vehicles that leave the origin within this interval, the proportion of these vehicles that pass a specific segment / sensor is calculated using the route choice model of DynaMIT and the travel times in this time period. For longer trips the travel times in the next few time periods are also considered. These proportions are used as weights between this OD flow and sensor counts/speeds from the sensors/segments along the paths between the OD pair. At each iteration of the calibration process, the weights are updated using the latest estimated route choice model and time–dependent link travel times estimated by DynaMIT.

The weights between speed–density parameters, segment capacities and sensor counts, segment speeds are set through a simple heuristic: 1 if the speed–density parameters and segment capacities are at the same segment as the count and speed data; 0 for all other situations. This is an extremely simplified way to capture the relationships. The supply parameters of a specific segment may also have significant enough influence on other neighboring segments.

Finally, the weights between the route choice model parameter and all the measurements are set to 1, which we consider a general heuristic (of even distribution of correlations).

4.4. Results

The results now presented are based on implementing all the technical considerations in the previous section. Fit–to–counts/speeds is calculated as the RMSN between the simulated counts/speeds and observed counts/speeds over all sensors and intervals. The overall fit–to–counts measurement was improved by 49.3% (from 0.426 to 0.216) and the fit–to–speeds was improved by 49.0% (from 0.321 to 0.164). The fit–to–counts and fit–to–speeds by time interval are presented in Figure 3.

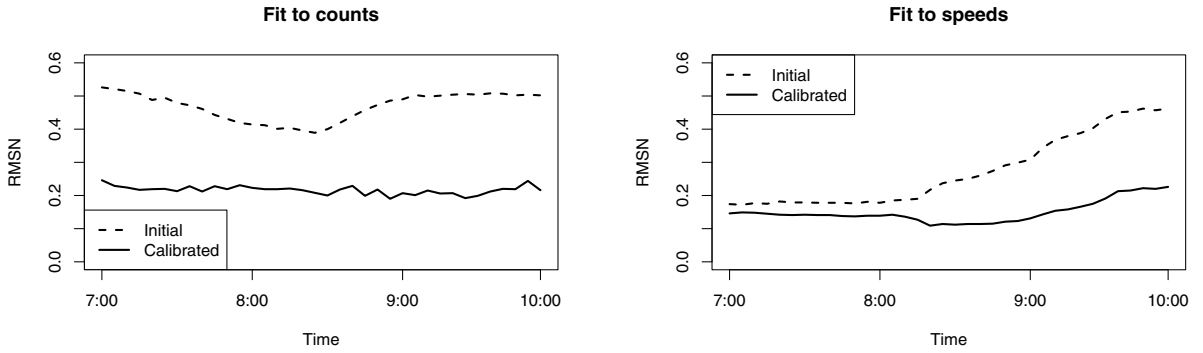

Fig. 3: Fit–to–counts and fit–to–speeds by time interval

Figure 4 shows the fit–to–counts at two different time intervals: 7:30am–7:35am and 8:30am–8:35am. The x–axis corresponds to the observed sensor counts in vehicles per 5 minutes. The y–axis corresponds to the simulated sensor counts in vehicles per 5 minutes after off–line calibration. Blue dots represent observed and simulated counts at a specific sensor. The red 45 degree line represents the perfect fit where the simulated counts exactly matches the observed counts. Most of the dots are very close to the 45 degree line, which indicates good fit. The RMSN between the simulated and observed counts is 0.213 for the left subfigure and 0.200 for the right subfigure.

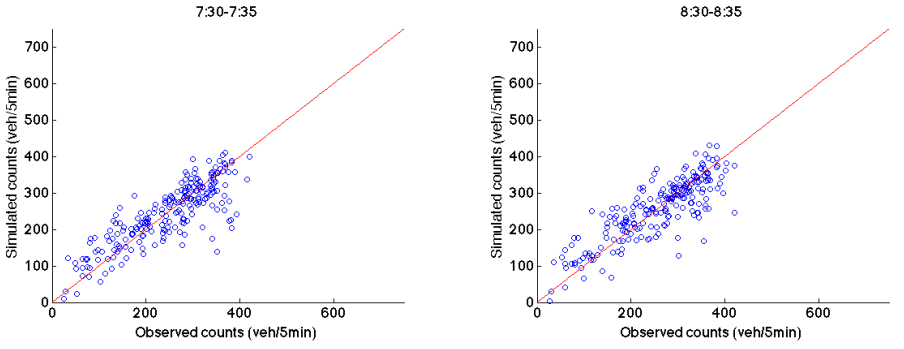

Fig. 4: Fit-to-counts at different intervals

5. Case Study II: Microscopic traffic simulator

In this case–study the calibration of a microscopic traffic simulator for a large number of specific traffic scenarios is tested using the W–SPSA algorithm. The ultimate aim is to generate a large set of detailed traffic variables using a calibrated simulator for scenarios with and without accidents.

5.1. Model and parameters

Here, the focus is on the calibration of a specific microscopic traffic simulator, MITSIMLab (Yang and Koutsopoulos, 1996), to replicate detailed variables for a large set of scenarios. MITSIMLab is a microscopic traffic simulation application developed to evaluate Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS) at the planing and operational level. Travel demand is input in the form of time-dependent OD flows, from which individual vehicles wishing to enter the network are generated. A probabilistic model is used to capture drivers route choice decisions and driving behaviour parameters and vehicle characteristics are randomly assigned to each driver-vehicle unit. MITSIMLab moves vehicles according to route choice, acceleration and lane changing models. The acceleration model captures drivers' response to neighbouring conditions as a function of surrounding vehicles motion parameters. The lane changing model integrates mandatory and discretionary lane–changes in a single model. Merging, drivers' responses to traffic signals, speed limits and incidents are also modelled in detail. The driving behaviour used for this case-study is the one proposed in Toledo et al. (2007), comprising a total of 101 different driving behaviour parameters grouped in 15 different behavioural models (Ciuffo and Lima Azevedo, 2014). More details on the formulation and implementation of MITSIMLab can be found in Yang and Koutsopoulos (1996) and Toledo et al. (2007)

The scenario–specific calibration here presented was preceded by a comprehensive sensitivity analysis to identify the most sensitive parameters to the available measurements (Ciuffo and Lima Azevedo, 2014). This task was carried out to reduce the number of supply parameters to consider during the calibration. The 11 most sensitive parameters out of the 101 driving behaviour model of MITSIMLab were therefore selected for the current experiment, comprising parameters from four different driving behaviour models: the reaction time model, lane–choice model, car–following model and driver heterogeneity model. Finally, the 101 parameters had been previously calibrated with trajectories collected during a specific day on the A44 motorway (Lima Azevedo, 2014). These calibrated values were used as initial parameters (and historical values) during the scenario–specific calibration.

Regarding the demand parameters, the total number of time dependent OD flows equals the number of OD pairs in the network times the number of simulated periods. For this specific case-study a seed OD was available, with time-intervals of 30 mins. As no significant intra-variability regarding the available measurements was found in almost all intervals, the same 30 min interval was considered for the dynamic OD matrix.

5.2. Network and data collection

The simulation case–study is an urban motorway (A44) near Porto, Portugal. This motorway was selected due to several safety related issues: dense traffic, unusually high number of lane changes due to frequent route–choice decision making, short spacing between interchanges and high percentage of heavy goods vehicles. A44 is a 3,940m long dual carriageway urban motorway with 5 major interchanges, two 3.50m wide lanes and 2.00m wide shoulders in each direction Acceleration and deceleration lanes are added to the main carriageway section at all interchanges, although often as short as 150m. On and off–ramps connect to local roads, which generally have tight horizontal curves, intersections or pedestrian crossings, features that tend to impose significant reductions in vehicle speeds.

The A44 is equipped with an automatic traffic counting station on each stretch, located at kilometers 3.700, 2.400, 1.750 and 0.050. The eight (four per traffic direction) loop sensors are able to count, classify and measure vehicle speeds in real time. The road concessionaire's data center keeps record of these outputs in a simplified data format and aggregated by periods of five minutes. As mentioned previously, a seed OD with a total of 100 OD pairs was available as initial demand parameters values. This seed OD was estimated using license plate recognition at the main entry and exit points of each interchange of the A44 (see Figure 5). As we focus on scenario–specific calibration the 30 min periods before and after a specific event (accident and no-accident) were considered, along with an initial

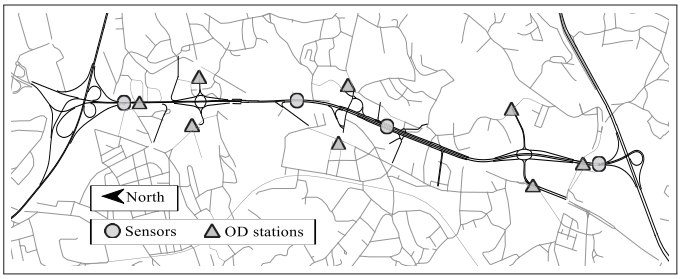

Fig. 5: A44 network

30 mins simulation warm–up.The sensor data was filtered using the method proposed by Chen et al. (2003). Sensor measurements marked as erroneous were discarded from the calibration.The scale of the calibration problem is:

- 11 driving behaviour parameters
- 200 time dependent OD flows
- 192 (maximum) sensor count measurements
- 192 (maximum) average speed measurements

Although the size of such a single calibration experiment is much smaller than the one presented in the previous case-study, the benefit of using a more efficient calibration algorithm, such as W–SPSA, is clear when the same experiment design needs to be carried out a large number of times, as shown later.

5.3. Weight matrix estimation

For the estimation of the weight matrix a *composite method* was used, where the demand parameter weights were estimated using simulation-based approximation and the driving behaviour parameter weights by heuristic approximation.

Similarly to the previous case–study, the weights between OD flows and sensor counts and speeds are approximated to the proportion of vehicles from a specific OD that are detected by the sensor at stake. Equation (28) was then used to estimate weights:

$$w_{ij}^{speeds} = w_{ij}^{counts} = \frac{\sum_{r=1}^{R} \frac{\Delta M_{rj}^{counts}}{\Delta \theta_i}}{R} \tag{28}$$

where w_{ij} is the weight regarding the OD path i and the speed or count measurement M_{rj}^{counts} at sensor j for replication r. The number of replications R should be such that stability in the values of the weights is achieved. In this case, the number of replications was empirically set equal to 3. Larger numbers of replications can also be tested. While this would proportionately increase the computational burden, it is noted that the simulation method is fully parallelizable, i.e. given sufficient computational resources, all runs can be performed in parallel.

Driving behaviour parameter weights were set to 1 as no distinction was made between individual effect on different loop sensor output. In a more complex approach, sensitivity analysis may be used to compute different driving behaviour parameter weights as distinct driving behaviour sub-model parameters may affect each sensor differently (e.g.: merging behaviour near ramps and weaving sections).

Regarding the specification of the objective function, the OLS assumption was assumed and the variance-covariance matrices $\mathbf{\Omega_M}$ and $\mathbf{\Omega_\theta}$ are specified as a block diagonal matrix with $\mathbf{\Omega_{M_k}}$ and $\mathbf{\Omega_\beta}$ as its diagonal elements, respectively. The assumed fixed weights of the optimizing function are $\Omega_{M_k^{counts}} = 0.3$, $\Omega_{M_k^{speeds}} = 0.5$ and $\Omega_\beta = 0.2$. These values were defined previously, based on the contribution of each information on the calibration process. As we focus on detailed traffic statistics a higher contribution was given to speed related data. A sensitivity analysis on these weight values may, however, enhance the calibration final results.

Finally, the constant parameters of the W–SPSA algorithm (A, a, α, γ and c) were set to previously estimated values for a generic SPSA application to MITSIMLab calibration (Vaze et al., 2009).

5.4. Individual–case calibration results

As example of this specific W–SPSA application, the results for the calibration for a specific rear-end accident that occurred at 8:30 on the 27^{th} of February 2007, at the km 3.300 in the South-North direction are here presented.

The number of iterations used in SPSA is typically large, but in the W–SPSA framework much fewer iterations are required to reach satisfactory values (Lu et al., 2015). As we aim at reducing this number as much as possible, the stopping criteria was a threshold of relative improvement between consecutive iterations of 5% in both count and speeds RMSN.

After just 30 iterations, the W–SPSA converged and the RMSN improved by 80.1% for speed observations, reaching the value of 0.19, and by 77% for counts, with a final value of 0.22 (see Figure 6). As a result of the goodness–of–fit, the W–SPSA method quickly converged to the individual loop-based measurements (see Figure 7 a. and b.). In Figure 7 c. and d., the final calibrated demand parameters (OD pairs) and driving behaviour parameters are plotted against their initial/historical values: the seed OD and the trajectory-based parameter values.

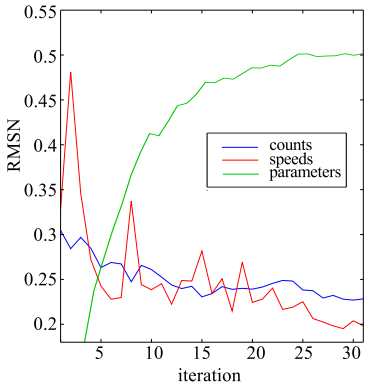

Fig. 6: W–SPSA test performance for a specific scenario calibration

5.5. Multiple scenario calibration results

In this section, the above W–SPSA configuration was tested in the calibration of a large set of different traffic events occurred in the A44 motorway. Here, we focused our attention in the replication of detailed traffic variables (artificial trajectories) in accident scenarios, and test whether its values would differ much from the non-accident ones. If such variability is replicated by the simulation model its use in specific safety studies can be considered.

Sensor and accident data was collected for the period of 2007 to 2009. During the three years in analysis, 173 accidents were recorded. As both the sensor and the accident data has a resolution of 5 min, this value was used as temporal unit of an event itself. Similarly, the nature of the accident location record required a spatial observation unit of 50m. With the spatial and temporal units, 710 segments of 50 m and 257,184 time periods of 5 min (excluding the periods erroneous sensor data) were obtained, resulting in a total of more than 180×10^6 events to be considered for simulation. As expected, most of these events are non-accident events. Due to computational limitations, a random sampling technique was necessary and a sampling rate of 3.5×10^{-5} was selected for non-accident events with non-erroneous sensor data (6,400), such that the simulation time to generate the needed simulation output (vehicle trajectories) and the computer memory needed to store them during the final scenario comparison analysis would remained tractable. The outputs of lower sub-samples were also analysed to confirm the low sample size bias. All available accident events with good sensor data were considered (144). This configuration resulted in a total of 6,544 events to be simulated in MITSIMLab.

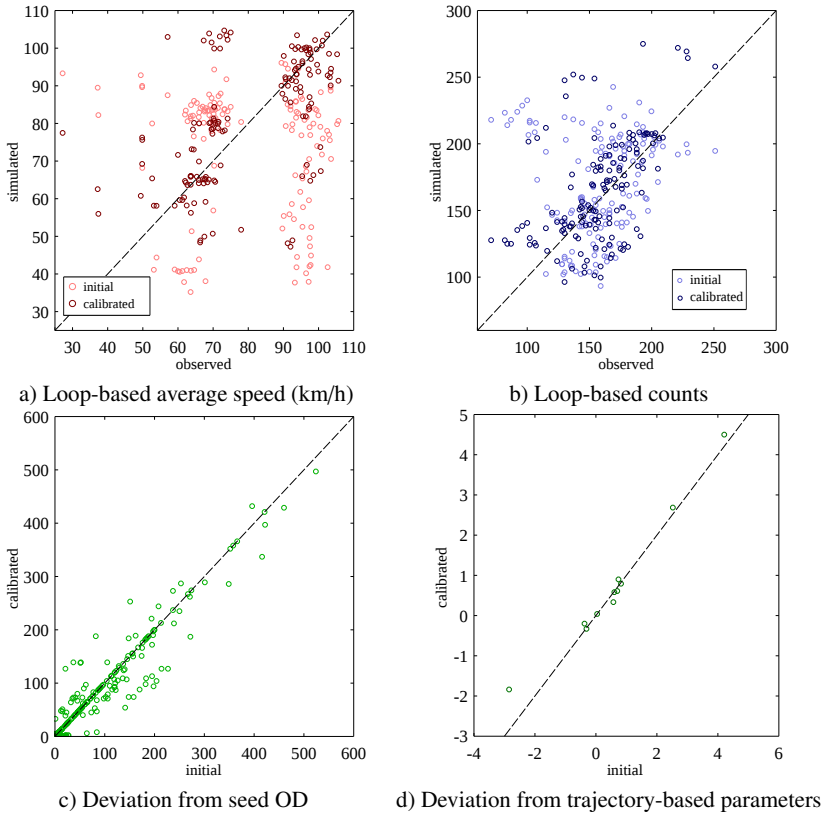

Fig. 7: W–SPSA test performance for a specific scenario calibration

Finally, for this multiple calibration scenario, the W–SPSA algorithm was implemented in MATLAB and ran (along with MITSIMLab) under Scientific Linux in a cluster with 80 cores with 1GB of RAM memory each. This computational resource was essential for the distribution of the several calibration tasks carried out.

In Figure 8, the distribution of the performance of the W–SPSA calibration is presented in terms of RMSN reduction for counts and speeds. Using just 30 iterations of the W–SPSA algorithm, the reductions rates are quite satisfactory; yet, for a non-negligible fraction of the scenarios, the reduction rates remained bellow 10%. These low performances mainly affected scenarios where the starting value of the objective function was already low. Further iterations in a dedicated processing would be necessary to improve these calibrations.

The final comparison between the different outputs from calibrated simulation of accident and no-accident scenarios allowed the identification of several differences, both at the calibrated driving behaviour parameters level and at the detailed calibrated simulation outputs, such as deceleration rates or headways (see Lima Azevedo, 2014, for the detailed analysis).

6. Conclusion

In this paper, W–SPSA is implemented, modified, and applied in two real–world case studies to validate its performance and scalability. While it was demonstrated earlier Lu et al. (2015) that W–SPSA can be far more efficient than SPSA, there were relevant challenges to tackle, namely on the generation of the weight matrix $\mathbf{W}$, which is the key innovation of W–SPSA. In this paper, we present a series of alternative ways to formally generate the weight matrix, as well as heuristic and combined approaches that can extend its applicability.

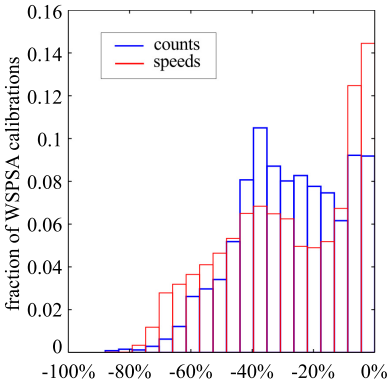

Fig. 8: Distribution of the RMSN reduction rates during the scenarios calibration

In the first case study, demand and supply input and parameters of DynaMIT, a mesoscopic DTA system, are jointly calibrated for the entire expressway network of Singapore. The field data, especially the sensor counts data are analyzed and processed before the calibration process. In the second case study, demand and supply parameters of MITSIMLab, a microscopic traffic simulator, are jointly calibrated for a motorway network in Portugal. The calibration is then repeated several thousand times with different inputs, within the context of a demanding road–safety analysis.

W–SPSA is a general and powerful approach. When applying W–SPSA to different setups, specific requirements of each case study call for several empirical considerations and extension. For example, as a result of the very high number of degrees of freedom in large–scale calibration problems, it is often possible for the calibration process to stray away of reasonable parameter values. In order to prevent the drift of parameter values too far away from the initial (reasonable) range, it is recommended to apply boundaries during the calibration process. In the case of unbounded optimisation approaches, such as those used in this research, it is up to the researcher to ensure that the values in each iteration do not lie outside the pre–determined reasonable bounds and –if they do– to take measures to resolve the issue.

The performance of W–SPSA can also suffer when the scale of the calibration inputs and parameters are different. Balakrishna (2006) suggests a ratio perturbation method, when applying SPSA and argues that, because of the significant difference in magnitude of different parameter values, perturbing them with a same value is not applicable. For example, most of the OD flows are between 0 to a few hundreds, while the capacity parameter is a value usually between 0 and 10 (vehicle per second). Multiple perturbation step size magnitudes have to be decided for all the different parameters. A more convenient way is to perturb all the parameters with a same step size magnitude but in a ratio way. As a generalisation of SPSA, W–SPSA can also benefit from this approach. In the first case study, the original non-ratio perturbation was applied for OD flows and the ratio perturbation was applied for other parameters.

In the SPSA algorithm, within each iteration, after determining a perturbation step size, the perturbation directions are generated randomly based on a selected distribution (Bernoulli in many cases). It was found in experimentations within the first case study that, when using W–SPSA, at the beginning stage of the calibration process, if the start parameter values are believed to be biased towards a specific direction, a fixed perturbation leads to much faster improvement of the objective function value. More specifically, the initial OD flows were found to be mostly smaller than true values (according to the simulated counts). Therefore, in each iteration, instead of perturbing all the OD flows based on a randomly generated vector (i.e., some + some - then some - some +), all the OD flows are perturbed in a same direction (i.e., all + then all -). This fixed perturbation was applied at the very beginning stage and only for OD flows. It dramatically accelerated the improvement of fit–to–counts. After a few iterations, random perturbation was applied for all parameters. It is noted that this finding is based on empirical experiments with the expressway network and dataset in the first case study; no theoretical proof was done to prove its generality, something that is an interesting direction for future research.

When applying W–SPSA with a very large number of calibration parameters, it is possible to have unstable convergence if all parameters are perturbed simultaneously. One practical approach, that seemed to work well in the first

case study, is to randomly perturb a subset of the calibration parameters in each iteration. This method may lead to a worse convergence rate, but it is unbiased.

When applied properly, using these empirical recommendations as applicable, W–SPSA appears to outperform SPSA significantly and achieves great improvement over the reference case. Results show that after the off-line calibration, the traffic simulation systems are able to reproduce the observed traffic condition with a high level of accuracy. In this research, we have focused on the off–line calibration only. However, many traffic simulation applications (arguably the most interesting and challenging, e.g. those that involve DynaMIT (Ben-Akiva et al., 2010a)) can benefit from online calibration (Antoniou et al., 2007a). W–SPSA can also benefit such applications, e.g. within the framework suggested by Antoniou et al. (2007b).

Acknowledgements

This research was supported by the National Research Foundation Singapore through the Singapore MIT Alliance for Research and Technology's FM IRG research programme. The authors gratefully acknowledge the Land Transport Authority of Singapore for providing one of the datasets and the Portuguese National Grid Initiative for the computational support in one of the case-studies (https://wiki.ncg.ingrid.pt).

References

Antoniou, C., Barcelo, J., Brackstone, M., Celikoglu, H., Ciuffo, B., Punzo, V., Sykes, P., Toledo, T., Vortisch, P., Wagner, P., 2014. Traffic Simulation: Case for guidelines. JRC Scientific and Technical Research Report. Publications Office of the European Union, Luxembourg. ISBN 978-92-79-35578-3.

Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2007a. Nonlinear kalman filtering algorithms for online calibration of dynamic traffic assignment models. IEEE Transactions on Intelligent Transportation Systems 8, 661–670.

Antoniou, C., Koutsopoulos, H.N., Yannis, G., 2007b. An efficient non-linear kalman filtering algorithm using simultaneous perturbation and applications in traffic estimation and prediction, in: Procs. of the IEEE Intelligent Transportation Systems Conf., Seattle, USA. pp. 217–222.

Ashok, K., Ben-Akiva, M.E., 2000. Alternative approaches for real–time estimation and prediction of time–dependent origin–destination flows. Transportation Science 34, 21–36.

Ashok, K., Ben-Akiva, M.E., 2002. Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. Transportation Science 36, 184–198.

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

Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., Wen, Y., 2007. Calibration of microscopic traffic simulation models: Methods and application. Transportation Research Record: Journal of the Transportation Research Board 1999, 198–207.

Balakrishna, R., Koutsopoulos, H.N., 2008. Incorporating within-day transitions in the simultaneous offline estimation of dynamic origindestination flows without assignment matrices. Transportation Research Record: Journal of the Transportation Research Board , 31–38.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H.N., Mishalani, R., 2001. Network state estimation and prediction for real-time traffic management. Networks and Spatial Economics 1, 293–318.

Ben-Akiva, M., Koutsopoulos, H.N., Antoniou, C., Balakrishna, R., 2010a. Traffic simulation with dynamit, in: Barceló, J. (Ed.), Fundamentals of Traffic Simulation, Springer, New York, NY, USA. pp. 363–398.

Ben-Akiva, M., Koutsopoulos, H.N., Toledo, T., Yang, Q., Choudhury, C.F., Antoniou, C., Balakrishna, R., 2010b. Traffic simulation with mitsimlab, in: Barceló, J. (Ed.), Fundamentals of Traffic Simulation, Springer, New York, NY, USA. pp. 233–268.

Cantelmo, G., Cipriani, E., Gemma, A., Nigro, M., 2014. An adaptive bi–level gradient procedure for the estimation of dynamic traffic demand. IEEE Transactions in Intelligent Transportation Systems 15, 1348–1361.

Chen, C., Kwon, J., Rice, J., Skabardonis, A., Varaiya, P., 2003. Detecting errors and imputing missing data for single-loop surveillance systems. Transportation Research Record 1855, 160167.

Cipriani, E., Florian, M., Mahut, M., Nigro, M., 2011. A gradient approximation approach for adjusting temporal origin-destination matrices. Transportation Research Part C: Emerging Technologies 19, 270–282.

Ciuffo, B., Lima Azevedo, C., 2014. A sensitivity-analysis-based approach for the calibration of traffic simulation models. IEEE Transactions on Intelligent Transportation Systems 15, 1289–1309.

Frederix, R., Viti, F., Corthout, R., Tampre, C.M., 2011. New gradient approximation method for dynamic origin-destination matrix estimation on congested networks. Transportation Research Record: Journal of the Transportation Research Board , 19–25.

Frederix, R., Viti, F., Tampre, C.M., 2013. Dynamic origin–destination estimation in congested networks: theoretical findings and implications in practice. Transportmetrica A: Transport Science 9, 494–513.

Huang, Z., Ma, X., Koutsopoulos, H.N., 2010. A numerical optimization approach for calibration of dynamic emission models based on aggregate estimation of ARTEMIS, in: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, IEEE. pp. 1221–1226.

Lee, J.B., Ozbay, K., 2008. Calibration of a macroscopic traffic simulation model using enhanced simultaneous perturbation stochastic approximation methodology, in: Transportation Research Board 87th Annual Meeting.

Lima Azevedo, C., 2014. Probabilistic Safety Abalysis using Traffic Microscopic Simulation. Ph.D. thesis. Instituto Superior Técnico, Portugal.

Lu, L., 2014. W-SPSA: An Efficient Stochastic Approximation Algorithm for the Off-line Calibration of Dynamic Traffic Assignment Models. Master's thesis. Massachusetts Institute of Technology.

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.

Ma, J., Dong, H., Zhang, H.M., 2007. Calibration of microsimulation with heuristic optimization methods. Transportation Research Record: Journal of the Transportation Research Board 1999, 208–217.

Paz, A., Molano, V., Gaviria, C., 2012. Calibration of corsim models considering all model parameters simultaneously, in: Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, IEEE. pp. 1417–1422.

Ramming, M.S., 2001. Network knowledge and route choice. Ph.D. thesis. Massachusetts Institute of Technology.

Spall, J.C., 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. on Automatic Control 37, 332–341.

Spall, J.C., 1998. An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Tec. Digest 19, 482–492.

Toledo, T., Koutsopoulos, H., Ben-Akiva, M.E., 2007. Integrated driving behavior modeling. Transportation Research Part C: Emerging Technologies 15, 96112.

Vaze, V., Antoniou, C., Wen, Y., Ben-Akiva, M., 2009. Calibration of dynamic traffic assignment models with point-to-point traffic surveillance. Transportation Research Record: Journal of the Transportation Research Board 2090, 1–9.

Wei, Z., 2010. Critical enhancements of a dynamic traffic assignment model for highly congested, complex urban network. Master's thesis. Massachusetts Institute of Technology.

Yang, Q., Koutsopoulos, H.N., 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transportation Research Part C: Emerging Technologies 4, 113–129.