High Dimensional Origin Destination Calibration Using Metamodel Assisted Simultaneous Perturbation Stochastic Approximation

Mun Chon Ho, Joanne Mun-Yee Lim, Chun Yong Chong, *Member, IEEE*, Kah Keong Chua, and Alvin Kuok Lim Siah

***Abstract*— The huge traffic data generated by intelligent transportation system (ITS) leads to the development of many advanced traffic models. These traffic models consist of many adjustable parameters which need to be calibrated before they are used in practice. This paper focuses on the offline calibration of origin-destination (OD) input parameters of a simulation-based traffic model to match its output with sensor data. An improved Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm called Metamodel Assisted Simultaneous Perturbation Stochastic Approximation (MSPSA) is proposed in this paper to calibrate high-dimensional OD parameters within a tight computational budget. The proposed MSPSA combines the gradient of SPSA with the gradient of a differentiable metamodel function to improve the calibration efficiency. An integer program is also used to fine-tune the OD estimates. The proposed MSPSA algorithm is tested on a simple synthetic toy network and complex road network of Kuala Lumpur (KL), Malaysia. The proposed MSPSA algorithm is compared against SPSA and state-of-the-art Weighted Simultaneous Perturbation Stochastic Approximation (WSPSA) in both transportation networks. For KL network, synthetic and real-world sensor measurements are used as ground truth references to evaluate the performance of each approach. Based on the simulation results, the proposed MSPSA algorithm is able to gain at least 50% of improvement as compared to SPSA and WSPSA in both synthetic and real-world scenarios.**

***Index Terms*— Origin-destination calibration, stochastic approximation, simulation-based optimization, metamodel, SPSA.**

I. Introduction

WITH the advancement of ITS, high-resolution traffic and mobility data becomes readily available worldwide. This facilitates the development of traffic models whose purpose is to assist practitioners in the design and operations of the urban mobility network. Due to the complex dynamics of transportation networks, simulation-based traffic models are usually used to estimate the traffic state of transportation networks. Practitioners can use these simulation-based traffic models to extrapolate traffic conditions and carry out what-if analyses on transportation networks. However, it is very important to ensure that these high-resolution traffic models are well calibrated with field data collected from ITS before they are used on real traffic networks. With an enormous amount of data generated by ITS, it is possible to calibrate the traffic simulation model to improve its reliability. The main inputs of most traffic simulation models are the traffic demand parameters, also known as OD parameters. Each OD parameter represents the number of vehicles which travel from one origin to one destination. Since it is impractical to directly observe the origin and destination of each vehicle in a real transportation network, the OD parameters need to be estimated from other data sources. According to Osorio [1], the problem can be divided into two classes. The first one is the OD calibration problem, where the main objective is to calibrate the OD demand inputs of a specific traffic model by matching the model outputs with sensor measurements. The calibrated OD demand is not intended to predict the actual OD demand directly. The second is the OD estimation problem, where the main objective is to obtain the actual OD demand directly. This paper focuses on the first class of the problem. Specifically, this paper utilizes vehicle counts information collected from traffic cameras and induction loops installed on roads to calibrate the OD demand.

This paper focuses on the offline calibration of OD parameters for simulation-based traffic models. During the calibration process, the OD parameters are estimated and evaluated iteratively until the simulation output matches closely with the field data. However, OD calibration is a challenging problem because it is often high-dimensional. The size of the solution space increases quadratically with the number of origins and destinations. Experimenting with all possible values of parameters is usually impractical since it is intractable to run simulations for all possible scenarios. Moreover, OD calibration is an underdetermined problem because the number of OD parameters is usually higher than the number of observations (e.g. vehicle counts information obtained from traffic sensors). The underdetermined nature of the OD calibration problem was already recognized in the early days by many researchers [2], [3], [4]. The situation becomes worse when traffic dynamics are considered (e.g. multiple time intervals for

Manuscript received 28 September 2021; revised 28 April 2022 and 28 October 2022; accepted 29 December 2022. Date of publication 12 January 2023; date of current version 29 March 2023. This work was supported in part by Recogine Technology Sendirian (SDN) Berhad (BHD) [31]; and in part by the School of Engineering, Monash University Malaysia. The Associate Editor for this article was N. Bekiaris-Liberis. *(Corresponding author: Joanne Mun-Yee Lim.)*

Mun Chon Ho and Joanne Mun-Yee Lim are with the School of Engineering, Monash University Malaysia, Subang Jaya 47500, Malaysia (e-mail: Joanne.Lim@monash.edu).

Chun Yong Chong is with the School of Information Technology, Monash University Malaysia, Subang Jaya 47500, Malaysia.

Kah Keong Chua and Alvin Kuok Lim Siah are with Recogine Technology SDN BHD, Subang Jaya 47650, Malaysia.

Digital Object Identifier 10.1109/TITS.2023.3234615

each OD). Hence, the OD parameters need to be determined systematically.

To alleviate the aforementioned problem, a new OD calibration algorithm called MSPSA is proposed in this paper. The contributions of this paper are summarized as follows:

- The MSPSA algorithm integrates a metamodel into SPSA algorithm. The metamodel provides domain knowledge about the problem to guide the solution estimate of SPSA. This allows MSPSA algorithm to be used in high-dimensional problems and large-scale networks.
- Simulation results in synthetic traffic scenarios show that the proposed MSPSA achieved the lowest root mean squared normalized error (RMSNE) as compared to the basic SPSA [5] algorithm and the state-of-the-art WSPSA [6] algorithm under tight computational budget. Therefore, the proposed MSPSA allows practitioners to spend less computational resources to obtain a good quality solution.
- The performance of the proposed MSPSA is also evaluated using a real-world traffic scenario in the KL network, where the quality of the initial estimate is not guaranteed. Based on the simulation results, the proposed MSPSA is less sensitive to the quality of the initial estimate compared to other benchmark methods, proving its effectiveness and real-world applicability.

The rest of this paper is organized as follows. Section II reviews the recent works related to OD calibration. Section III formulates the OD calibration problem. Section IV details the proposed MSPSA algorithm. Section V describes the simulation setup. Section VI compares the calibration performance between the proposed MSPSA and other methods in three different traffic scenarios. Finally, concluding remarks and potential future work are presented in Section VII.

II. Related Works

Direct search method which does not require gradient information of the objective function is one of the approaches used for parameter calibrations. General-purpose algorithms such as genetic algorithm (GA) [7] and differential evolution (DE) [8] are some of the commonly used direct search approaches. These direct search methods are popularly used because of their simplicity and can take in any type of traffic observation to formulate the objective function. The generality of direct search methods comes with a shortcoming, which is the lack of computational efficiency. These algorithms usually require a very high number of iterations before obtaining a reasonably good solution [5], [8]. For a large-scale transportation network, each simulation may take from minutes to hours to evaluate. In practice, traffic model calibration is usually done within a tight computational budget. Extending these general-purpose algorithms to improve their efficiency is an active area of research.

In the context of OD calibration, SPSA [5] is extensively used [9], [10], [11]. SPSA algorithm comes from a family of stochastic approximation algorithms. SPSA works by perturbing the decision variables to estimate the gradient of the objective function. However, as the problem scale increases in relation to the network size and the number of time intervals, the gradient approximation error of SPSA increases as well [6]. Thus, Lu et al. [6], Antoniou et al. [12] introduced a weight matrix which contains information on spatial and temporal correlation in a traffic network to improve the convergence and robustness of the SPSA algorithm. A cluster-wise SPSA [13] is proposed to divide OD flows into a small number of clusters to reduce the bias of the gradient estimate. principal component analysis (PCA), which is a dimensionality reduction technique, is introduced by Djukic et al. [14] to predict dynamic OD matrices. Motivated by this, Prakash et al. [15] and Qurashi et al. [16] combined PCA with Kalman filter and SPSA, respectively, to improve the OD calibration performance. However, all PCA-related algorithms require historical OD demand data to identify the principal components. To address this issue, Qurashi et al. [17] proposed a data assimilation framework to generate historical estimates. Although OD demand can be considered as a continuous variable by the SPSA algorithm, it is in fact discrete in the context of agent-based simulation. A discrete version of SPSA is proposed by Wang et al. [18] to deal with discrete optimization problems. Recently, Oh et al. [19] proposed Weighted Discrete Simultaneous Perturbation Stochastic Approximation (WDSPSA) to minimize the error distance between the simulated and observed measurements.

Some researchers use domain knowledge to speed up the OD calibration process. For instance, Toledo et al. [20] proposed using the linear assignment matrix approximation. Recently, a metamodel (or surrogate model), which approximates the input-output relationship of the simulator, has been proposed in existing studies [1], [21], [22] to optimize expensive functions. For instance, the metamodel proposed by Osorio [1] contains information about the transportation network, such as the road congestion function, the route choice function, and the physical connectivity between roads. This information serves to constrain the search space of the optimizer to speed up the calibration process. Their experimental results show that their approach outperforms SPSA and is not sensitive to initial points. Despite this, the metamodel proposed by Osorio [1] requires pre-tuned parameters before it is used for calibration. Ideally, researchers can spend a large amount of time to pre-tune their parameters to suit their case study. But in reality, pre-tuning is an exhaustive approach and highly computationally expensive. Thus, the experimental results presented in their paper [1] might not reflect the actual usage of OD calibration in practice.

Motivated by the simplicity of the SPSA algorithm and the effective reduction in the search space by a metamodel, a hybrid variant of SPSA algorithm, which is named MSPSA, is proposed in this paper.

III. Problem Formulation

For convenience, Table I lists the notations which are frequently used throughout this paper.

In most of the transportation research, the traffic network is modelled as a directed graph $(\mathcal{V}, \mathcal{M})$ with nodes $\mathcal{V}$ and links $\mathcal{M}$. Nodes represent junctions, and links represent roads. Two-way roads are represented as two separate links with opposite

TABLE I
NOTATIONS USED IN THIS PAPER

Notation	Description
$\mathcal{M}$	Set of roads.
$\mathcal{M}'$	Set of roads with sensor measurements.
$\mathcal{P}$	Set of time intervals.
$\mathcal{P}'$	Set of time intervals with sensor measurements.
$\mathcal{Q}$	Set of OD pairs.
o	Origin identifier.
d	Destination identifier.
p_1	Departure time interval identifier.
p_2	Sensor measurement time interval identifier.
m	Road identifier.
x	Vector containing OD parameters.
y	Vector containing simulated vehicle counts.
y''	Vector containing ground truth vehicle counts.
$\hat{y}$	Vector containing vehicle counts estimated by metamodel.
k	Iteration number.
Δ_k	Perturbation vector at iteration k.
x_k	x at iteration k.
$\lvert \bullet \rvert$	Cardinality of a set or absolute value of a scalar.
$\mathcal{L}(x)$	Loss function evaluated at x.
f	Simulation function.

directions for traffic simulation. In this research, a traffic simulator is used to compute vehicle counts y given a vector x of OD parameters. Thus, the OD calibration problem can be formulated in an optimization framework, and it is given as:

$$\min_x \quad \mathcal{L}(x) = \mathcal{L}''(y'', y) + \tilde{\mathcal{L}}(\tilde{x}, x) \tag{1}$$

$$y'' = (y''_{m,p_2} : m \in \mathcal{M}', p_2 \in \mathcal{P}') \tag{2}$$

$$y = (y_{m,p_2} : m \in \mathcal{M}', p_2 \in \mathcal{P}') \tag{3}$$

$$x = (x_{o,d,p_1} : (o,d) \in \mathcal{Q}, p_1 \in \mathcal{P}) \tag{4}$$

$$y = f(x) \tag{5}$$

$\mathcal{L}$ is the loss function to be minimized. $\mathcal{L}''$ is the goodness-of-fit function between y'' and y. $\tilde{\mathcal{L}}$ is the goodness-of-fit function between x and prior OD parameters $\tilde{x}$. $\mathcal{P}$ is the set of time intervals. $\mathcal{Q}$ is the set of OD pairs. In traffic simulation, any road can be selected to become the source (origin) or sink (destination) of network traffic, hence $\mathcal{Q} \in \mathcal{M} \times \mathcal{M}$. $\mathcal{M}' \subseteq \mathcal{M}$ and $\mathcal{P}' \subseteq \mathcal{P}$ are the observed set of roads and the observed set of time intervals, respectively. $y''_{m,p_2} \in \mathbb{N}_0$ is the ground truth traffic counts on road m at time interval p_2. $\mathbb{N}_0$ is a set of natural numbers including zero. $x_{o,d,p_1} \in \mathbb{N}_0$ is the number of vehicles which intend to travel from origin o to destination d at departure time interval p_1. $y_{m,p_2} \in \mathbb{N}_0$ is the number of vehicles passing through road m during time interval p_2 in traffic simulation. f is the simulation function. For simplicity, this paper does not consider $\tilde{\mathcal{L}}$.

IV. METHODOLOGY

A. SPSA

SPSA determines the optimal solution through the iterative update rule shown in Eq. (6).

$$x_{k+1} = x_k - a_k \tilde{g}_k \tag{6}$$

$$x_k = (x_{o,d,p_1,k} : (o,d) \in \mathcal{Q}, p_1 \in \mathcal{P}) \tag{7}$$

$$\tilde{g}_k = \frac{\mathcal{L}(x_k^+) - \mathcal{L}(x_k^-)}{2c_k \Delta_k} \tag{8}$$

$$x_k^{\pm} = x_k \pm c_k \Delta_k \tag{9}$$

$$\Delta_k = (\Delta_{o,d,p_1,k} : (o,d) \in \mathcal{Q}, p_1 \in \mathcal{P}) \tag{10}$$

$$a_k = \frac{a}{(A+k+1)^{\alpha}} \tag{11}$$

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

k is the iteration number. x_k is the estimated OD parameters vector which contains $x_{o,d,p_1,k}$ at iteration k as shown in Eq. (7). $\tilde{g}_k$ is the estimated gradient at x_k. Δ_k is a random perturbation vector containing $\Delta_{o,d,p_1,k}$ which is an independent random variable generated through the Bernoulli process taking values of $+1$ and -1 with equal probabilities. x_k^+ and x_k^- are positively and negatively perturbed solution vectors, respectively. a_k and c_k are positive scalars controlling the step size and perturbation amplitude, respectively. a, c, A, α and γ are the SPSA algorithm parameters tunable by the user. Since the domain of the OD calibration problem is discrete, each OD parameter is rounded to its nearest integer before being evaluated through traffic simulation.

B. WSPSA

WSPSA [6] improves upon SPSA by introducing weight parameters $w_{o,d,p_1,m,p_2,k}$ to reduce search noise. $w_{o,d,p_1,m,p_2,k}$ reflects the degree of correlation between x_{o,d,p_1} and y_{m,p_2} at iteration k. The value of $w_{o,d,p_1,m,p_2,k}$ is 0 if there is no correlation between x_{o,d,p_1} and y_{m,p_2} at iteration k. According to [6], $w_{o,d,p_1,m,p_2,k}$ is estimated using Eq. (13).

$$w_{o,d,p_1,m,p_2,k} = \frac{x_{o,d,p_1,m,p_2,k}}{\sum_{m,p_2} x_{o,d,p_1,m,p_2,k}} \tag{13}$$

$x_{o,d,p_1,m,p_2,k}$ is the number of vehicles with origin o, destination d, departure interval p_1, and able to pass through road m at time interval p_2 in simulation at iteration k. Then, the gradient $g'_{o,d,p_1,k}$ estimated by WSPSA for $x_{o,d,p_1,k}$ is calculated based on the weighted sum of measurement error changes related to parameter $x_{o,d,p_1,k}$ as shown in Eq. (14).

$$g'_{o,d,p_1,k} = \frac{\sum_{m,p_2} E_{m,p_2,k} \times w_{o,d,p_1,m,p_2,k}}{2c_k \Delta_{o,d,p_1,k}} \tag{14}$$

$E_{m,p_2,k}$ is the change in measurement error squared on road m at time interval p_2 due to perturbation at iteration k. $E_{m,p_2,k}$ is calculated using Eq. (15).

$$E_{m,p_2,k} = (y''_{m,p_2} - y^+_{m,p_2,k})^2 - (y''_{m,p_2} - y^-_{m,p_2,k})^2 \tag{15}$$

$y^+_{m,p_2,k}$ and $y^-_{m,p_2,k}$ are y_{m,p_2} simulated at x_k^+ and x_k^-, respectively.

C. MSPSA

In this paper, an improved algorithm called MSPSA is proposed. The random perturbation mechanism in SPSA moves the current solution vector to both negative and positive directions, even if all elements of the solution vector are expected to move in the same direction to reduce loss value. Therefore, if the optimal solution is far away from the initial estimate, relying on random perturbation alone takes an enormous amount of iterations for SPSA to arrive at the optimal solution, especially in high dimensional problems. On the other hand, differentiable functions do not suffer from this problem. Motivated by this, MSPSA computes a new gradient $g_{o,d,p_1,k}$ for $x_{o,d,p_1,k}$ using the average of $g'_{o,d,p_1,k}$ and the gradient of a differentiable loss function $\hat{\mathcal{L}}_k$ as shown in Eq. (16).

$$g_{o,d,p_1,k} = \frac{g'_{o,d,p_1,k} + \nabla_{x_{o,d,p_1,k}} \hat{\mathcal{L}}_k(x_k)}{2} \tag{16}$$

$$\hat{\mathcal{L}}_k(x_k) = \sum_{m,p_2} (y''_{m,p_2} - \hat{f}_{m,p_2,k}(x_k))^2 \tag{17}$$

$\nabla_{x_{o,d,p_1,k}}$ is the gradient operator with respect to $x_{o,d,p_1,k}$. $\hat{f}_{m,p_2,k}$ is the metamodel function which estimates vehicle counts on road m at time interval p_2 at iteration k based on Eq. (18).

$$\hat{f}_{m,p_2,k}(x_k) = \sum_{o,d,p_1} x_{o,d,p_1,k} \times \eta_{o,d,p_1,m,p_2,k} \tag{18}$$

$$\eta_{o,d,p_1,m,p_2,k} = \frac{x_{o,d,p_1,m,p_2,k}}{\overline{x}_{o,d,p_1,k}} \tag{19}$$

$\overline{x}_{o,d,p_1,k}$ is the departing OD trips between o and d at time interval p_1. $\eta_{o,d,p_1,m,p_2,k}$ is the fraction of $\overline{x}_{o,d,p_1,k}$ that can pass through road m at time interval p_2 at iteration k in the simulation. After $g_{o,d,p_1,k}$ is computed, a tentative solution $x'_{o,d,p_1,k}$ at iteration k can be obtained based on Eq. (20).

$$x'_{o,d,p_1,k} = x_{o,d,p_1,k} - a_k g_{o,d,p_1,k} \tag{20}$$

When solving a problem in the continuous domain, $x'_{o,d,p_1,k}$ could be used as the solution estimate at the next iteration. However, OD calibration is a discrete problem. In a situation where 10 input OD parameters are similarly correlated with 1 vehicle count output measurement which is 3 units below its target value, moving each OD parameter by positive 1 unit overshoots the target value by 7 units. So, in this situation, not all parameters should be updated even though they are similarly correlated with the output. Therefore, MSPSA incorporates an integer programming method to fine-tune the estimation. Different from the existing SPSA algorithms, the new solution vector x_{k+1} of MSPSA at iteration $k+1$ is obtained by minimizing the objective function of an integer program as shown in Eq. (21). The constraints of integer programming are shown in Eqs. (22) to (26).

$$x_{k+1} = \arg\min_x \sum_{m,p_2} \hat{E}_{m,p_2}(x) + W_0 \sum_{o,p_1} \overline{\overline{E}}_{o,p_1}(x) \tag{21}$$

$$\hat{E}_{m,p_2}(x) = |y''_{m,p_2} - \hat{f}_{m,p_2,k}(x)| \tag{22}$$

$$\overline{\overline{E}}_{o,p_1}(x) = \left(\sum_d x_{o,d,p_1}\right) - \overline{\overline{x}}_{o,p_1,k} \tag{23}$$

$$x_{o,d,p_1} \in \mathbb{N}_0 \tag{24}$$

$$x_{o,d,p_1} \geq \min\{x_{o,d,p_1,k}, x'_{o,d,p_1,k}\} \tag{25}$$

$$x_{o,d,p_1} \leq \max\{x_{o,d,p_1,k}, x'_{o,d,p_1,k}\} \tag{26}$$

$\hat{E}_{m,p_2}(x)$ is the absolute difference between y''_{m,p_2} and $\hat{f}_{m,p_2,k}(x)$. $\overline{\overline{E}}_{o,p_1}(x)$ is the number of vehicles that exceeds the road capacity $\overline{\overline{x}}_{o,p_1,k}$ of origin o at time interval p_1. W_0 is a scalar used to scale the importance of $\overline{\overline{E}}_{o,p_1}(x)$ during minimization. $\overline{\overline{E}}_{o,p_1}(x)$ is considered because, in a high dimensional OD calibration problem, many OD pairs share the same origin. Therefore, it is important to respect the origin road's capacity to avoid the saturation of the origin road, or else there is a discrepancy between the number of vehicles intended by the calibration algorithm and the number of vehicles generated in the simulation. This discrepancy introduces noise during the calibration process. Since $\overline{\overline{x}}_{o,p_1,k}$ depends on congestion pattern which is dynamic and complex, the value of $\overline{\overline{x}}_{o,p_1,k}$ is discovered through running simulation at iteration k as shown in Eq. (27). If the origin flow intended by MSPSA algorithm exceeds the simulated origin flow, the value of $\overline{\overline{x}}_{o,p_1,k}$ is set to the simulated origin flow, else it is set to the origin's free flow capacity.

Algorithm 1 MSPSA Algorithm

1: **procedure** MSPSA($\mathcal{L}$, x_0, c, A, α, γ, W_0, $\overline{\overline{g}}$)
2: $k \leftarrow 0$
3: $x_k \leftarrow x_0$
4: $x_b \leftarrow x_0$
5: **while** Termination criteria is not met **do**
6: Simulate $f(x_k)$
7: Compute $\overline{\overline{x}}_{o,p_1,k}$ using Eq. (27)
8: Compute $\eta_{o,d,p_1,m,p_2,k}$ using Eq. (19)
9: Generate Δ_k
10: Compute c_k using Eq. (12)
11: Obtain x_k^+ and x_k^- from Eq. (9)
12: Simulate $f(x_k^+)$ and $f(x_k^-)$
13: $x_b \leftarrow \arg\min_{x \in \{x_b, x_k, x_k^+, x_k^-\}} \{\mathcal{L}(x)\}$
14: Compute $g_{o,d,p_1,k}$ using Eq. (16)
15: **if** $k = 0$ **then**
16: Compute a using Eq. (28)
17: Compute a_k using Eq. (11)
18: Compute $x'_{o,d,p_1,k}$ using Eq. (20)
19: Compute x_{k+1} using integer program based on Eq. (21)
20: $k \leftarrow k+1$
21: **return** x_b

$$\overline{\overline{x}}_{o,p_1,k} = \begin{cases} \sum_d \overline{x}_{o,d,p_1,k} & \text{if} \sum_d x_{o,d,p_1,k} \geq \sum_d \overline{x}_{o,d,p_1,k} \\ N_o^l \times \overline{\overline{l}} & \text{otherwise} \end{cases} \tag{27}$$

N_o^l is the number of lanes of the origin road o and $\overline{\overline{l}}$ is the free flow lane capacity. Eq. (24) ensures that x_{o,d,p_1} is a natural number including zero. Eq. (25) and Eq. (26) ensure that the value of $x_{o,d,p_1,k+1}$ stays between $x_{o,d,p_1,k}$ and $x'_{o,d,p_1,k}$ since the linear assumption made by the metamodel might not hold true if the x is far away from x_k.

D. MSPSA Algorithm Description

This section describes the proposed MSPSA algorithm step by step. The MSPSA algorithm is shown in Algorithm 1. Lines 2 to 4 describe the initialization stage of MSPSA. At this stage, x_k and x_b are set to the initial estimate x_0. x_b is used to store the best solution obtained during the calibration process. Lines 5 to 20 describe the main optimization loop of MSPSA. First, x_k is simulated to obtain $\mathcal{L}(x_k)$. After the simulation, the values of $\overline{\overline{x}}_{o,p_1,k}$ and $\eta_{o,d,p_1,m,p_2,k}$ are also estimated at Line 7 and Line 8, respectively. Then, at Lines 9 to 11, a random perturbation vector Δ_k is generated and the value of c_k is computed to obtain perturbed solutions $x_k^{\pm}$. After that,

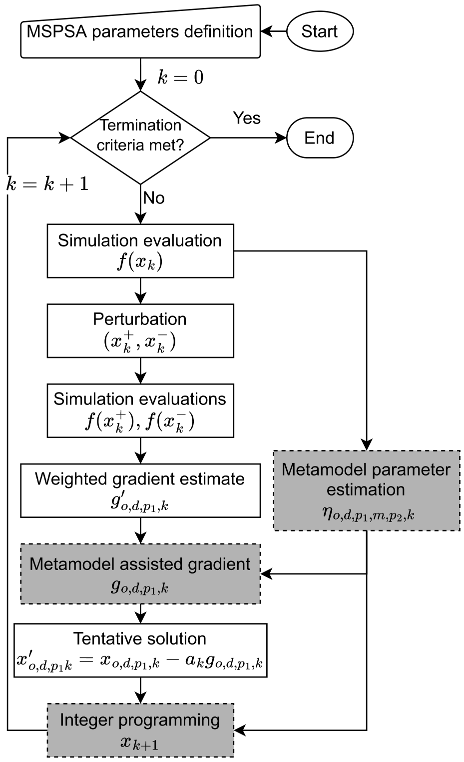

Fig. 1. The flowchart of the proposed MSPSA algorithm. The differences between MSPSA and SPSA are highlighted using grey boxes.

at Line 12, both x_k^- and x_k^- are simulated to obtain $\mathcal{L}(x_k^+)$ and $\mathcal{L}(x_k^-)$, respectively. The best solution obtained so far is assigned to x_b at Line 13. At Line 16, the value of a is calculated using Eq. (28) if the current iteration is the first iteration.

$$a = \frac{\overline{\overline{g}} \times (A+1)^{\alpha}}{\max\{g_{o,d,p_1,0} : (o,d) \in \mathcal{Q}, p_1 \in \mathcal{P}\}} \tag{28}$$

$\overline{\overline{g}}$ is the maximum allowable step size to be taken by each OD parameter at the first iteration. Then, the values of a_k and $x'_{o,d,p_1,k}$ are computed at Line 17 and Line 18, respectively. Finally, at Line 19, an integer program is solved to obtain x_{k+1}. The procedure from Line 5 to Line 20 is repeated until termination criteria is met, such as when the maximum number of iterations is reached. The best solution x_b found so far is returned at Line 21. The flowchart of the proposed MSPSA algorithm is shown in Fig. 1.

V. Simulation Setup

A. Simulator

In this research, an open-source traffic simulator called Simulation of Urban Mobility (SUMO) [23] is used. The resolution of the simulation is mesoscopic, where vehicles move between queues based on queueing theory. The simulator is set to use dynamic routing mode. In this mode, each vehicle computes the fastest path between its origin and destination at its departure time. Then, a portion of vehicles periodically recomputes their fastest paths based on the recent state of traffic in the network. This routing approach allows vehicles to adapt to traffic jams and other changes in the network.

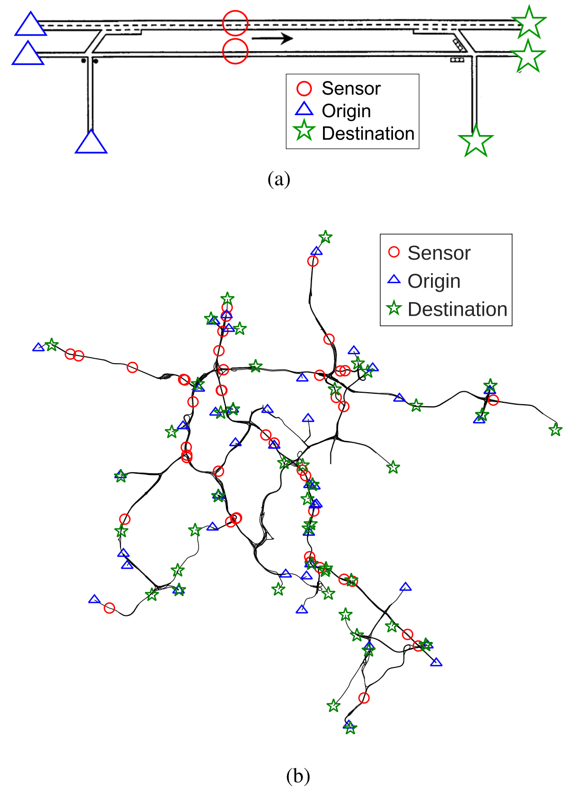

Fig. 2. Two transportation networks used which are (a) synthetic toy network proposed by Astarita et al. [25] and (b) KL network. Two symbols may overlap with each other if they are on different sides of a two-way road.

The flexibility of this mode facilitates future research works, such as scenario planning. The rerouting probability is set to 0.5 to have an outcome similar to dynamic user equilibrium (DUE) [24]. The rerouting period is set to 60 seconds.

B. Transportation Networks

The proposed MSPSA algorithm is validated using two different transportation networks. One simple toy network and one complex real-world KL network. The toy network and the KL network are shown in Fig. 2a and Fig. 2b, respectively. The network topology of the toy network is proposed by Astarita et al. [25]. Based on Fig. 2a, the toy network consists of 2 observable roads, 3 origins, and 3 destinations. Since every origin can connect to every destination, there are a total of $3 \times 3 = 9$ OD pairs in the toy network. The map of KL network is retrieved from OpenStreetMap [26]. The roads without predecessors are selected to be the origins and the roads without successors are selected to be the destinations. There are 43 origins and 49 destinations in the KL network. This results in 1615 OD pairs which can be connected by at least one path in the KL network. The sensor locations in KL network are provided by the traffic authority. After associating each sensor with its nearby road, around 36 roads in KL network (Fig. 2b) are considered observable roads.

C. Traffic Scenarios

In terms of test beds, three different traffic scenarios have been modelled:

TABLE II

DESCRIPTION OF THE DATA FOR THE TOY NETWORK AND THE KL NETWORK

	Toy Network	Kuala Lumpur (KL) Network
Number of roads	12	1106
Number of observed roads	2	36
Number of OD pairs	9	1615
Number of time intervals	6	6
Dimension size of OD parameter vector	54	9690
Interval length	15 mins	15 mins
Number of vehicles generated	4800	15000
Rerouting probability	0.5	0.5
Rerouting interval (s)	60	60
Maximum number of iterations	50	50
$\bar{\bar{l}}$ (vehs/15 mins)	400	400

1) Scenario 1: This scenario uses a small-scale toy network and synthetic sensor measurements as ground truth. This scenario is used primarily for debugging and verification purposes.

2) Scenario 2: This scenario uses a real-world KL network and synthetic sensor measurements as ground truth.

3) Scenario 3: This scenario uses real-world KL network and real-world sensor data as ground truth.

Scenarios using synthetic measurements (Scenario 1 and Scenario 2) are prepared using the guidelines provided by Antoniou et al. [27]. First, OD trips are randomly generated and used as input for the traffic simulator. These OD trips are treated as the true OD demand. After the simulation is completed, the output measurements of the simulator are used as the calibration targets. Then, the initial OD estimate for each synthetic scenario is set to 70% of its true value with uniform random fluctuations over each OD pair in the range of ±15% of the true value, as shown in Eq. (29).

$$x_0 = (0.7 + 0.3 \times U[0, 1]) \odot x'' \tag{29}$$

x'' is the true OD demand vector. $U[0, 1]$ is a random vector with a continuous uniform distribution between 0 and 1. $\odot$ is the element-wise multiplication operator. x_0 is the initial OD estimate at $k = 0$. On the other hand, Scenario 3 uses the randomly generated OD trips directly as the initial OD estimate and real-world sensor measurements as the calibration target. So, it is not guaranteed that the initial OD estimate is close to the true OD demand. The real-world sensor measurements are taken during the morning peak hour (7:00-8:00 AM (GMT+8)) on Monday.

There are around 4800 and 15000 vehicles generated for the toy network and KL network, respectively. The duration of all traffic scenarios is set to 90 minutes, with 30 minutes warm-up period and 60 minutes calibration period. The 90-minute duration is divided into six 15-minute intervals. Hence, the numbers of OD parameters to be estimated are $9 \times 6 = 54$ and $1615 \times 6 = 9690$ for toy network and KL network, respectively. The maximum number of iterations is set to 50, which is equivalent of 150 simulation evaluations, for each scenario. The network parameters are summarized in Table II.

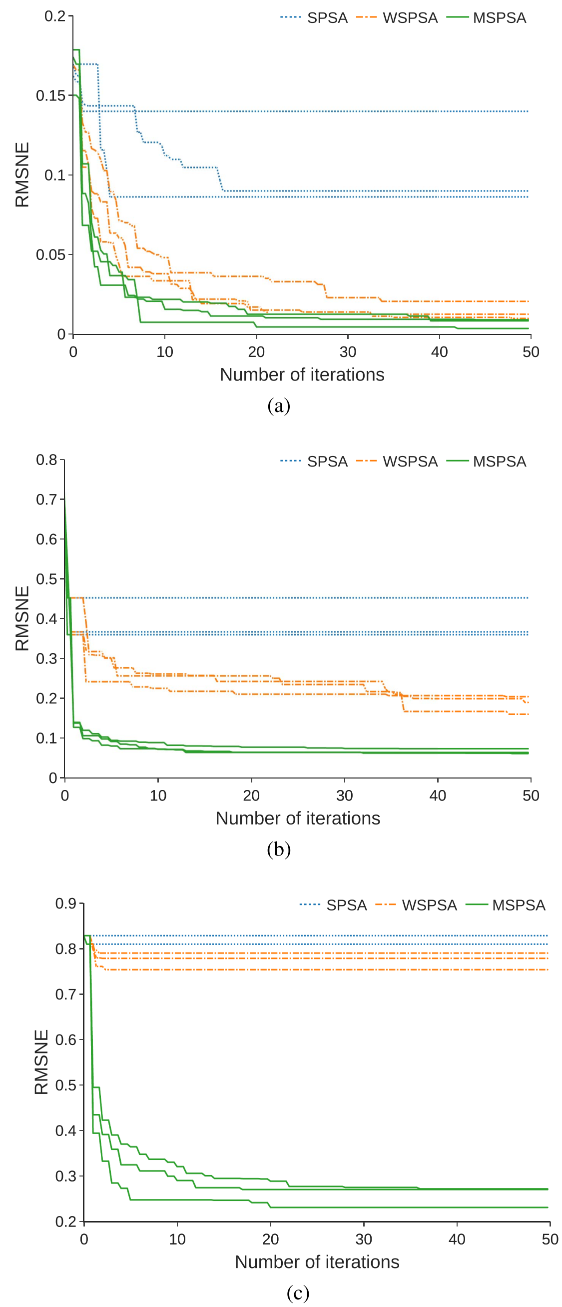

Fig. 3. Performance of each method for (a) Scenario 1 with toy network and synthetic sensor measurements, (b) Scenario 2 with KL network and synthetic sensor measurements, and (c) Scenario 3 with KL network and real-world sensor measurements. The x-axis is the number of iterations and the y-axis is the value of RMSNE evaluated at x_b.

D. Loss Function and Evaluation Metrics

The performance of the proposed MSPSA algorithm is evaluated using RMSNE, which is also $\mathcal{L}''$. RMSNE is used in many similar researches [15], [21] and is defined in Eq. (30). In addition to RMSNE, root mean squared error (RMSE) and percentage change from initial point (PCIP) in RMSNE are also used as evaluation metrics in this research. RMSE and PCIP are defined in Eq. (31) and Eq. (32), respectively.

$$\text{RMSNE} = \frac{\sqrt{|\mathcal{M}'||\mathcal{P}'| \sum_{m,p_2} (y''_{m,p_2} - y_{m,p_2})^2}}{\sum_{m,p_2} y''_{m,p_2}} \tag{30}$$

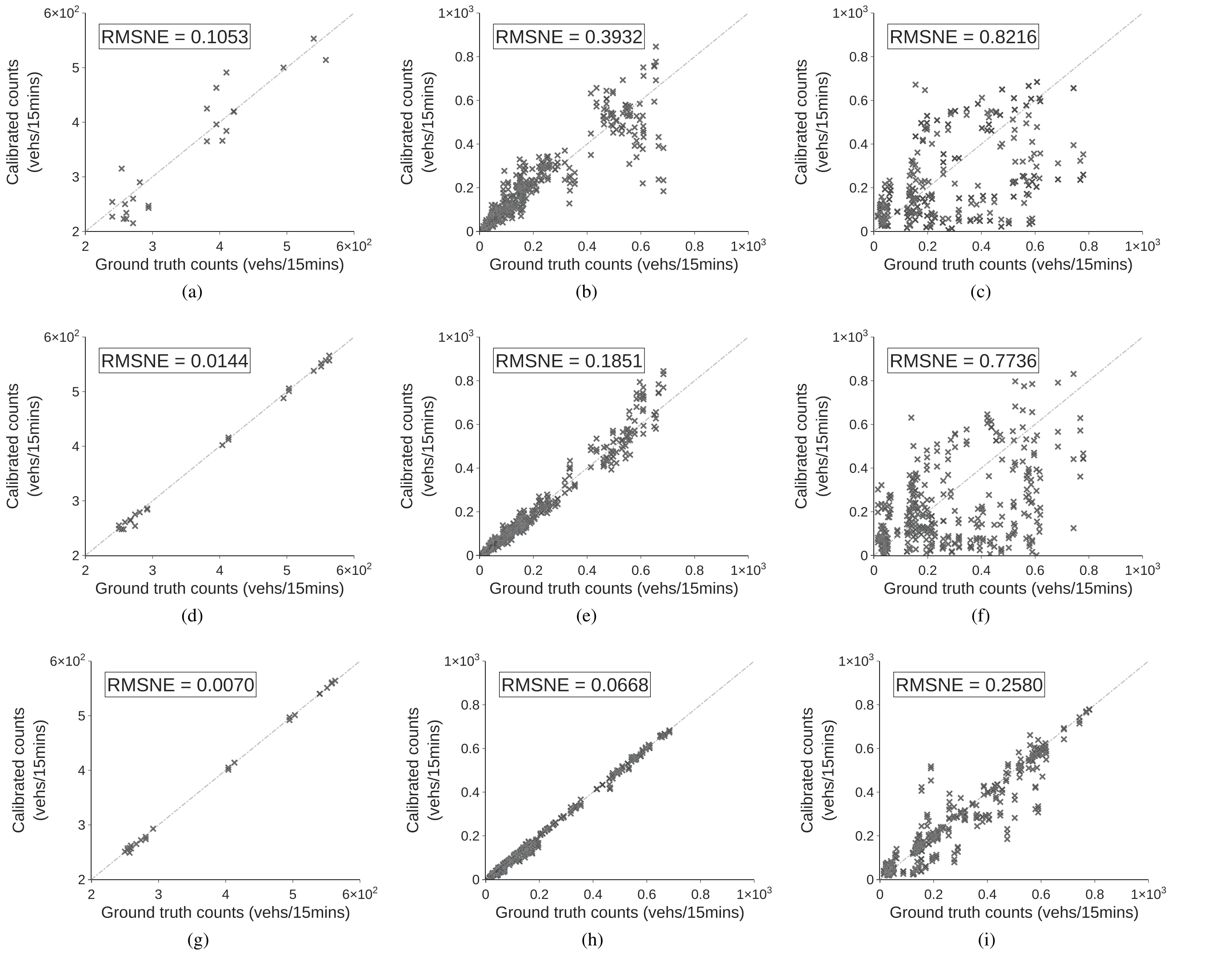

Fig. 4. Comparison between ground truth vehicle counts and calibrated vehicle counts for (a) SPSA in Scenario 1, (b) SPSA in Scenario 2, (c) SPSA in Scenario 3, (d) WSPSA in Scenario 1, (e) WSPSA in Scenario 2, (f) WSPSA in Scenario 3, (g) MSPSA in Scenario 1, (h) MSPSA in Scenario 2, and (i) MSPSA in Scenario 3. The x-axis is the ground truth vehicle counts and the y-axis is the calibrated vehicle counts.

$$\text{RMSE} = \sqrt{\frac{\sum_{m,p_2}(y''_{m,p_2} - y_{m,p_2})^2}{|\mathcal{M}'||\mathcal{P}'|}} \tag{31}$$

$$\text{PCIP} = 100 \times \frac{\mathcal{L}(x_0) - \mathcal{L}(x_b)}{\mathcal{L}(x_0)} \tag{32}$$

E. Implementation of Calibration Algorithms

The performance of the proposed MSPSA algorithm is compared against other approaches which are SPSA [5] and WSPSA [6]. The values of A, c, α and γ are set to 5, 5, 0.602, and 0.101, respectively. These parameters are chosen based on the guidelines provided by Spall [28]. The values of $\overline{\overline{g}}$ are set to 20, 20 and 100 for Scenario 1, Scenario 2, and Scenario 3, respectively. The value of W_0 is set to 100. To reduce the noise caused by perturbations in high dimensional problems, all calibration algorithms only perturb 50% and 30% of the OD parameters at each iteration for Scenario 2 and Scenario 3, respectively.

F. Computing Environment

All the calibration algorithms are run on a Ubuntu 18.0.4 LTS server with 48 CPU cores (Intel Xeon CPU @ 3.00GHz) and 62 GB memory. The models are built using Python 3.10.6 (i.e. Scipy 1.9.1 [29] for integer programming and Tensorflow 2.10 [30] for gradient computation).

VI. Results and Discussion

A. Performance Evaluation

This section presents the OD calibration results in all scenarios. The RMSNEs achieved by all calibration algorithms in Scenario 1, Scenario 2, and Scenario 3 are shown in Fig. 3a, Fig. 3b, and Fig. 3c, respectively. Each calibration algorithm has 3 curves, which correspond to experiments with different random seeds, in each plot. Different random seeds also provide different initial estimates for Scenario 1 and Scenario 2. A lower value of RMSNE indicates a better calibration accuracy of a calibration algorithm.

TABLE III
PERFORMANCE MEASURES OF DIFFERENT CALIBRATION ALGORITHMS FOR THREE TRAFFIC SCENARIOS

Scenario	Approach	RMSNE	RMSE (#vehs)	PCIP (%)	Average Simulation Time (s)
1	SPSA[5]	0.1053	36.51	36.98	4.65
1	WSPSA[6]	0.0144	5.55	91.32	5.48
1	MSPSA	**0.0070**	**2.69**	**95.88**	**5.05**
2	SPSA[5]	0.3932	78.85	44.79	173.65
2	WSPSA[6]	0.1851	37.12	74.02	**30.84**
2	MSPSA	**0.0668**	**13.39**	**90.63**	55.50
3	SPSA[5]	0.8216	243.87	0.76	269.56
3	WSPSA[6]	0.7736	229.62	6.55	85.45
3	MSPSA	**0.2580**	**76.59**	**68.83**	**70.91**

Based on Fig. 3a, SPSA is very sensitive to the initial estimate in Scenario 1. The figure also shows that the RMSNEs of WSPSA and MSPSA are able to converge a value less than 0.05 within 10 iterations. Therefore, all calibration algorithms are implemented correctly. In Scenario 2, the best solution for each experiment replication obtained by SPSA is actually a perturbed solution at first iteration. After that, there are no RMSNE improvements made by SPSA. On the other hand, WSPSA and MSPSA achieve their best RMSNEs within 40 and 20 iterations, respectively. In Scenario 3, the initial RMSNE is the highest among all scenarios because the initial estimate is far away from the optimal solution. Thus, the RMSNEs of SPSA and WSPSA fail to converge to a value below 0.7. On the other hand, the RMSNEs of MSPSA converge quickly at the first 10 iterations and are able to reach values below 0.3 within 20 iterations, showing MSPSA's unmatched performance. Overall, the convergence speeds of RMSNEs of MSPSA in all scenarios are the highest among all calibration algorithms. In addition, the proposed MSPSA is robust against the quality of the initial estimates, given that the RMSNEs of MSPSA converge to a similar value for each scenario.

The numerical results for all scenarios are shown in Table III. The values of RMSNE and RMSE shown in Table III are the values evaluated at x_b at the last simulation evaluation and are averaged across different experiment replications. #vehs denotes the number of vehicles. The average simulation time is defined as the average time taken between simulation evaluations.

Based on Table III, the proposed MSPSA achieves the lowest RMSNE (i.e. 0.007, 0.0668, and 0.2580 for Scenario 1, Scenario 2, and Scenario 3, respectively) and RMSE (i.e. 2.69, 13.39, and 76.59 for Scenario 1, Scenario 2, and Scenario 3, respectively) compared to other calibration algorithms for all scenarios. As a result, the PCIP of MSPSA is the highest (i.e. 95.88%, 90.63%, and 68.83% for Scenario 1, Scenario 2, and Scenario 3, respectively) for all scenarios. It is worth noting that the RMSNEs of all calibration algorithms for Scenario 3 are higher than that of Scenario 1 and Scenario 2. This is because the ground truth data in Scenario 3 is obtained from real-world traffic sensors. This means that the sensor noise and the behavioural difference between the traffic simulator and the real world contribute to the losses, hence, reducing the PCIP for Scenario 3.

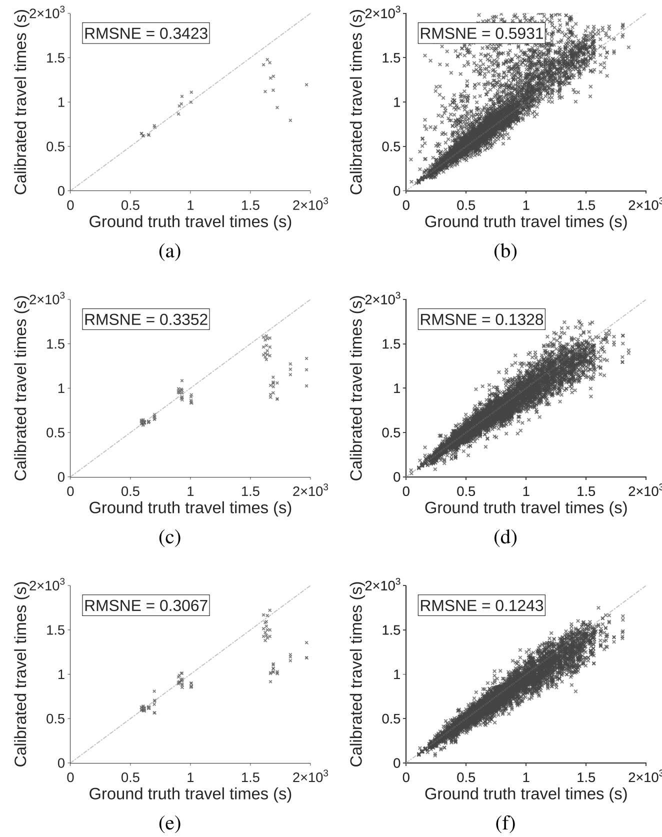

Fig. 5. Comparison between ground truth OD travel times and calibrated OD travel times for (a) SPSA in Scenario 1, (b) SPSA in Scenario 2, (c) WSPSA in Scenario 1, (d) WSPSA in Scenario 2, (e) MSPSA in Scenario 1, and (f) MSPSA in Scenario 2. The x-axis is the ground truth OD travel times and the y-axis is the calibrated OD travel times.

Based on Table III, the average simulation time increases with the network size and level of travel demand. For Scenario 1, the average simulation times of all calibration algorithms are similar to each other. For Scenario 2 and Scenario 3, the average simulation time of SPSA is the longest (i.e. 173.65 and 269.56 seconds, respectively) because it overestimates the OD demand in the network, causing the traffic network to be saturated. In Scenario 2, the average simulation time of MSPSA (i.e. 55.50 seconds) is higher than that of WSPSA (i.e. 30.84 seconds) because additional time is taken to solve the integer program which considers a larger network. However, this additional time is out weighted by the time taken to simulate the saturated network caused by WSPSA (i.e. 85.45 seconds) in Scenario 3. Thus, MSPSA algorithm is able to achieve the lowest RMSNE while keeping the average simulation time reasonable in each scenario. In summary, the proposed MSPSA outperforms SPSA (and WSPSA) by 93.35%, 83.01%, and 68.60% (51.39%, 63.91%, and 66.65%) in terms of RMSNE in Scenario 1, Scenario 2, and Scenario 3, respectively. On average, the MSPSA outperforms SPSA and WSPSA by 81.65% and 60.65%, respectively.

B. Comparison Between Groud Truth Vehicle Counts and Calibrated Counts

Fig. 4 illustrates the 45-degree plots to compare the ground truth vehicle counts with calibrated vehicle counts by all calibration algorithms in all scenarios. The left, middle, and right columns correspond to Scenario 1, Scenario 2, and Scenario 3, respectively. The top, middle, and bottom rows correspond to SPSA, WSPSA, and MSPSA, respectively. The scatteredness in the subfigures of Fig. 4 replicates and validates the RMSNE values achieved by all calibration algorithms.

C. OD Travel Times Comparison

Fig. 5 illustrates the 45-degree plots to compare the ground truth OD travel times with calibrated OD travel times by all calibration algorithms in Scenario 1 and Scenario 2. The left and right columns correspond to Scenario 1 and Scenario 2, respectively. The top, middle, and bottom rows correspond to SPSA, WSPSA, and MSPSA, respectively. Although OD travel times are not included in the calibration loss function, MSPSA achieves the lowest RMSNE of OD travel times in each scenario (i.e. 0.3067 and 0.1243 for Scenario 1 and Scenario 2, respectively) compared to SPSA and WSPSA. This shows the superiority of MSPSA over other calibration algorithms because it can produce the lowest RMSNEs in vehicle counts and OD travel times at the same time.

VII. Conclusion

OD calibration is an important and challenging task due to the high-dimensional input and expensive-to-evaluate simulation function. This paper proposes an efficient algorithm, called MSPSA, to speed up this high-dimensional OD calibration process. The proposed MSPSA consists of a differentiable metamodel function which provides a gradient to guide the search direction of SPSA. The metamodel is also used in an integer program to fine-tune the solution. The effectiveness of the proposed MSPSA is validated on a simple toy network and complex KL network. Synthetic and real-world ground truth data are considered for each transportation network. The proposed MSPSA is benchmarked against SPSA and state-of-the-art WSPSA. Experiments with a computational budget of 150 simulation evaluations have been carried out. Based on the simulation results, the proposed MSPSA outperformed SPSA and WSPSA by at least 50% in terms of RMSNE in both synthetic and real-world scenarios within a tight computational budget. Furthermore, the proposed MSPSA is robust against the quality of initial OD estimation. This allows MSPSA to be used in a real-world situation where prior OD is not available. Future research includes extending MSPSA to calibrate OD parameters using other types of sensor measurement such as vehicle speed and density [11]. To achieve this, only differentiable functions that map vehicle counts to these measurements are required, since this paper already defined the metamodel function which maps the OD parameters to vehicle counts. If the measured values are continuous, a mixed integer program can be used to fine-tune the estimation.

References

[1] C. Osorio, "High-dimensional offline origin-destination (OD) demand calibration for stochastic traffic simulators of large-scale road networks," *Transp. Res. B, Methodol.*, vol. 124, pp. 18–43, Jun. 2019.

[2] H. J. Van Zuylen and L. G. Willumsen, "The most likely trip matrix estimated from traffic counts," *Transp. Res. B, Methodol.*, vol. 14, no. 3, pp. 281–293, 1980.

[3] M. G. H. Bell, "The real time estimation of origin-destination flows in the presence of platoon dispersion," *Transp. Res. B, Methodol.*, vol. 25, nos. 2–3, pp. 115–125, 1991.

[4] E. Cascetta, "Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator," *Transp. Res. B, Methodol.*, vol. 18, nos. 4–5, pp. 289–299, Aug. 1984.

[5] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992.

[6] L. Lu, Y. Xu, C. Antoniou, and M. Ben-Akiva, "An enhanced SPSA algorithm for the calibration of dynamic traffic assignment models," *Transp. Res. C, Emerg. Technol.*, vol. 51, pp. 149–166, Feb. 2015.

[7] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[8] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[9] V. Vaze, C. Antoniou, Y. Wen, and M. Ben-Akiva, "Calibration of dynamic traffic assignment models with point-to-point traffic surveillance," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2090, no. 1, pp. 1–9, Jan. 2009.

[10] M. E. Ben-Akiva, S. Gao, Z. Wei, and Y. Wen, "A dynamic traffic assignment model for highly congested urban networks," *Transp. Res. C, Emerg. Technol.*, vol. 24, pp. 62–82, Oct. 2012.

[11] B. Kostic and G. Gentile, "Using traffic data of various types in the estimation of dynamic O-D matrices," in *Proc. Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, Jun. 2015, pp. 66–73.

[12] 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," *Transp. Res. Proc.*, vol. 7, pp. 233–253, Jan. 2015.

[13] A. Tympakianaki, H. N. Koutsopoulos, and E. Jenelius, "C-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 231–245, Jun. 2015.

[14] T. Djukic, J. W. C. Van Lint, and S. P. Hoogendoorn, "Application of principal component analysis to predict dynamic origin–destination matrices," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2283, no. 1, pp. 81–89, Jan. 2012.

[15] A. A. Prakash, R. Seshadri, C. Antoniou, F. C. Pereira, and M. Ben-Akiva, "Improving scalability of generic online calibration for real-time dynamic traffic assignment systems," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 48, pp. 79–92, Dec. 2018.

[16] M. Qurashi, T. Ma, E. Chaniotakis, and C. Antoniou, "PC–SPSA: Employing dimensionality reduction to limit SPSA search noise in DTA model calibration," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1635–1645, Apr. 2020.

[17] M. Qurashi, Q.-L. Lu, G. Cantelmo, and C. Antoniou, "Dynamic demand estimation on large scale networks using principal component analysis: The case of non-existent or irrelevant historical estimates," *Transp. Res. C, Emerg. Technol.*, vol. 136, Mar. 2022, Art. no. 103504.

[18] Q. Wang and J. C. Spall, "Discrete simultaneous perturbation stochastic approximation on loss function with noisy measurements," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 4520–4525.

[19] S. Oh, R. Seshadri, C. L. Azevedo, and M. E. Ben-Akiva, "Demand calibration of multimodal microscopic traffic simulation using weighted discrete SPSA," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 5, pp. 503–514, May 2019.

[20] T. Toledo and T. Kolechkina, "Estimation of dynamic origin–destination matrices using linear assignment matrix approximations," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 618–626, Jun. 2013.

[21] C. Osorio, "Dynamic origin-destination matrix calibration for large-scale network simulators," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 186–206, Jan. 2019.

[22] T. Dantsuji, N. H. Hoang, N. Zheng, and H. L. Vu, "A novel metamodel-based framework for large-scale dynamic origin–destination demand calibration," *Transp. Res. C, Emerg. Technol.*, vol. 136, Mar. 2022, Art. no. 103545.

[23] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.

[24] M. Ashfaq, Z. Gu, S. T. Waller, and M. Saberi, "Comparing dynamic user equilibrium and noniterative stochastic route choice in a simulation-based dynamic traffic assignment model: Practical considerations for large-scale networks," *J. Adv. Transp.*, vol. 2021, pp. 1–16, May 2021.

[25] V. Astarita, K. Er-Rafia, M. Florian, M. Mahut, and S. Velan, "Comparison of three methods for dynamic network loading," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1771, no. 1, pp. 179–190, Jan. 2001.

[26] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct./Dec. 2008.

[27] C. Antoniou et al., "Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation," *Transp. Res. C, Emerg. Technol.*, vol. 66, pp. 79–98, May 2016.

[28] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 817–823, Jul. 1998.

[29] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, Feb. 2020.

[30] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[31] *Recogine Technology Sdn Bhd*. Accessed: Jun. 1, 2022. [Online]. Available: https://recogine.com/

Mun Chon Ho received the B.E. degree in electrical and computer system engineering from Monash University Malaysia in 2018, where he is currently pursuing the Ph.D. degree in electrical and computer system engineering. His research interests include artificial intelligence and simulation calibration.

Joanne Mun-Yee Lim received the Ph.D. degree in engineering from Multimedia University. She is currently a Professional Engineer with the Board of Engineers Malaysia and a Senior Lecturer with the ECSE, School of Engineering, Monash University Malaysia. Her research interests include transportation, vehicular networks, the Internet of Things (IoT), artificial intelligence, unmanned aerial vehicle (UAV), and applications.

Chun Yong Chong (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Malaya, Malaysia, in 2012 and 2016, respectively. He is currently a Senior Lecturer with the School of Information Technology, Monash University Malaysia. His current research interests include software maintenance, software clustering, software remodularization, software fault prediction, and metamorphic testing.

Kah Keong Chua received the B.IT. degree in computer engineering and the M.Eng.Sc. degree from UTAR. He has several years of industry experience focusing on intelligent transportation systems (ITS). He has years of experience in image processing and machine learning. Currently, he is one of the Research and Development Core Members with Recogine Technology, a leading company in Malaysia ITS sector.

Alvin Kuok Lim Siah received the B.Sc. and M.Eng. degrees in computer science from UTAR. He has many years of experience in software engineering in companies, such as Intel Technology, Recogine Technology, and Fermion Laboratory. He had filed several trade secrets invention for manufacturing automation, resource optimization planning and computer vision for visual inspection with Intel assembly and test manufacturing. Over the past years, he had led the Research and Development at Recogine Technology for Malaysia ITS sector.