Efficient Gradient Estimation of Traffic Assignment Models with Iterative Backpropagation

A.U.Z Patwary*[1], Hong Lo[2] and Francesco Ciari[1]

[1]Dept. of Civil, Geological and Mining Engineering, Polytechnique Montréal, Canada

[2] Department of Civil Engineering, Hong Kong University of Science and Technology

SHORT SUMMARY

Traffic assignment (TA) optimization is at the heart of many transportation planning and operation problems. For a reasonably sized network with high-dimensional decision variables, TA optimization quickly becomes intractable due to high computation time and a large number of function evaluations. Generally, TA models have cyclic dependencies among their components and hence, have no closed-form gradients, which is crucial for high dimensional optimization. This paper proposes an efficient TA gradient estimation technique called Iterative Backpropagation (IB) to solve this problem. IB exploits the iterative TA solution algorithms and generates the TA gradients while the TA model converges. IB neither requires solving any system of equations nor any additional functional evaluations irrespective of the problem dimension. In our experiments, IB gradients match the finite-difference gradients at machine precision. IB gradients are usable with any state-of-the-art gradient-based optimization algorithms and can be extended to a wide range of TA optimization problems.

Keywords: Iterative Backpropagation, TA Optimization, Gradient-based, OD estimation

1. INTRODUCTION

Network optimization or network design with an embedded traffic assignment (TA) model is at the heart of transportation planning and operations. Many transportation problems incorporate traffic assignment models into their formulation to capture the effects of equilibrium route choice and road congestion (Lee et al., 2020; Yang & Bell, 1997). The goal is generally determining a particular set of control variables to achieve a specific desired outcome (Liu et al., 2017; Yang & Bell, 1997). Another problem involving TA optimization is calibration (Lee et al., 2020; Patwary et al., 2021). TA models often require a combination of many submodules and procedural steps, making them computationally expensive, lacking closed-form solution and gradient information. Optimization of such a model involving a large network and a large number of decision variables, thereby, quickly becomes intractable. This paper proposes an efficient gradient estimation method to solve high-dimensional optimization problems with an embedded TA model. We demonstrate the proposed algorithm in the context of origin-destination (OD) demand calibration. The approach, however, applies to a wide variety of network optimization or network design problems with an embedded TA model, opening up an efficient way to solve large-scale bi-level problems.

Many techniques have been developed in the literature to solve bi-level problems, where the lower level solves the TA problem and the upper level optimizes the decision variables iteratively (Yang & Bell, 1997). For simulation-based TA optimization problems, general-purpose black-box algorithms, such as genetic algorithm (GA) (Chiappone et al., 2016), simultaneous perturbation stochastic approximation (SPSA) (Oh et al., 2019), neural network (Otković et al., 2013), etc. have been developed. Some hybrid semi- black-box algorithms have also been proposed where a simplified traffic model is used to mimic the physical properties of the assignment model in a trust region framework (Patwary et al., 2021; Zhang et al., 2017). In the context of OD estimation for static TA models, multiple gradient, sub-gradient and approximate gradient methods have been developed to solve the problem. Tobin and Friesz (1988) proposed a sensitivity analysis method for optimizing TA assignment models. However, this method requires less number of routes than the number of links (Patriksson, 2004) which does not hold for large-scale application.

A pure gradient descent optimization technique is not bounded by such mathematical assumptions and can inherently incorporate the congestion effect of changing OD demand. However, due to cyclic dependency between different components, TA outputs does not have closed-form gradients. Gradient-based TA optimization generally avoids this cyclic dependency by assuming independence among TA components in the solution process (Spiess, 1990; Yang et al., 2001). Numerical gradients computation methods, on the other hand, require at least the same number of function evaluations as the problem dimension at each iteration of the optimization process, making them computationally prohibitive for high dimensional TA optimization.

SPSA (Spall, 1992), is a very attractive approximate gradient-based large-scale optimization technique that approximates the gradient by simultaneously perturbing all the decision variables. Some exciting modifications have been proposed to increase the applicability of SPSA for large-scale OD demand estimation in recent years (Lu et al., 2015; Oh et al., 2019). The successful use of SPSA demonstrates the necessity and potential of developing efficient gradient estimation techniques and pure gradient-based optimization algorithms.

This paper proposes an efficient gradient estimation framework called Iterative Backpropagation (IB) for network optimization with an embedded TA model. IB takes advantage of the iterative TA solution algorithms (e.g., fixed point, method of successive average, gradient projection, Frank Wolf etc.) to simultaneously calculate the analytical gradients at the current evaluation while the assignment process converges. The method requires no additional TA function evaluations and can be parallelized, i.e., it scales well with high dimension.

In this paper, section 2 develops the general framework of the IB algorithm for TA optimization problems. Section 3 then calculates and compares IB and finite-difference gradients for a small two-link network with respect to a single OD demand. Finally, section 4 concludes the paper.

2. METHODOLOGY

Problem Statement

We define an optimization problem in equation (1) where $g: \mathbb{R}^n \rightarrow \mathbb{R}$ is an objective function that maps $\psi * \in \mathbb{R}^n$, the equilibrium output vector from a TA model to an objective value to be minimized. $\psi *$ can consist of equilibrium link volumes, link travel times, transit vehicle exit and entry counts, travel times, transit waiting times, etc. n is the dimension of $\psi *$. $\theta \in \mathbb{R}^s$ is the parameter vector to be optimized within the vector space Θ. s is the dimension of the parameter vector to be optimized.

$$min_{\theta} g\big(\psi * (\theta)\big); \theta \in \Theta \quad (1)$$

For an iterative gradient descent scheme, the update rule for iteration i in the optimization process takes the following form. Here, β is the learning rate of the gradient descent process.

$$\theta^{j+1} = \theta^j - \beta \times \nabla_\theta g\left(\psi * \left(\theta^j\right)\right) \quad (2)$$

Given the gradient vector $\nabla_\theta g\left(\psi * \left(\theta^j\right)\right) \in R^{s\times 1}$ is known, the optimization process can take advantage of any sophisticated gradient-based optimization algorithms such as RMSProp (Hinton et al., 2012) or ADAM (Kingma & Ba, 2015).

Applying the chain rule for multivariate function derivative, the gradient $\nabla_{\theta} g\left(\psi * \left(\theta^{j}\right)\right)$ can be calculated using equation (3). Here, $\nabla_{\psi} g\left(\psi * \left(\theta^{j}\right)\right)$ is the $\mathbb{R}^{n\times 1}$ gradient vector of $\frac{\partial g\left(\psi*\left(\theta^{j}\right)\right)}{\partial \psi_h}$; $h = 1,2, \dots n$ and $\frac{\partial \psi*\left(\theta^{j}\right)}{\partial \theta}$ is the $R^{s\times n}$ Jacobian matrix consisting of $\frac{\partial \psi*_h\left(\theta^{i}\right)}{\partial \theta_k}$; $h = 1, 2, 3 \dots . n$ and $k = 1, 2, 3, \dots . . s$. We use the superscript to denote the number of optimization iteration and the subscript to denote the element of a vector throughout the paper.

$$\nabla_{\theta} g\left(\psi * \left(\theta^{j}\right)\right) = \frac{\partial \psi * \left(\theta^{j}\right)}{\partial \theta} \nabla_{\psi} g\left(\psi * \left(\theta^{j}\right)\right) \tag{3}$$

We calculate the Jacobian matrix $\frac{\partial \psi*\left(\theta^{j}\right)}{\partial \theta}$ consisting of the partial derivative of the traffic assignment model output with respect to the decision variables using the iterative backpropagation method outlined in the following section.

Iterative Backpropagation

TA with fixed demand is to determine a route flow proportion for each route in each OD pair that are non-negative and satisfies some variations of Wardrop's principle (Wardrop, 1952). Given a set of routes $r \in R$ connecting a set of OD pairs $od \in OD$, the equilibrium TA conditions for $\forall r \in R, od \in OD$ can be written in terms of route choice proportion $w_{r,od}$ as shown in equation (4), which depends on the route costs $E_{od} = \{\eta_{r,od} : \forall r \in R_{od}\}$, which themselves are functions of the route flows $F = \{f_{r,od} : \forall r \in R, od \in OD\}$. $w_{r,od}$ here, can be derived based on the multinomial logit model generating the stochastic user equilibrium (SUE) route flows or deterministic assignment procedure generating the user equilibrium (UE) route flows.

$$\begin{gathered} f_{r,od} - w_{r,od}\left(E_{od}(F)\right) q_{od} = 0; \\ F = \left\{f_{r,od} : \forall r \in R, od \in OD\right\}; \\ E = \left\{\eta_{r,od} : \forall r \in R_{od}\right\}; \\ f_{r,od}, w_{r,od} \geq 0 \end{gathered} \tag{4}$$

The iterative solution algorithms for TA problem solves the following sub problem at each iteration.

$$f_{r,od}^{i+1} = f_{r,od}^{i} + \alpha \times \left[q_{od} w_{r,od}\left(E_{od}\left(F^{i}\right)\right) - f_{r,od}^{i}\right]; \forall r \in R, od \in OD \tag{5}$$

Here, $q_{od} w_{r,od}\left(E_{od}\left(F^{i}\right)\right)$ is the approximation of the route flow $f_{r,od}$ based on the set of route costs of the i^{th} iteration $\mathrm{E}_{od}\left(F^{i}\right)$. At solution convergence, this approximation will become exact subject to the acceptable convergence tolerance. α is ideally a monotonically decreasing learning rate. The equilibrium traffic flow is achieved once the gap $\left[q_{od} w_{r,od}\left(E_{od}\left(F^{i}\right)\right) - f_{r,od}^{i-1}\right]$ coverges to zero. The process of calculating $q_{od} w_{r,od}\left(E_{od}\left(F^{i}\right)\right)$, i.e., a set of auxiliary route flows, is also known as network loading. In the IB algorithm, we denote this step as the forward pass.

Assuming the equilibrium route flow is achieved after the I^{th} iteration, the equilibrium route flow gradient $\nabla_{\theta} f_{r,od}^{*}$ with respect to the upper-level decision variable vector θ can be written in the following form.

$$\nabla_\theta f*_{r,od} = \nabla_\theta f_{r,od}^{I} = \nabla_\theta f_{r,od}^{I-1} + \alpha \times \nabla_\theta\left[q_{od} w_{r,od}\left(E_{od}(F^{I-1})\right) - \nabla_\theta f_{r,od}^{I-1}\right]; \forall r \in R, \text{od} \in OD \quad (6)$$

The flow gradient $\nabla_\theta\left[q_{od} w_{r,od}\left(E_{od}(F^{I-1})\right)\right]$ can be calculated using the chain rule of differentiation, as shown in equation (7).

$$\nabla_\theta\left[q_{od} \times w_{r,od}\left(E_{od}(F^{I-1})\right)\right] = \nabla_\theta q_{od} \times w_{r,od}\left(E_{od}(F^{I-1})\right) + q_{od} \times \nabla_{E_{od}} w_{r,od}\left(E_{od}(F^{I-1})\right) \times \frac{\partial E_{od}(F^{I-1})}{\partial \theta} \quad (7)$$

$\frac{\partial E_{od}}{\partial \theta}$ in equation (7) is the Jacobian matrix containing the route cost gradient $\nabla_\theta \eta_{r,od} = \frac{\partial \eta_{r,od}}{\partial \theta}$ at each row. We can again apply the chain rule to calculate the route cost gradient $\nabla_\theta \eta_{r,od}$ as shown in the equation below.

$$\nabla_\theta \eta_{r,od}(F^{I-1}) = \nabla_F \eta_{r,od}(F^{I-1}) \times \frac{\partial F^{I-1}}{\partial \theta} \quad (8)$$

$\frac{\partial F^{I-1}}{\partial \theta}$ is the route flow gradient Jacobian from the previous iteration; i.e., iteration $I-1$ contains the route flow gradient $\nabla_\theta f_{r,od}^{I-1}$ at each row. So, using equations (6)-(8), we can calculate the approximate route flow gradients of a specific iteration as a function of the route flow gradients of the previous iteration. In the IB algorithm, we define this step as gradient backpropagation. The TA conditions and the gradient formulations shown in equations (4)-(8) can be extended to link based formulation for a static TA and BPR based link performance function by adding equations (9) - (10) to the formulation.

$$x_l = \sum_{od \in OD} \sum_{r \in R_{od}} f_{r,od}\, \delta_{r,l} \quad (9)$$

$$\nabla_\theta x_l = \sum_{od \in OD} \sum_{r \in R_{od}} \nabla_\theta f_{r,od} \delta_{r,l} \quad (10)$$

The flow and flow gradient update rules in equations (6) - (7) are written in the link level as equations (11) - (12).

$$x_l^I = x_l^{I-1} + \alpha \times \left[\sum_{od \in OD} \sum_{r \in R_{od}} q_{od} w_{r,od}\left(E_{od}(X^{I-1})\right)\delta_{r,l} - x_l^{I-1}\right] \quad (11)$$

$$\nabla_\theta x_l^* = \nabla_\theta x_l^I = \nabla_\theta x_l^{I-1} + \alpha \times \left[\sum_{od \in OD} \sum_{r \in R_{od}} \nabla_\theta q_{od} w_{r,od}\left(E_{od}(X^{I-1})\right)\delta_{r,l} - \nabla_\theta x_l^{I-1}\right] \quad (12)$$

We now formally define the IB algorithm, as shown in the following pseudocode.

Algorithm 1: IB Pseudocode

Step 1: Initialize the flow and gradient components ($f_{r,od}/x_l = 0$, $\frac{\partial f_{r,od}}{\partial \theta} = \mathbb{0}$; $\forall r \in R, od \in OD$). Set counter $i = 0$.

Step 2: Perform the network loading based on the current route flow F^i, and calculate the approximate route flow $f'_{r,od} = q_{od} w_{r,od}\left(E_{od}(F^i)\right)$ or in case of the link-based formulation, link flow $x'_l = \sum_{od \in OD} \sum_{r \in R_{od}} q_{od} w_{r,od}\left(E_{od}(X^i)\right)\delta_{r,l}$.

Step 3: Perform the gradient backpropagation using equations (7) - (8) and calculate approximate route flow gradients $\nabla_\theta f'_{r,od} = \nabla_\theta\left[q_{od} w_{r,od}\left(E_{od}(F^i)\right)\right]$, or in case of link based formulation, link flow gradients $\nabla_{\theta} x'_l = \sum_{od \in OD} \sum_{r \in R_{od}} \nabla_{\theta}\left[q_{od} w_{r,od}\left(E_{od}(X^i)\right)\right] \times \delta_{r,l}$.

Step 4: Update the route flow using equation (5) or link flow using equation (11).

Step 5: Update the route flow gradient using equation (6) or link flow gradient using equation (12).

Step 6: Go back to step 2 until the TA algorithm converges.

Step 7: Stop.

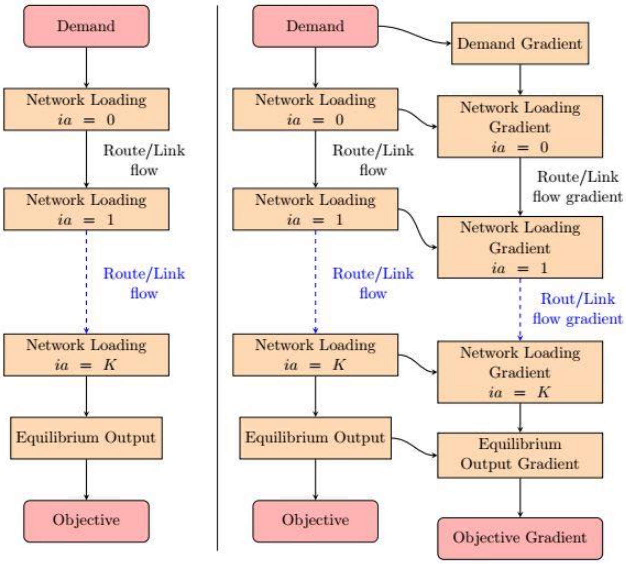

Figure 1: Traditional TA solution algorithm (left) versus IB algorithm (right)

IB does not require solving any system of equations. It directly calculates the gradient by substituting the required information from the network loading step into equations (6) - (12). These equations can be coded in a matrix form, enabling vector computation instruction set of modern computing hardware to calculate the maximum number of gradients in parallel. The original TA solution remains usable as IB does not affect the original TA equilibrium convergence. Figure 1 shows the comparison between traditional iterative TA solution algorithms and IB.

In the above formulation, besides calculating flow gradients, IB implicitly calculates the gradients of additional TA components, such as choice model and flow model gradient, while calculating $\nabla_{E_{od}} w_{r,od}\left(E_{od}(F^{I-1})\right)$ and $\nabla_F \eta_{r,od}(F^{I-1})$ in equations (7) and (8). Moreover, the gradients of other non-essential components, such as smart card entry or exit counts, tolls if any, can be derived from the route flow gradients. This makes IB versatile and a powerful tool to compute general TA equilibrium output gradients $\frac{\partial \psi_*(\theta^j)}{\partial \theta}$, as discussed in the previous section.

3. APPLICATION OF IB

In this section, we demonstrate the IB gradient computation with respect to OD demand in a small network depicted in Figure 2. The network has two links L_1 and L_2, both with a free flow travel time of 10 and capacities of 50 and 70, respectively. These two links connect 1 OD pair with demand q. We use the BPR function as the link performance function. The objective is to compute the gradient of TA equilibrium output components, i.e., link/route flow gradient and route choice gradient with respect to the OD demand q at two levels: $q = 10$ and $q = 100$ using IB. We demonstrate IB in both SUE and UE TA problems. While calculating iteration specific route flows, SUE uses the logit model and UE uses the 'all or nothing' loading procedure in each iteration. We shall compare and validate the gradients calculated by IB with those by the finite difference technique.

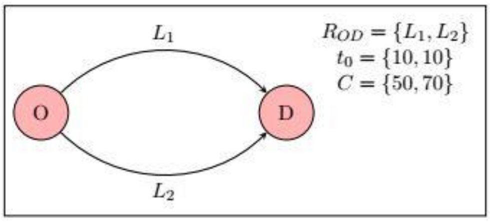

Figure 2: Two link network

Results and Discussion

Table 1: TA solution for the two-link network

	Demand	$x*/f*$	$w*$	$t*$
SUE	$q = 10$	9.933 0.067	0.9933 0.0067	10.002 15
	$q = 100$	65.629 34.371	0.6563 0.3437	14.453 15.099
UE	$q = 10$	10 0	1 0	10.002 15
	$q = 100$	67.8165 32.1835	0.6782 0.3218	15.0764 15.0764

Table 1 shows the TA solution for SUE and UE for both congested and uncongested scenarios. The gradients are all identical except $\frac{dt_2^*}{dq}$ at $q = 10$ where the IB gradient produces a value of 5.71E-13 whereas the FD produces a value of 0 (as shaded in Table 2). This happens because the supposed change in t_2 at $q = 10$ for a 1E-08 change in demand should have been 5.71E-21 which is below the machine precision of 1E-16 of MATLAB. So, the FD method missed the change and reported a gradient of 0.

Table 2: IB vs. FD TA equilibrium gradients for the two-link network

	Demand	$\nabla_q x*/f*$	$FD\ \nabla_q x*/f*$	$\nabla_q w*$	$FD\ \nabla_q w*$	$\nabla_q t*$	$FD\ \nabla_q t*$
SUE	$q = 10$	0.9933 0.0067	0.9933 0.0067	-6.23E-06 6.23E-06	-6.23E-06 6.23E-06	9.35E-04 5.71E-13	9.35E-04 0.00E+00
	$q = 100$	0.1242 0.8758	0.1242 0.8758	-0.0053 0.0053	-0.0053 0.0053	0.0338 0.0101	0.0338 0.0101
UE	$q = 10$	1 0	1 0	0 0	0 0	9.60E-04 0.00E+00	9.60E-04 0
	$q = 100$	0.6782 0.3218	0.6782 0.3218	0 0	0 0	0.2031 0.0031	0.2031 0.0031

Computation Speed Comparison for High Dimensional Case

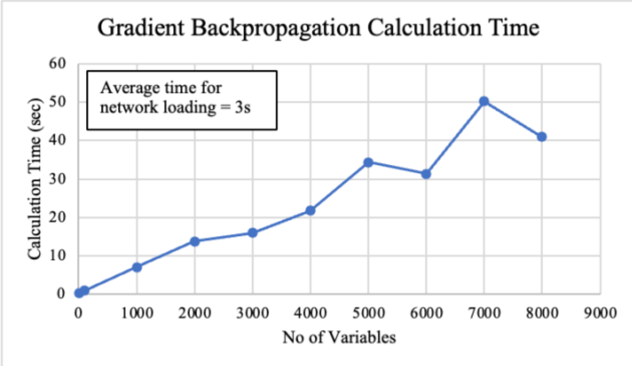

Figure 3: Gradient backpropagation calculation time (HK Network)

Figure 3 shows the computation time for IB paired TA network loading instances in the Hong Kong multimodal network against the problem dimension. Results indicate that for a TA assignment with 3s of network loading time, one IB gradient backpropagation step takes at most 50 sec for a problem dimension of 7000. Given FD requires $n + 1$ function evaluation, it should take $(7000 + 1) \times 3 = 21003$ sec. So, the IB gradient estimation technique is approximately $21003/50 \approx 420$ times faster than FD for TA gradient estimation.

4. CONCLUSIONS

This paper proposed an efficient gradient estimation framework, referred to as iterative backpropagation (IB), for high dimensional TA embedded optimization problem. It inserts a gradient backpropagation step along with the traditional network loading step of the TA solution algorithm for calculating the gradients. IB does not require solving for any system of equations or additional function evaluation irrespective of the problem dimension. We tested out IB for a small two links network where IB matched finite difference gradients at machine precision. Numerical analysis showed that in a large multimodal network and high dimensional problem, IB was 420 times faster than traditional FD gradient estimation technique. The framework is generic and applicable to many high dimensional TA embedded optimization problems.

ACKNOWLEDGMENT

The study is supported by HSBC 150th Anniversary Charity Programme HKBF17RG01, National Science Foundation of China (No. 71890970, No. 71890974), General Research Fund (No. 16212819, No. 16207920) of the HKSAR Government, and the Hong Kong PhD Fellowship.

REFERENCES

Chiappone, S., Giuffrè, O., Granà, A., Mauro, R., & Sferlazza, A. (2016). Traffic simulation models calibration using speed-density relationship: An automated procedure based on genetic algorithm. *Expert Systems with Applications*, *44*, 147–155. https://doi.org/10.1016/j.eswa.2015.09.024

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural Networks for Machine Learning. In *Human and Machine Hearing* (pp. 419–440). Cambridge University Press. https://doi.org/10.1017/9781139051699.031

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Lee, E., Patwary, A. U. Z., Huang, W., & Lo, H. K. (2020). Transit interchange discount optimization using an agent-based simulation model. *9th International Workshop on Agent-Based Mobility, Traffic and Transportation Models, Methodologies and Applications*.

Liu, B. Z., Ge, Y. E., Cao, K., Jiang, X., Meng, L., Liu, D., & Gao, Y. (2017). Optimizing a desirable fare structure for a bus-subway corridor. *PLoS ONE*, *12*(10), 1–21. https://doi.org/10.1371/journal.pone.0184815

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.

Oh, S., Seshadri, R., Azevedo, C. L., & Ben-Akiva, M. E. (2019). Demand Calibration of Multimodal Microscopic Traffic Simulation using Weighted Discrete SPSA. *Transportation Research Record*, *2673*(5), 503–514. https://doi.org/10.1177/0361198119842107

Otković, I. I., Tollazzi, T., Šraml, M., Ištoka Otković, I., Tollazzi, T., & Šraml, M. (2013). Calibration of microsimulation traffic model using neural network approach. *Expert Systems with Applications*, *40*(15), 5965–5974. https://doi.org/10.1016/j.eswa.2013.05.003

Patriksson, M. (2004). Sensitivity analysis of traffic equilibria. *Transportation Science*, *38*(3), 258–281. https://doi.org/10.1287/trsc.1030.0043

Patwary, A. U. Z., Huang, W., & Lo, H. K. (2021). Metamodel-based calibration of large-scale multimodal microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies*, *124*, 102859. https://doi.org/10.1016/j.trc.2020.102859

Spall, J. C. (1992). Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*. https://doi.org/10.1109/9.119632

Spiess, H. (1990). A gradient approach for the O-D matrix adjustment problem. *Centre for Research on Transportation, University of Montreal, Canada*, *693*, 1–11.

Tobin, R. L., & Friesz, T. L. (1988). Sensitivity analysis for equilibrium network flow. *Transportation Science*, *22*(4), 242–250. https://doi.org/10.1287/trsc.22.4.242

Wardrop, J. G. (1952). *bY*.

Yang, H., & Bell, M. G. H. (1997). Traffic restraint, road pricing and network equilibrium. *Transportation Research Part B: Methodological*, *31*(4 PART B), 303–314. https://doi.org/10.1016/s0191-2615(96)00030-6

Yang, H., Meng, Q., & Bell, M. G. H. (2001). Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium. *Transportation Science*. https://doi.org/10.1287/trsc.35.2.107.10133

Zhang, C., Osorio, C., & Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B: Methodological*, *97*, 214–239.