An open-source framework for the robust calibration of large-scale traffic simulation models

Vishal Mahajan[1], Guido Cantelmo[2], and Constantinos Antoniou[1]

[1]Technical University of Munich, Arcisstraße 21, Munich 80333, Germany
[2]Transport Division, Technical University of Denmark, Denmark

Short summary

Large-scale traffic simulation models are a crucial tool for simulating and evaluating different transport solutions. However, due to the scale and complexity of these models, numerous parameters exist that can significantly influence their outputs. The problem of estimating these parameters is referred to as the Dynamic Traffic Assignment (DTA) calibration problem. After more than 30 years of research, several algorithms have been proposed that can - with a certain degree of success - address this challenge, even for large instances or in the presence of noisy data. Two challenges, however, remain critical today and are addressed in this paper. From a purely methodological perspective, DTA calibration is a highly under-determined problem, meaning multiple plausible solutions exist. This is particularly relevant when calibrating demand parameters. Therefore, in this paper, we propose two techniques inspired by the field of computer science that allow for enhancing robustness: *bagging* and *Stochastic Parameter Averaging* (or *SPA*). The second contribution of this research is more practical. While many algorithms have been proposed, the source codes of these algorithms are often not shared with the scientific community. As a consequence, most papers still use as a benchmark model the SPSA, an algorithm proposed roughly 30 years ago. Therefore, this study introduces an end-to-end open-source framework for DTA calibration. The model can calibrate supply and demand parameters, include state-of-the-art optimizers (*W-SPSA*, *SPSA*, *Bayesian Optimization*), an auto-tuning option to calibrate their parameters, and the bagging/SPA extension already mentioned. The conceptual framework proposed in this research is general and includes a few algorithms already. It is currently linked with the open traffic simulator *SUMO* to demonstrate its effectiveness. Researchers can use this framework as a benchmark or extend it using new simulators and optimizers. The method is tested both in controlled settings, as well as using the real-world large scale network of Munich.
Keywords: DTA calibration, ensemble, large-scale optimization, OD estimation, open-source (Topics: Transport Network Modelling, Big data analytics, Operations research applications)

1 Introduction

A transportation system is made up of different parts and their interactions, which results in travel demand and supply of transport services (Cascetta, 2001). Dynamic Traffic Assignment (DTA) simulators are advanced tools commonly used by researchers and practitioners to represent the traffic flow variations and behavioral choices in a large-scale network (Ben-Akiva et al., 2012). Due to the scale and complexity of these models, numerous parameters exist that can significantly influence their outputs. DTA calibration is the process of estimating the value of these parameters so that the difference between the simulated data (counts, travel time, speed) and observed data is minimized. The resulting optimization problem, however, is notoriously complex to solve. One issue is whether to calibrate supply and demand parameters together or separately (Toledo et al., 2014). Second, it is virtually impossible to guarantee optimal solutions in real-life settings. That is because the problem is highly under-determined (due to a large number of variables compared to a relatively small amount of observations), highly nonlinear (due to congestion dynamics), and non-convex (non-unique optimal solution). For an extensive review of these issues, as well as their solutions, we refer to (Antoniou et al., 2016). In general, many algorithms have been proposed that can partially cope with the above-mentioned challenges. However, two aspects remain critical and are addressed in this study. First, many models proposed in the literature are difficult to reproduce. This is because the code is not publicly available, often due to restrictions in data availability or in the DTA model (e.g., commercial software). As a consequence, many authors benchmark their models against the SPSA (an algorithm proposed in 2006) or spend considerable time reproducing algorithms from the literature. In addition, while the DTA calibration process has been tested in many real-life experiments, due to the non-convex nature of the problem, obtaining robust and reliable estimates is still an open challenge. The work proposed in this research aims at answering these two questions. First, we propose an end-to-end open-source framework for DTA calibration that includes state-of-the-art solvers. The framework is designed for the open source DTA model *SUMO* (Lopez et al., 2018) and can be used to benchmark new models. Second, we propose to use parameter ensembling techniques, most notably *bagging* and *Stochastic Parameters Averaging* (*SPA*) to obtain more robust estimates. The framework has been successfully tested on the large-scale network of Munich, showing that it can be deployed in practice.

2 Methodology

The methodology is divided into two parts. First, we introduce the framework for DTA calibration and its main features. Then, we introduce the algorithms for SPA and bagging.

End-to-end calibration framework

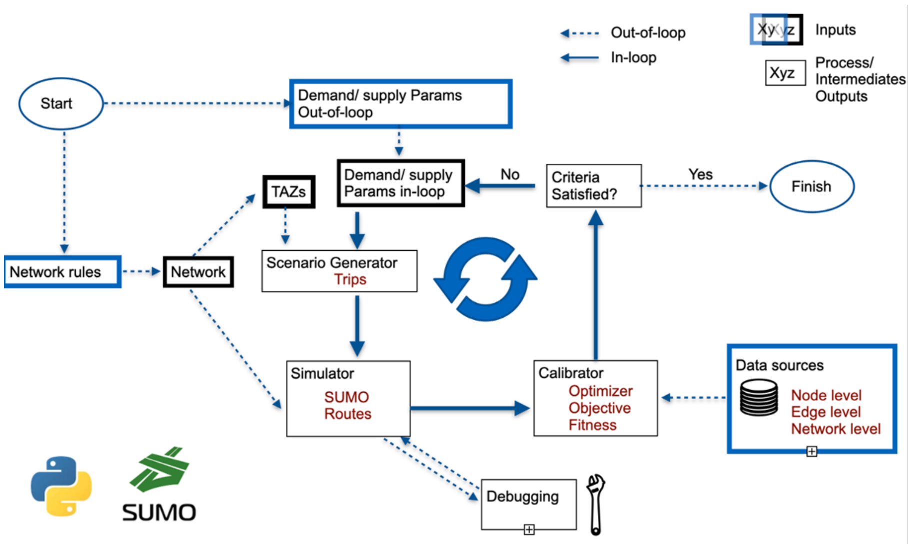

Figure 1: Experimental setup (Mahajan et al., 2023)

We developed a Python-based platform for the calibration of both demand and supply parameters of DTA models using the SUMO traffic simulator (Lopez et al., 2018). A schematic representation of the platform is shown in Figure 1. Given the simulation inputs (simulation network, traffic analysis zones, parameters), the platform estimates the parameters according to the proposed methodology. An initial Origin-Destination (OD) matrix generates trips between edges in different Traffic Analysis Zones (TAZs). The routing algorithm in SUMO assigns routes to these trips. Automatic or online routing is used for the traffic assignment. The parameters influencing the routing of vehicles are *re-routing probability*, *re-routing period*, and *re-routing adaptation steps*. Travel time of different edges can be scaled using the parameter *edge priority factor*. The parameters which affect the delays are *junctions flow penalty* at junctions and *unsignalized junction penalty*. We use Bayesian optimization for calibrating these selected supply parameters.

Concerning the demand parameters, we implemented a state-of-the-art optimizer (W-SPSA Antoniou et al. (2015)) by extending the Python SPSA implementation by Mayer (2017). Further, considerable time and manual effort is usually spent in fine-tuning the hyperparameters of the SPSA to enhance calibration performance. Therefore, we included in the framework an automatic tuning function to automatically optimize the SPSA hyper-parameters. The auto-tuning function uses the assignment matrix to create an analytical approximation of SUMO. Bayesian optimization is used to find the optimal parameters that reduce the error of the analytical model.
Finally, two additional extensions have been included, namely bagging and SPA. These methods, shortly described in the next sub-section, have been tested using W-SPSA. However, similarly to

most of the features described in this section, they can be combined with other optimizers once they are implemented in the framework. The complete platform is implemented using Python and is available on GitHub (https://github.com/vishalmhjn/actrys). In short, the model currently implements the following features:

- Three optimizers (Bayesian Optimization, SPSA, W-SPSA)
- A heuristic model to pre-process historical demand and remove bias
- The framework sequentially optimizes supply and demand parameters
- It includes an auto-tune function for the hyper-parameters of the optimizer
- It includes parameter ensembling to enhance robustness

Parameter ensembling to reduce estimators' variance

This sub-section introduces the methodological contribution of this research. The methodology focuses only on improving the estimates for the demand parameters, therefore we refer to this sub-task as the *OD estimation* problem. Without loss of generality, the OD estimation problem can be operationalized as follows:

$$\underset{\boldsymbol{X}}{\text{minimize}} \sum_{t=1}^{T} \left[w_1 z_1 \left(\boldsymbol{M_t^o}, \boldsymbol{M_t^s} \right) + w_2 z_2 \left(\boldsymbol{X_t}, \boldsymbol{X_t^a} \right) \right] \tag{1}$$

In Equation 1, $\boldsymbol{M_t^o}$ and $\boldsymbol{M_t^s}$ refer to the observed and simulated traffic data. Similarly $\boldsymbol{X_t}$ and $\boldsymbol{X_t^a}$ refer to the estimated and a-priori values of the demand (OD) parameters. Finally z_1, and z_2 are functions that measure the discrepancy between simulated and observed data (Goodness-of-Fit, GoF), while w_1 and w_2 are weights for these functions. The dependence between simulated traffic data and the OD matrices is directly obtained from the DTA traffic simulator (SUMO, in this study). Other constraints that are often applied in practice are ignored here for ease of reading. The OD estimation problem is highly under-determined, meaning that many solutions exist that are theoretically feasible for a given optimization formulation. This also implies that even state-of-the-art models such as W-SPSA will unavoidably find a local solution for Equation 1, resulting in parameters with considerable variance. In this study, we hypothesize that, due to variance in the spatiotemporal demand patterns, variance in sampling distribution or measurement errors can be considered as a manifestation of the desired (or "true") solution. Parameter averaging, such as in the bagging and averaging techniques, can help to cancel out some of the variances in the individual solution so that the averaged solution is closer to the desired one.

Therefore, we introduce two algorithms that can be used to combine bagging and SPA within the OD estimation problem.

W-SPSA with Bagging: when using bagging (*B-W-SPSA*) (Breiman, 1996), we run multiple estimators, such as W-SPSA, (in parallel or in serial order), and record the final estimates of each of the runs or cycles (since SPSA is stochastic in nature). To promote exploration, at each run we perturb the initial seed matrix - i.e., $\hat{X}^a \leftarrow X^a + \epsilon$, where ϵ is a normally distributed error term. Each run leads to different local optima. The final result is obtained as an average of these results

W-SPSA with Stochastic Parameter Averaging: The Stochastic Parameter Averaging (SPA) is a new algorithm applied in this research and inspired by the Stochastic Weight Averaging (SWA) (Izmailov et al., 2018), used in the field of computer science to find the weights of Deep Neural Networks (DNNs) while avoiding local minima. In SPA, the optimization is divided into two parts. In the first phase, the optimizer (W-SPSA, in this case) reduces the error in Equation 1. In this case, there is no difference between normal W-SPSA and W-SPSA with SPA. Phase two begins once the model achieves a local solution. In this step, the gain coefficients (i.e, the SPSA hyper-parameters) are reset. The next optimization cycle uses the iterate from the previous cycle as the initial parameters, and hence it is referred to as "warm restart". Resetting of SPSA gain coefficients resembles the cyclic learning rate, and allows the algorithms to explore new solutions.

3 Results and discussion

We present the results of our optimization model for two case studies. We compare the results of three models, namely *W-SPSA*, *B-W-SPSA*, and *SPA*. Two scenarios are analyzed:

1. **Scenario 1**: Analytical simulator with synthetic sensor counts. A randomly generated assignment matrix is used for mapping OD flows (randomly sampled using a distribution function).
2. **Scenario 2**: SUMO and Munich regional network with synthetic sensor counts data. Given OD flows (Moeckel et al., 2020) are simulated and corresponding sensor counts are recorded as desired counts. The Munich regional network is divided into 73 zones resulting in 5256 OD pairs. The network consists of a total of 8761 links.

Scenario one serves to visually explain how bagging and SPA build on and improve the performances of W-SPSA. Scenario 2 tests how parameter ensembling performs when using SUMO on a large, real network. In both cases, the Weighted Average Percentage Error (WAPE) and the Root Mean Squared Error (RMSE) are used as evaluation criteria for OD fitness and count fitness.
In **Scenario 1**, we use a random perturbation R^x to introduce bias in the OD matrix. Then, we use the proposed algorithms to estimate the optimal parameters. The objective function minimizes the error with respect to historical OD flows and traffic counts. In Figure 2, we show the contours of the OD fitness errors for single W-SPSA estimates, SPA estimates, and bagged estimates - for selected ODs. Due to high dimensional optimization, fitness error is influenced by thousands of the demand parameters, so the plot shows the conditional error (because it depends on multiple parameters) region with the values of the pair of zones on X and Y-axes. The two columns in this figure correspond to levels of random perturbation R^x 30%, and 90%. It is evident, that in both cases, the single estimates are scattered in the region, but the averaged estimates from SPA and bagging are lying with the region of lower errors as compared to the single W-SPSA estimates. Intuitively, this illustrates how bagging/SPA help to reduce the variance in the estimates from single W-SPSA estimates. While the two models overall achieve the objective of reducing variance in the estimates, our results show that - for the same number of objective functions evaluation - bagging systematically outperforms SPA. Therefore, in Scenario 2 we will mostly focus on comparing bagging and traditional W-SPSA.

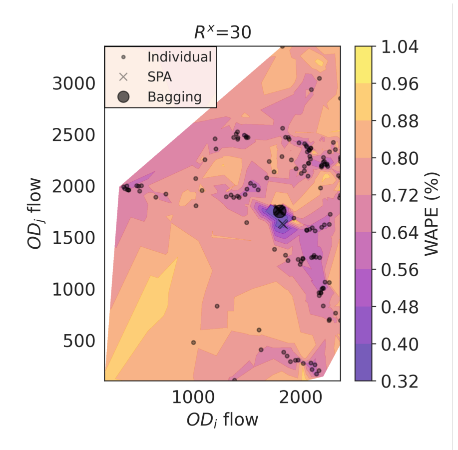

Figure 2: Contour plots showing the parameter values of the objective function for selected pair of the zones at different values of the R^x.

Scenario 2: We show the results of the calibration for the Munich scenario using the SUMO platform in Table 1. In all experiments, we assume a systematic bias $B^x = 0.60$ and a relatively smaller factor for randomness ($R^x = 20\%$). By default, we use only sensor counts in the objective function. However, we also simulate the case where we introduce artificial randomness in the sensor counts to mirror data errors. In this case, we also introduce speeds in the objective function to test the open-source framework when multiple data sources are available.
We use the W-SPSA with manual optimization of the hyper-parameters as the baseline. The corresponding improvement in speed fitness and OD fitness are 36.65% and -59.71%, respectively. A negative value of improvement tells that the estimated OD is worse than the initial OD values, which points to the ineffectiveness of the optimization. When using W-SPSA combined with bagging (*B-W-SPSA*), all error metrics substantially improve. The counts and speeds error improve by about 80% and 44%, respectively. More notably, the error in the OD flows is also reduced of about 20%. To further test our hypothesis that bagging can reduce variance by filtering noise, we can look at the experiments where some sensor noise was introduced. When the noise is low

Table 1: Results of the Munich scenario with synthetic data

		Sensor noise	**% Improvement WAPE (RMSE)**		
Model	B^x		Count	Speed	OD
W-SPSA	0.6	0	68.11 (69.01)	36.65 (50.27)	-59.71 (-82.84)
B-W-SPSA	0.6	0	**82.34 (84.31)**	**44.36 (55.72)**	**20.08 (12.88)**
B-W-SPSA*	0.6	15%	61.54 (59.53)	48.85 (65.22)	9.06 (-7.06)
B-W-SPSA*	0.6	30%	41.29 (39.06)	32.47 (33.56)	0.33 (-19.02)
B-W-SPSA*	0.6	45%	25.07 (23.79)	28.43 (14.43)	-0.38 (-38.48)

*: both counts and speeds are used in the objective function

(15%), the model still performs better than W-SPSA (except for the counts). When sensor noise increases, performances deteriorate. However, this is expected. More importantly, *B-W-SPSA* has a low error on the OD flows even for high sensor noise, which highlights the robustness of the framework when SPSA is combined with bagging.

4 Conclusions

Robust and efficient algorithms for DTA calibration are required for virtually any application where DTA models are deployed. However, obtaining robust estimates is still an open challenge in the research community. This research contributes to this direction in two ways. First, we provide an open-source framework for DTA calibration. The framework already includes several features, and can easily be extended to new DTA models and/or optimization algorithms. We hope that this can be useful to other researchers when developing new and better algorithms. Second, we propose two parameter ensembling techniques, the SPA and the bagging, and test them with the W-SPSA algorithm. Both techniques show that ensembling can lead to more robust estimates compared to W-SPSA. Conceptually, SPA and bagging are similar, as they both achieve better estimates by averaging results. However, SPA does this sequentially, while bagging runs independent optimizations. In practice, bagging can be more convenient when it is possible to run several simulations in parallel, while SPA can be more suited to explore different regions sequentially. We tested our framework on the large-scale network of Munich, Germany, showing how ensembling techniques allow filtering noise and estimating robust solutions. In the future, we hope to collaborate with other researchers, to include other algorithms, and interface the platform with other DTA models.

References

Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., ... others (2016). Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation. *Transportation Research Part C: Emerging Technologies*, *66*, 79–98.

Antoniou, C., Lima Azevedo, C., Lu, L., Pereira, F., & Ben-Akiva, M. (2015). W-spsa in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Part C: Emerging Technologies*, *59*, 129-146. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0968090X15001710` (Special Issue on International Symposium on Transportation and Traffic Theory) doi: https://doi.org/10.1016/j.trc.2015.04.030

Ben-Akiva, M. E., Gao, S., Wei, Z., & Wen, Y. (2012). A dynamic traffic assignment model for highly congested urban networks. *Transportation Research Part C: Emerging Technologies*, *24*, 62-82. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0968090X12000186` doi: https://doi.org/10.1016/j.trc.2012.02.006

Breiman, L. (1996, Aug 01). Bagging predictors. *Machine Learning*, *24*(2), 123-140. Retrieved from `https://doi.org/10.1007/BF00058655` doi: 10.1007/BF00058655

Cascetta, E. (2001). Transportation systems engineering: Theory and methods. Springer New York, NY. doi: https://doi.org/10.1007/978-1-4757-6873-2

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., & Wilson, A. G. (2018). *Averaging weights leads to wider optima and better generalization.* arXiv. Retrieved from `https://arxiv.org/abs/1803.05407` doi: 10.48550/ARXIV.1803.05407

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., ... Wießner, E. (2018, November). Microscopic traffic simulation using sumo. In *The 21st ieee international conference on intelligent transportation systems* (pp. 2575–2582). IEEE. Retrieved from `https://elib.dlr.de/127994/`

Mahajan, V., Cantelmo, G., & Constantinos, A. (2023). One-shot heuristic and ensembling for automated calibration of large-scale traffic simulations. *preprint*. Retrieved from `https://mediatum.ub.tum.de/node?id=1701188`

Mayer, A. (2017). Noisyopt: A python library for optimizing noisy functions. *Journal of Open Source Software*, *2*(13), 258. Retrieved from `https://doi.org/10.21105/joss.00258` doi: 10.21105/joss.00258

Moeckel, R., Kuehnel, N., Llorca, C., Moreno, A. T., & Rayaprolu, H. (2020, Feb 25). Agent-based simulation to improve policy sensitivity of trip-based models. *Journal of Advanced Transportation*, *2020*, 1902162. Retrieved from `https://doi.org/10.1155/2020/1902162` doi: 10.1155/2020/1902162

Toledo, T., Kolechkina, T., Wagner, P., Ciuffo, B., Lima Azevedo, C., Marzano, V., & Flötteröd, G. (2014, 09). Network model calibration studies. In (p. 22). CRC Press.