Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

MICROSCOPIC VEHICULAR TRAFFIC SIMULATION: TOWARD ONLINE CALIBRATION

Yulong Wang
John A. Miller

School of Computing
University of Georgia
Boyd Research and Education Center
Athens, GA 30602, USA

Casey Bowman

Department of Mathematics
University of North Georgia
82 College Circle
Dahlonega, GA 30597, USA

ABSTRACT

The modern world requires accurate and efficient traffic modeling to facilitate commerce and ensure citizens' safety. Traffic simulations play an important role in this endeavor by allowing traffic engineers to test traffic systems and policies before implementing them. This requires traffic simulation models that have the ability to accurately represent real-world traffic systems, and which are also capable of re-calibrating model parameters when needed through online calibration. This work presents four contributions toward this endeavor. The data science system SCALATION was extended with agent-based modeling and makes use of virtual threads for each vehicle, which improves the efficiency of simulations. The modeling, simulating, and data loading schema were all optimized to enhance the system performance as well. Additionally, a new arrival model strategy was implemented improving the accuracy of the model calibration phase.

1 INTRODUCTION

In the era of Big Data, simulation models may be utilized more like other forecasting models/systems. A forecasting system involves ongoing data collection to feed models both for making new forecasts and periodic retraining of the forecasting models themselves. The types of forecasting systems include statistical models (e.g., Auto-Regressive (AR) to Seasonal, Auto-Regressive, Integrated, Moving Average with Exogenous Variables (SARIMAX)), machine learning models (Full-Connected Neural Networks (FCNN), to Long Short-Term Memory (LSTM) models, to Sequence-to-Sequence (Seq2Seq) models, to Transformer models), as well as simulation models used for forecasting.

This approach to simulation is different from traditional Verification and Validation (V&V) that focuses on making a model that passes V&V and is ready for use. Forecasting models that are not fed with up-to-date data or have not been retrained to for a long time will become stale and their forecasts will degrade. The challenge in following this approach for simulation has been the lack of sufficient data and the computational cost of retraining (often called online calibration when applied to simulation models).

A key element of retraining (or online calibration) is efficient and effective optimization. Statistical models often use advanced nonlinear optimization algorithms such as Limited Memory, Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm, although AR may use the highly efficient Durbin-Levinson algorithm. Neural network models use advanced forms of Stochastic Gradient Descent (SGD). In simulation models, calibration involves setting several parameters that affect with way entities/agents in the model behave. The values of these parameters are adjusted to bring the simulation model into agreement with observational data by minimizing a loss function (e.g., Mean Square Error (MSE), Mean Absolute Error, or symmetric Mean Absolute Percentage Error (sMAPE). Given a parameter vector setting, forecast values must be compared to observed values. For statistical and neural network models, a forecasted value by computing a formula or forward propagation through a network. Unfortunately, making forecasts involves running one or more replications of a simulation model, which is substantially more time-consuming. Further complicating the situation is the inherent noisiness of simulation output. Algorithms such as L-BFGS are less likely to work because they require many functional evaluations per step and are confused by the noisiness of the response surface.

This paper describes the ongoing work in the SCALATION Project to develop a forecasting systemic approach for simulation modeling. SCALATION already includes several statistical and machine learning techniques for classification, prediction, and forecasting (Miller 2023). It also provides a large collection of optimization algorithms. In particular, this paper focuses on the following problem: forecasting vehicle traffic flow on highway systems. Such a problem is well suited to study because of the importance of the problem and the availability of big data collected at high frequency by road sensors. The Caltrans Performance Measurement System (PeMS) collects traffic flow, speed and occupancy data for each lane of major highways every five minutes and sensors are densely packed on highways.

Efficient traffic control systems are essential to the development of the modern world and thus require a great amount of analysis to inform their design and maintenance. Microscopic traffic simulation has a long history of aiding in these analyses and can be a powerful tool for learning about and optimizing traffic systems.

The rest of the paper is organized as follows: Section two presents the main goals as well as an overview of the project focusing on what needs to be done to support online calibration. Section three presents the arrival and driver models. The need for online calibration is discussed in section four. Related work previously done by this group as well as by other research projects is compared in section five. Section six details the additions and changes to SCALATION to support online calibration. The use of optimization in calibration is addressed in section seven. Section eight presents the experimental results that compare alternative means for providing online calibration. Finally, section nine gives the conclusions and future work.

2 GOALS AND PROJECT OVERVIEW

The goal of this ongoing project is to build an agent-based, discrete event, microscopic traffic simulation system within SCALATION (Miller et al. 2010) that can be used with online calibration and data assimilation to continually keep its model's parameters up to date for accurate traffic flow forecasting.

To accomplish this several crucial problems must be solved.

1. Selection of traffic/car-following model
2. Selection of calibration/optimization methodology
3. Extension of SCALATION to agent-based simulation
4. Extension of calibration methodology to online calibration
5. Efficient execution with Java virtual threads

The system will be capable of continuously executing a traffic simulation that can periodically check its accuracy and determine if a re-calibration procedure is necessary and if so, carry that process out with an appropriate calibration technique. The system will require efficient execution due to both the large number of modeled vehicles in a microscopic traffic network, and to the periodic high expense of re-calibrating. Java virtual threads show great promise as an efficient solution to these expensive computations.

3 ARRIVAL MODELS AND CAR FOLLOWING MODELS

The distribution of inter-arrival times can be recovered from the distribution of vehicle counts using the process of inversion (Cinlar 1975), which produces a Non-homogeneous Poisson Process (NHPP). The technique presented by Leemis (1991) is used in this work. Using the vehicle count data from the PeMS system at the first sensor in the modeled network, inter-arrival times are generated using the NHPP, which are then used by the simulation to produce new vehicles at the appropriate time points.

Once vehicles have been generated they must move around the network in a realistic way, which requires the use of a car-following model. Two models have been used and compared in this work, the Intelligent Driver Model (IDM) (Treiber et al. 2000), and Gipps' Model (Gipps 1981), which are both popular choices for car-following.

4 ONLINE CALIBRATION

Models of dynamic real-world systems with very noisy data not only require a thorough calibration process at inception but also require continued maintenance of the calibration since it is highly likely that model parameters will need to be changed over time. To this end, online calibration can be utilized to allow new data to keep the model as fresh and up-to-date as possible. This process requires fast and efficient procedures for deciding if re-calibration is needed, and if so, for carrying out the calibration itself. For this purpose, optimization algorithms that can execute rapidly, and with good accuracy are essential.

Hanisch et al. (2005) articulated a blueprint of how an online model could work where model parameters are constantly updated during the simulation. However, they did not focus on how the parameters of an online model should be calibrated by an optimization algorithm, or on how an arrival model affects this process. Antoniou et al. (2005) transform the calibration problem into a state-space model and use Kalman Filters to update estimates of the state-space as the simulation executes.

5 RELATED WORK

Researchers have formulated several designs and prototypes for online calibration frameworks. Henclewood et al. (2012) pointed out that an online, data-driven calibration algorithm needs to be used to accurately conduct time-sensitive traffic forecasting. However, their system was limited to off-the-shelf market software and a single optimization algorithm. Papathanasopoulou et al. (2016) proposed that driver behavior is quite heterogeneous, and therefore car-following models are highly susceptible to compounding errors quite rapidly. They proposed an online calibration technique for short time horizons using very recent data and re-calibrates parameters as the simulation executes. Biller et al. (2022) introduced an online learning framework for digital twin development where the model should take in real-time data from the physical one to optimize the digital one. The idea of a digital twin with online learning not only endorses the online calibration of a simulation model but also incorporates it into a bigger picture to conceptualize and redefine the mission and utilization of simulation and modeling with online calibration. Earlier work on this project (Bowman et al. 2022) began the design and implementation of the system, but required various improvements to continue including an agent-based model and efficient calibration.

6 EXTENSIONS TO SUPPORT ONLINE CALIBRATION

Figure 1 shows a framework for online calibration using our system. In order to make online calibration effective as well as efficient, certain critical components of the system had to be either implemented or enhanced, as discussed below.

6.1 Agent-Based Simulation

SCALATION supports three main simulation paradigms: event-scheduling, process-interaction, and agent-based. Due to the high time demands of simulation calibration, an efficient simulation paradigm is required. Of the three, the event-scheduling paradigm is the most efficient, but unfortunately, cars being the main entity in the simulation would have their logic fragmented into multiple event types that would make model development challenging and the code hard for others to follow. Although less efficient, process-interaction models will keep the code for a car together and can be understood as a script indicating the behavior of cars. Some of the logic is prescribed in the resources that the process (car) interacts with. So long as the interactions are fairly standard, this paradigm works well. Our previous paper on simulation

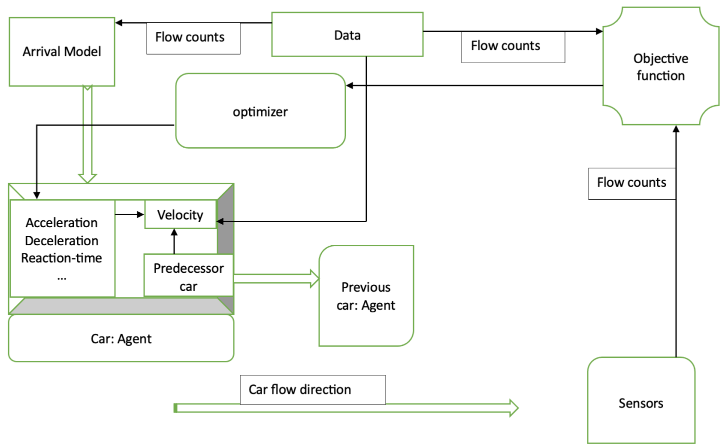


Figure 1: Pipeline of Calibration.

calibration used this paradigm. Once more detail and capability for cars are required, adding ad-hoc code to a process-interaction model becomes awkward and error-prone. Consequently, we have switched to agent-based simulation because it allows an agent (car) to have some knowledge about its environment and have capabilities for making non-trivial decisions, e.g., making safe lane changes. Safe lane changes require an agent (car) to know the location and speed of cars in its neighborhood, and using this information, decide whether to change lanes. Although the slowest of the three paradigms, the agent-based has the greatest capability of producing realistic simulation models. Furthermore, as SCALATION runs on the Java Virtual Machine (JVM), the recent introduction of Virtual Threads allows agents (each running as a Virtual Thread) to run more efficiently and allows more agents to be operating at the same time.

In general, there is Agent-Based Modeling (ABM) and a sub-category called Agent-Based Simulation (ABS) (Macal and North 2005; Onggo and Foramitti 2021). While ABM can be used in many domains to study, for example, emergent behavior, ABS is usually interested in applying simulation methodology to ABM to obtain results supporting the comparison of systems in terms of their efficiency and quality. SCALATION's ABS falls into the causal sub-category of ABS, where actions are ordered in time, i.e., temporal causality (Preddy and Nance 2002; Wagner and Nardin 2018) is enforced. This requires the agents to run on top of a parallel simulation engine or for the agents to be scheduled by a global scheduler. Although use a parallel simulation (Hybinette, Kraemer, Xiong, Matthews, and Ahmed 2006; Zhang and Rose 2012) would be preferred for faster execution (so long as the hardware is there to support it), SCALATION currently uses a global scheduler due to its simplicity.

6.2 Refactoring Simulation Model

Calibration of simulation model parameters requires many functional evaluations as part of an optimization algorithm. Each functional evaluation requires the simulation to be run. It is therefore imperative to streamline the runtime profile of the model as much as possible. The agent-based Vehicle Traffic Flow Model creates `Source`s, `Sink`s, `Highway`s, and `Junction`s over which `Car` agents travel. In addition, an Arrival submodel is created based on the collected Caltrans PeMS data. Rather than creating these anew for every functional evaluation, they are simply reset. Consequently, the highway structure indicated by their GPS and relation to each other was built only once and all the data including all the sensor's flow counts and average speed of each lane etc. were computed and processed before the simulation instantiation which is implemented by the Scala companion object. During each simulation and calibration, the arrival model would be called to predict the imminent simulating time window data to feed and run the traffic simulation model, and actual simulating could be repeatedly invoked to work on the existing infrastructure, substantially speeding up the calibration process.

6.3 Virtual Threads

As multiple agents can interact with each other at one time, to achieve concurrency, each agent could be carried by a thread to execute its task during the lifetime and scheduled to execute in a manner that maintains temporal causality. However, a thread is a heavy computational resource to allocate and utilize. With the emergence in the JVM of user-mode threads like Java virtual threads (or Kotlin coroutines), one agent could be carried by a virtual thread to benefit the performance where the number of live agents to be executed is high and agents could be generated, paused and terminated frequently during run time. There are multiple implementations of lightweight user threads across different programming languages (e.g., Simula-67 had coroutines). This work uses Java Virtual Threads supported by Java 20 released in 2023, which were first introduced as an experimental feature in Java 19 released in 2022, and allow several virtual threads to be mapped into one operating system thread. This greatly reduces the number of actual threads that need to be created and thereby speeds up and reduces the memory footprint of the simulation. In this work, each `Car` is carried by a virtual thread.

7 OPTIMIZATION FOR CALIBRATION

The calibration of model parameters ultimately requires the use of an optimization procedure, and there are many to choose from. Traditional gradient-based optimization is not practical for this problem because the cost of function evaluations is very high, and thus gradient-free methods are chosen as the techniques in this project. Each of the algorithms used carries out gradient-free optimization in a different way.

The optimization problem in this project is one of calibrating the parameters of two car-following models, Gipps' Model and the Intelligent Driver Model (IDM). Furthermore, as the parameters all belong to specific, restricted domains, it requires constrained optimization. These parameter domains are shown in Table 1.

Table 1: Model Parameters and Domains.

Definition	Symbol	Domains	Units	Definition	Symbol	Domains	Units
Acceleration	a	$[0.5, 10.0]$	m/sec^2	Time Headway	T	$[1.0, 5.0]$	sec
Deceleration	b	$[-10.0, -0.5]$	m/sec^2	Dist. Headway	s	$[3.0, 12.0]$	m
Reaction Time	τ	$[1.0, 3.0]$	sec	IDM Parameter	δ	$[3.0, 8.0]$	–

7.1 Simultaneous Perturbation Stochastic Approximation

Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall et al. 1992) is an example of a stochastic approximation method, and has become an often used optimization algorithm for calibrating simulation models (Rahman et al. 2020; Paz et al. 2012; Ma et al. 2007; Abdalhaq and Baker 2014; Paz et al. 2015; Hale et al. 2015; Lee and Ozbay 2009; Markou et al. 2019; Bowman et al. 2022). This is due to the small number of function evaluations required to approximate the real gradient. The original method requires three executions of the objective function regardless of the dimensionality of the input variable. To accommodate a constrained approach to this algorithm, coordinates of the parameter space that pass outside the domain are projected back into the domain.

There have been efforts in the past to improve the algorithm such as the Weighted-SPSA algorithm (Lu et al. 2015) which incorporates a weight matrix into the gradient estimation, which is claimed to reduce the approximation by a significant margin. Another variation of the algorithm is the Cluster-based SPSA algorithm (Tympakianaki et al. 2015) which divides the dimensionality of the gradient vector into clusters and only perturbs the variables in a cluster, leaving the remaining variables unperturbed. The clustered perturbations are then combined into a single gradient approximation later. The claim is that this may reduce bias in the gradient approximation, but does not require much more computational expense.

Figure 2: The Selected CA Highway Segment by Google Earth.

7.2 Genetic Algorithms

Genetic Algorithms have been used for a long time as a robust, derivative-free optimization procedure. As with SPSA, GAs are an example of a stochastic approximation method since random variables are used in the optimization process. However, unlike SPSA, GAs do not attempt to replicate or approximate a gradient-like optimization procedure. The GA written for this project was introduced by Bowman et al. (2022) and is specific to the optimization problem presented here. An elitist approach is used where the top n candidate solutions are kept and then used to produce new candidates via crossover and mutation. The new population is completed by producing some number m of new random candidates from the domain.

7.3 Nelder-Mead Simplex

The Nelder-Mead Simplex Algorithm (Nelder and Mead 1965) is an example of a direct-search method that does not require gradients to conduct the search for optima. A simplex of order $n+1$ is created and slowly transformed, where n is the dimensionality of the search space. The transformations gradually surround and shrink to an optimal point. Each vertex of the simplex could theoretically move outside of the domain of the search space, so any coordinates which have done so can be projected back into the domain.

8 EXPERIMENTAL RESULTS

8.1 Caltrans PeMS Data

This work selected a segment of northbound US Highway 101 in San Mateo, CA and there are five sensors along the road at which data was utilized. These five sensors were chosen because they produced high-fidelity data more often than many of the other sensors along that highway. Figure 2 shows the locations of these sensors (Google Earth - April 2022). To achieve a realistic traffic model, we first used the Caltrans PeMS sensors metadata including the latitude and longitude to build a road map structure and compute the distance between sensors using the SCALATION `LatLong` utility. For the car flows and speeds, the provided data from the sensors is divided into 5-minute intervals. In order to smooth the data, we have aggregated 5-minute intervals into 15-minute intervals. This vehicle flow data feeds the arrival model and produces the inter-arrival time distribution for the simulation source. Average speed data is used to define the vehicles' initial speeds in each lane in each specific time interval. Tuesday data was chosen because the flow is less likely to deviate as outliers, and the time frame for each day is 6:00 AM (first

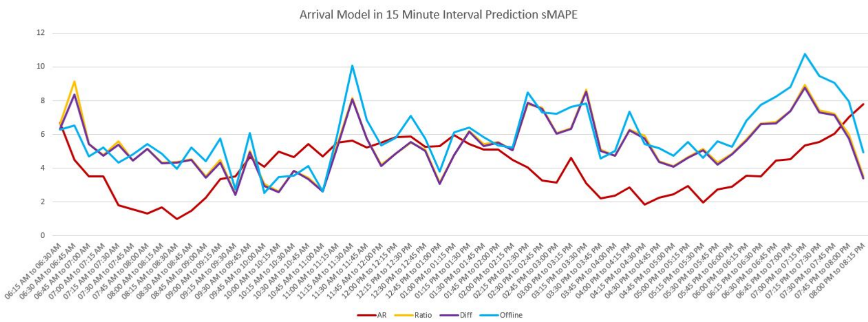


Figure 3: Arrival Model Comparison.

15 minutes for warmup) to 8:15 PM. A total of 13 Tuesdays in 2022 were used in this project. Arrival models use past data, like using the previous 4 weeks, to predict the arrivals on the current day, however, some data are missing in the required data range due to the malfunction of sensors, so to make the data consistent, the missing data were imputed using the SCALATION `ImputeNormalWin` utility which fills the missing values with the median of normal distribution for a window.

8.2 Arrival Model Results

Two NHPP arrival models, Ratio and Difference, were included for comparison with last year's work, which are based on simple rate of change calculations. A new model using an Auto-Regressive (AR) model was also brought in for comparison. The AR model (with $p = 1$) can reasonably predict the traffic flow in the next 15-minute interval based on the previous five data points. Meanwhile, the Ratio and Difference methods only sample the same time interval from the four previous Tuesdays. Nine weeks of Tuesday data were used for training and forecasting across the different arrival models. The average sMAPE is shown in Figure 3 where the vertical axis is the sMAPE value and the horizontal axis represents the specific 15-minute intervals. The AR arrival approach showed better overall performance than the other two throughout the day.

8.3 Calibration Results

The initial starting point was picked from last year due to its low sMAPE value. The two driver models are compared across three different optimization algorithms (with IDM across three arrival models and Gipps with one arrival model) shown in Table 2. To avoid a car's velocity going to a negative or extremely small value, which might lead to the simulation stalling, the lower bound of the velocity is set to 1 m/s. A

Table 2: The Initial Starting Point for Calibration.

Point	a	b	τ	T	s	δ
$\mathbf{x}$	7.0	-7.0	1.5	3.0	7.5	4.0

change to the calibration compared to previous work is the addition of domain constraints for SPSA and NM which seems to have improved their performance. The GA naturally had domain constraints all along.

8.3.1 Car-Following Models

The results using the IDM are shown in Figure 4. With the domain constraint method, the GA, on average, outperformed the other two algorithms across both car-following models and all three arrival models. Using the IDM, the best average was achieved using the Ratio arrival method, whereas for Gipps' Model the best average was achieved using the Difference arrival model. Gipps' Model saw better results with each of three optimization algorithms. The online calibration framework will consist of several core components:

an arrival model, simulation model, car-following model, and optimization algorithm. Every one of these components must run efficiently and accurately, and they must integrate well with each other. Averages over all the algorithms and models are shown in the Table 3

Table 3: sMAPE Average overall Algorithms and Models.

IDM	GA	NM	SPSA
AR	5.772820369	6.248262513	6.314001267
Ratio	**4.433951999**	**5.067624876**	5.089318188
Difference	4.490026273	5.095964155	**5.087141782**
Gipps	GA	NM	SPSA
AR	4.053604126	4.709941469	4.377964926
Ratio	**2.87092118**	3.836619474	3.696376744
Difference	2.907277724	**3.668139412**	**3.633063287**

Gipps' car-following model was brought in as a point of comparison with the IDM to check the variable of the driver model component of the controlled experiment. Gipps' model differs from the IDM in two important ways. First, Gipps' model can be implemented with fewer required parameters which can lead to a more efficient optimization process in the calibration phase of the system. Second, they differ greatly in their mathematical formulations, where the IDM computes a new acceleration for each vehicle, which is then used to compute new velocities and positions, Gipps' model directly computes the new velocity of a vehicle. Figure 5 and Table 3 show that as with the IDM, Gipps' Model performs better with the GA compared with NM and SPSA. On average, NM and SPSA performed very similarly with the IDM, whereas SPSA clearly outperforms NM with Gipps' Model.

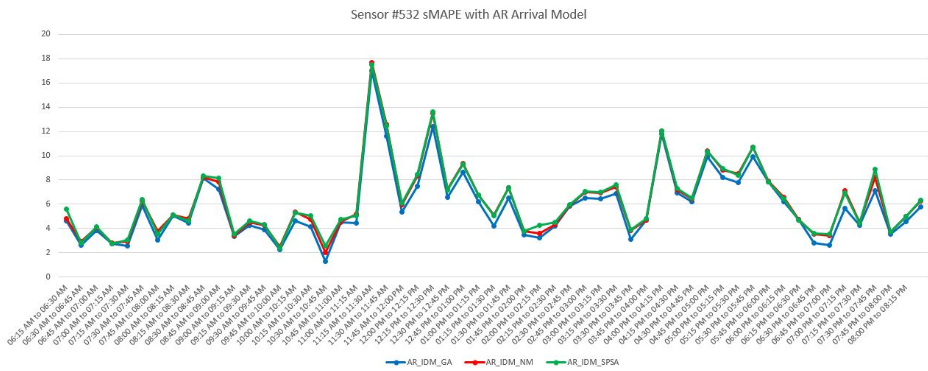
8.4 Execution Time Performance

The new system showed an improvement in execution efficiency by a factor of 3 to 5 times when compared to the earlier system. Factors contributing to this speed improvement are:

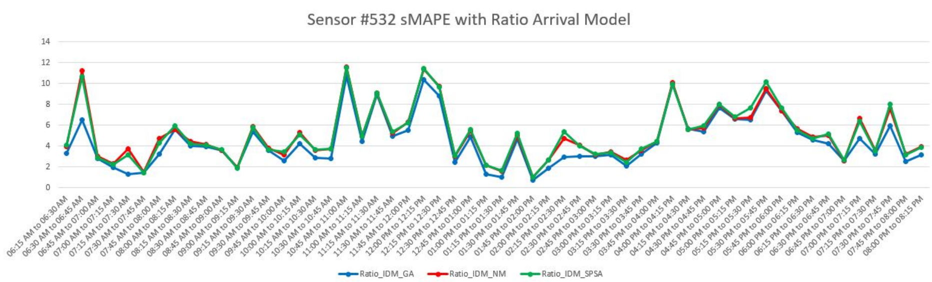
1. Refactoring the simulation: this includes avoiding the repetition of infrastructure built and destroyed in the old system, which costs the CPU computing and memory allocation and garbage collection; avoiding IO operations during the multiple calibrations and simulations; avoiding the same data processing and preparation for arrival and objective function repeatedly.
2. Virtual thread carriers: this schema replaced the last process interaction framework which relied on one main thread managing all others. The simulation has changed to an agent-based approach, which is powered by virtual threads, which are lighter-weight and require less resource allocation, which is easier and faster to execute.
3. Improved optimization: before the domain constraints were implemented, it was observed that SPSA or Nelder-Mead, for example, could take the deceleration parameter to an impossible value like -45 m/s, which will greatly strain the car-following model. The domains of the parameters of car-following models have been studied (Kurtc and Treiber 2016), and it is reasonable to restrict the optimization procedures to those domains.

Table 4: Execution Time of IDM.

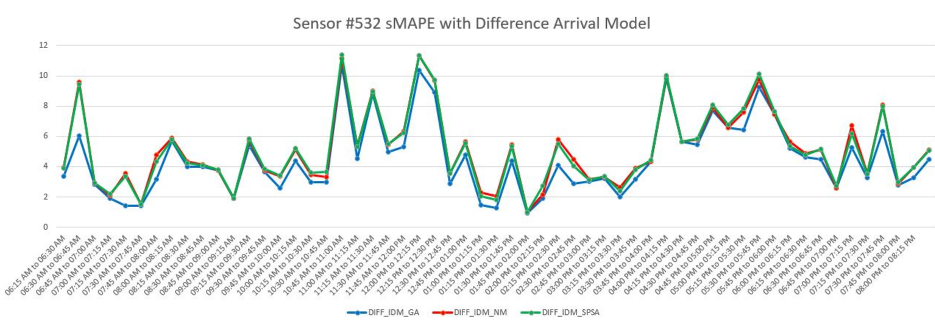
time (seconds)	GA	NM	SPSA
AR	21954	6753	7552
Ratio	22847	7132	6171
Difference	22684	5670	6160



(a)



(b)



(c)

Figure 4: sMAPE Scores for Sensor #532 (a) AR (b) Ratio (c) DIFF.

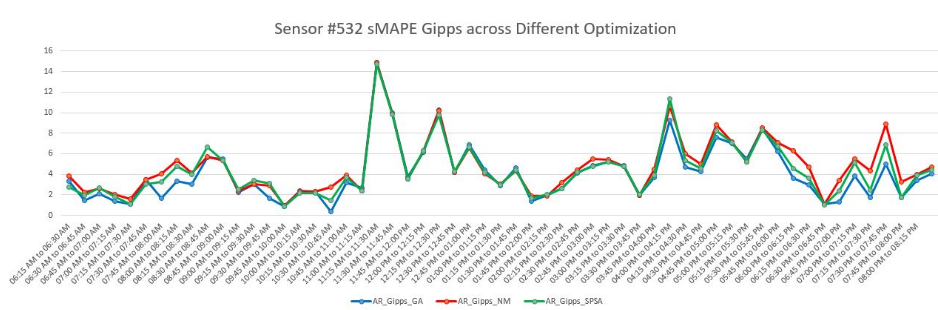


Figure 5: Gipps.

9 CONCLUSIONS AND FUTURE WORK

In this work, a new car-following model and a new arrival model have been integrated into the system allowing further investigation into the problem of traffic forecasting and online calibration. When comparing calibration results relative to arrival model the Ratio and Difference models outperformed the AR model across all three optimization algorithms and both car-following models. SCALATION is not limited to the algorithm or models used in this work, but it is very extendable and open for more models and optimization with its own well-founded self-implemented data science libraries. Hence it could provide the researchers and industrial more possibilities to develop and code for more modeling and analysis. The results show that the program performance has been improved compared to the previous work. It has provided a pathway toward online calibration. The optimization procedures have been improved with domain constraints added this year. Table 4 shows execution time comparisons with the IDM across optimization algorithms and arrival models. The GA takes significantly longer than the other algorithms, however the GA showed the best accuracy. Neither NM or SPSA seems to distinguish itself from the other in terms of execution time.

Traffic networks produce extremely noisy data which requires a Noisy Optimization (Soritz et al. 2022; Amaran et al. 2016) approach for calibration. A goal is to implement some of these additional optimization techniques and compare them to those already in use, as well as to continue to work on an augmented version of SPSA that has shown promise in trial experiments. Another direction toward improved efficiency is taking advantage of parallel computation where individual function evaluations are executed in their own threads in parallel. SPSA requires 3 (2 for gradient approximation and 1 for new point) function evaluations per iteration, while NM requires anywhere from 1 to $n+2=8$ function evaluations for the IDM and from 1 to 5 for Gipps' Model, and after the initial pool is constructed, the GA requires 19 (pop) $-$ 4 (top) = 15 function evaluations per iteration.

REFERENCES

Abdalhaq, B. K., and M. I. A. Baker. 2014. "Using Meta Heuristic Algorithms to Improve Traffic Simulation". *Journal of Algorithms* 2(4):110–128.

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury. 2016. "Simulation Optimization: A Review of Algorithms and Applications". *Annals of Operations Research* 240(1):351–380.

Antoniou, C., M. Ben-Akiva, and H. N. Koutsopoulos. 2005. "Online Calibration of Traffic Prediction Models". *Transportation research record* 1934(1):235–245.

Biller, B., X. Jiang, J. Yi, P. Venditti, and S. Biller. 2022. "Simulation: The Critical Technology in Digital Twin Development". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. Corlu, L. Lee, E. Chew, T. Roeder, and P. Lendermann, 1340–1355. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bowman, C., J. A. Miller, and Y. Wang. 2022. "Microscopic Vehicular Traffic Simulation: Comparison of Calibration Techniques". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C. Corlu, L. Lee, E. Chew, T. Roeder, and P. Lendermann, 2234–2245. Piscataway, New Jersey: IEEE: Institute of Electrical and Electronics Engineers, Inc.

Cinlar, E. 1975. *Introduction to Stochastic Processes*. Prentice-Hall. New Jersey. US.

Gipps, P. G. 1981. "A Behavioural Car-Following Model for Computer Simulation". *Transportation Research Part B: Methodological* 15(2):105–111.

Hale, D. K., C. Antoniou, M. Brackstone, D. Michalaka, A. T. Moreno, and K. Parikh. 2015. "Optimization-Based Assisted Calibration of Traffic Simulation Models". *Transportation Research Part C: Emerging Technologies* 55:100–115.

Hanisch, A., J. Tolujew, and T. Schulze. 2005. "Initialization of Online Simulation Models". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1795–1803. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Henclewood, D., W. Suh, M. Rodgers, M. Hunter, and R. Fujimoto. 2012. "A Case for Real-Time Calibration of Data-Driven Microscopic Traffic Simulation Tools". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1670–1681. Piscataway, New Jersey: IEEE: Institute of Electrical and Electronics Engineers, Inc.

Hybinette, M., E. Kraemer, Y. Xiong, G. Matthews, and J. Ahmed. 2006. "SASSY: A Design for a Scalable Agent-Based Simulation System using a Distributed Discrete Event Infrastructure". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 926–933. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kurtc, V., and M. Treiber. 2016. "Calibrating the Local and Platoon Dynamics of Car-Following Models on the Reconstructed NGSIM Data". In *Traffic and Granular Flow'15*, 515–522. Springer.

Lee, J.-B., and K. Ozbay. 2009. "New Calibration Methodology for Microscopic Traffic Simulation Using Enhanced Simultaneous Perturbation Stochastic Approximation Approach". *Transportation Research Record* 2124(1):233–240.

Leemis, L. M. 1991. "Nonparametric Estimation of the Cumulative Intensity Function for a Nonhomogeneous Poisson Process". *Management Science* 37(7):886–900.

Lu, L., Y. Xu, C. Antoniou, and M. Ben-Akiva. 2015. "An Enhanced SPSA Algorithm for the Calibration of Dynamic Traffic Assignment Models". *Transportation Research Part C: Emerging Technologies* 51:149–166.

Ma, J., H. Dong, and H. M. Zhang. 2007. "Calibration of Microsimulation With Heuristic Optimization Methods". *Transportation Research Record* 1999(1):208–217.

Macal, C. M., and M. J. North. 2005. "Tutorial on Agent-Based Modeling and Simulation". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2–15. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Markou, I., V. Papathanasopoulou, and C. Antoniou. 2019. "Dynamic Car–Following Model Calibration Using SPSA and ISRES Algorithms". *Periodica Polytechnica Transportation Engineering* 47(2):146–156.

Miller, J. A. 2023. "Introduction to Computational Data Science Using ScalaTion". Technical report, The University of Georgia.

Miller, J. A., J. Han, and M. Hybinette. 2010. "Using Domain Specific Language for Modeling and Simulation: Scalation as a Case Study". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yucesan, 741–752. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nelder, J. A., and R. Mead. 1965. "A Simplex Method for Function Minimization". *The Computer Journal* 7(4):308–313.

Onggo, B. S., and J. Foramitti. 2021. "Agent-Based Modeling and Simulation for Business and Management: A Review and Tutorial". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–15. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Papathanasopoulou, V., I. Markou, and C. Antoniou. 2016. "Online Calibration for Microscopic Traffic Simulation and Dynamic Multi-Step Prediction of Traffic Speed". *Transportation Research Part C: Emerging Technologies* 68:144–159.

Paz, A., V. Molano, and C. Gaviria. 2012. "Calibration of CORSIM Models Considering All Model Parameters Simultaneously". In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 1417–1422. IEEE.

Paz, A., V. Molano, E. Martinez, C. Gaviria, and C. Arteaga. 2015. "Calibration of Traffic Flow Models Using a Memetic Algorithm". *Transportation Research Part C: Emerging Technologies* 55:432–443.

Preddy, S. M., and R. E. Nance. 2002. "Key Requirements for CAVE Simulations". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M.

Charnes, Volume 1, 127–135. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rahman, M. Mijanoor, M. T. Ismail, and M. K. M. Ali. 2020. "Comparing the Calibration Methods for Intelligent Driver Model Using Beijing Data". *International Journal of Vehicle Systems Modelling and Testing* 14(4):215–231.

Soritz, S., D. Moser, and H. Gruber-Wölfler. 2022. "Comparison of Derivative-Free Algorithms for their Applicability in Self-Optimization of Chemical Processes". *Chemistry-Methods* 2(5):e202100091.

Spall, J. C. et al. 1992. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". *IEEE Transactions on Automatic Control* 37(3):332–341.

Treiber, M., A. Hennecke, and D. Helbing. 2000. "Congested traffic states in empirical observations and microscopic simulations". *Physical review E* 62(2):1805.

Tympakianaki, A., H. N. Koutsopoulos, and E. Jenelius. 2015. "c-SPSA: Cluster-Wise Simultaneous Perturbation Stochastic Approximation Algorithm and Its Application to Dynamic Origin–Destination Matrix Estimation". *Transportation Research Part C: Emerging Technologies* 55:231–245.

Wagner, G., and L. G. Nardin. 2018. "Adding Agent Concepts to Object Event Modeling and Simulation". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, S. J. A. Skoogh, and B. Johansson, 893–904. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhang, T., and O. Rose. 2012. "Developing an Agent-Oriented Parallel Simulator for Production Processes". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. Uhrmacher, 1–2. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

YULONG WANG is a Ph.D. student in Computer Science at the University of Georgia. His research interests focus on Computer Vision, Virtual Reality, and AI. His email address is yw98883@uga.edu

JOHN A. MILLER is a Professor of Computer Science at the University of Georgia. He has been the Associate Head for 6 years and is/has been Graduate Coordinator for 13 years. His research interests include Modeling Simulation, Web Services/Workflow, Database Systems, and Big Data/Data Science. Dr. Miller received a B.S. in Applied Mathematics from Northwestern University in 1980 and an M.S. and Ph.D. in Information and Computer Science from the Georgia Institute of Technology in 1982 and 1986, respectively. In his areas of interest, Dr. Miller has authored over 200 research papers. He has served as General/Program Chair for four international research conferences: ANSS, WSC, ICWS, and SCC, as well as one regional research conference ACM-SE. His email address is jamill@uga.edu.

CASEY BOWMAN is an Associate Professor at University of North Georgia in the Mathematics Department, beginning in 2007. He completed B.S. degrees in Mathematics and Computer Science in 2003, and an M.A. in Mathematics in 2005 from the University of Georgia. He received his Ph.D. in Computer Science from UGA in 2022. His research interests are microscopic traffic simulation, calibration of simulation models, agent-based simulation, and simulation optimization. His email address is casey.bowman@ung.edu.