SimMobility Short-Term

An Integrated Microscopic Mobility Simulator

Carlos Lima Azevedo, Neeraj Milind Deshmukh, Balakumar Marimuthu, Simon Oh, Katarzyna Marczuk, Harold Soh, Kakali Basak, Tomer Toledo, Li-Shiuan Peh, and Moshe E. Ben-Akiva

This paper presents the development of an integrated microscopic mobility simulator, SimMobility Short-Term (ST). The simulator is integrated because its models, inputs and outputs, simulated components, and code base are integrated within a multiscale agent- and activity-based simulation platform capable of simulating different spatiotemporal resolutions and accounting for different levels of travelers' decision making. The simulator is microscopic because both the demand (agents and its trips) and the supply (trip realization and movements on the network) are microscopic (i.e., modeled individually). Finally, the simulator has mobility because it copes with the multimodal nature of urban networks and the need for the flexible simulation of innovative transportation services, such as on-demand and smart mobility solutions. This paper follows previous publications that describe SimMobility's overall framework and models. SimMobility is an open-source, multiscale platform that considers land use, transportation, and mobility-sensitive behavioral models. SimMobility ST aims at simulating the high-resolution movement of agents (traffic, transit, pedestrians, and goods) and the operation of different mobility services and control and information systems. This paper presents the SimMobility ST modeling framework and system architecture and reports on its successful calibration for Singapore and its use in several scenarios of innovative mobility applications. The paper also shows how detailed performance measures from SimMobility ST can be integrated with a daily activity and mobility patterns simulator. Such integration is crucial to model accurately the effect of different technologies and service operations at the urban level, as the identity and preferences of simulated agents are maintained across temporal decision scales, ensuring the consistency and accuracy of simulated accessibility and performance measures of each scenario.

Microscopic traffic simulation applications are now part of the daily transportation planning and operation realities. To explore and evaluate complicated future transportation scenarios, one needs to conduct experiments. Physical experiments can be conducted to understand how the different options work together on a small scale. For the entire urban area, however, a simulation model is the only viable option. Microscopic traffic simulation models have been widely used to test different road network and intelligent transportation system solutions. They aim at replicating detailed vehicle motions and interactions by modeling agent decisions, such as route choice, accelerations, decelerations, and lane changes. These models are implemented as synchronous applications that update the kinematic parameters of each entity (driver–vehicle units, public transportation, management systems, and even pedestrians) at every simulation time step. Similar to other transportation simulators, the design of microscopic models is based on a demand and supply equilibrium representation. Traffic demand input is formulated either by defining it with respect to input flows and turning proportions at intersections or, for larger networks, with respect to origin–destination (O-D) matrices that will rely on route choice models for network assignment (*1*). An example of these simulators are Aimsun (*2*), Vissim (*3*), Q-Paramics (*4*), Transmodeler (*5*), ARTEMiS (*6*), CORSIM (*7*), DRACULA (*8*), HUTSIM (*9*), INTEGRATION (*10*), MITSIMLab (*11*), SUMO (*12*), and Cube Dynasim (*13*). The first four simulation tools belong to the short group of integrated platforms available for fast implementation and that have been successfully used in a variety of transportation projects, accounting for a share of more than 70% of practitioners' and researchers' preference (*14*).

In microscopic traffic simulation, the supply implementation relies on the specification of the network configuration, the traffic management algorithms, and the driving behavior model. From the initial models developed in the 1950s for car-following behavior (*15*), traffic microscopic simulation models now include multiple detailed behaviors and have reached a high level of maturity, not only among the research community but also with regular practitioners (*1*). For a comprehensive review of all driving behavior components used in simulation, the reader should refer to Barceló (*1*) and Hranac et al. (*16*).

Commercial simulation tools have devoted a large share of their new features to enhanced interfaces, visualizations, and, sometimes, calibration frameworks. At the same time, they have managed to integrate dedicated traffic control modules, public transportation, and pedestrian simulation (*2, 3*) into their core architecture. Conversely, the simulation research stream has proposed several innovative driving behavior models (*9, 17*), integration of communication technologies, or even emissions models (*18*). Some of these features can also be found in case studies using commercial software, but typically by means of coupling external modules to the main simulation tool, eventually compromising computational performance and interactive behaviors. Furthermore, several recent efforts in the research community can be found in the development of sophisticated activity-based modeling frameworks and their integration in

C. L. Azevedo, L.-S. Peh, and M. E. Ben-Akiva, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139. N. M. Deshmukh, B. Marimuthu, S. Oh, and K. Basak, Singapore-MIT Alliance for Research and Technology, 1 Create Way, 138602 Singapore. K. Marczuk and H. Soh, National University of Singapore, 21 Lower Kent Ridge Road, Singapore. T. Toledo, Technion–Israel Institute of Technology, Haifa, Israel. Corresponding author: S. Oh, simonoh.res@gmail.com.

Transportation Research Record: Journal of the Transportation Research Board, No. 2622, 2017, pp. 13–23.
http://dx.doi.org/10.3141/2622-02

simulation platforms that focus on individual's entire day activity pattern. Comprehensive agent-based modeling structures developed so far can be listed as TRANSIMS (*19*), MATSim (*20*), and FEATHERS (*21*). These applications model the multiple choices and behaviors of a single agent during a day and have been shown to better represent the interactions and dependencies of individual mobility. However, computational efficiency of such platforms has always been a major concern because they usually deal with the entire population of an area that is synthetically generated.

The above development streams have raised several challenges in the integration of complex mobility and transportation models within microscopic simulation. The next generation of simulators should include activity-based frameworks, integrated formulations with higher level, consideration of alternative modes such as on-demand mobility and autonomous vehicles, advanced and flexible driving behavior models, and the possibility to easily integrate innovative transportation services such as vehicle-to-vehicle communication or logistic services.

This paper presents SimMobility Short-Term (SimMobility ST), a new open-source microscopic mobility simulator integrated in a multilevel simulation platform. Designed with an agent-based framework, SimMobility ST aims at simulating the movement of agents (traffic, transit, pedestrians, and goods) and the decisions and operation of control centers within 1 day. It considers individual travel behavior in detail using an activity-based formulation. In the next two sections, both the modeling framework and the system architecture are presented. Its modular structure is described in more detail with a focus on its multiple components along with the recent applications that showcase its flexibility in the simulation of different and innovative mobility scenarios. Finally, the successful calibration process of the demand–supply parameters of SimMobility ST for the city of Singapore is described.

SimMobility SIMULATOR

Overall Framework

SimMobility is designed as three primary modules segmented according to time frames (*22*). The short-term model functions at the operational level; it simulates movement of agents at a microscopic granularity (within day) and is presented in more detail in the next section. The mid-term (day-to-day) simulator handles transportation demand for passengers and goods; it simulates agents' behavior, including their activity and travel patterns, and it shares several mobility decisions with the short-term level (e.g., with the route choice model) (*22*). The long-term (year-to-year) model captures land use and economic activity, with special emphasis on accessibility. It predicts the evolution of land use and property development and use, determines the associated life-cycle decisions of agents, and accounts for interactions among individuals and firms. The high-level design of SimMobility is shown in Figure 1.

SimMobility must therefore include all the key mobility-related decisions that people make in their everyday lives. These decisions may be personal decisions of households or the commercial decisions of firms (*23*). To support this level of representation, SimMobility is based on the concept of agent-based simulation or microsimulation. Representation of individuals as agents in the model is necessary to simulate how people will react in the future to new infrastructures, new technologies, innovations in system management, and policy changes.

The SimMobility framework is fully modular in the sense that each of the levels can run independently and only access the other levels when necessary. The key to multiscale integration in SimMobility is a single database model that is shared across all levels. Every agent exists and is recognized at all levels simultaneously, and information is used according to each level's needs; in this way, behaviors will remain consistent and, even if levels are run separately, the impacts from one level's model will be propagated to the others gracefully.

In previous work, these models have not been fully integrated. While there is limited interaction of outputs, there is no internal coherence. SimMobility is unique in that the same pool of agents is used across all time frames. For further details on SimMobility's overall framework and mid-term and long-term simulators, the reader is referred to Lu et al. (*22*) and Le et al. (*24*), respectively.

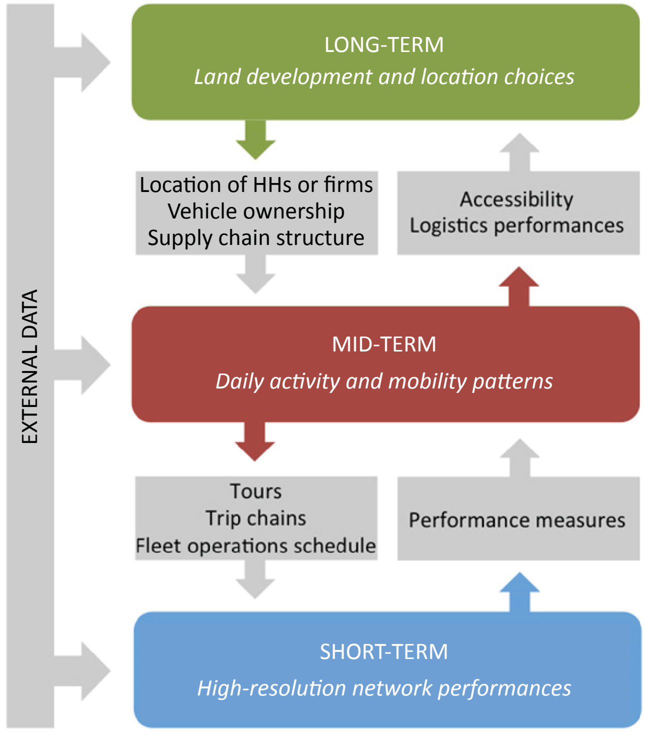
Short-Term Framework

SimMobility ST is an agent-based, multimodal microscopic simulator where agents' movements are captured at a very fine resolution (up to 100 ms). SimMobility ST comprises three main components. The microscopic movement module is responsible for advancing drivers, pedestrians, and goods on the transportation network according to their respective behavioral and decision models. The control and management module simulates the control centers, such as traffic and parking control, bus control, rail control, autonomous fleet control, and logistic control. The outcomes of these control actions will influence an agent's movement decisions, path choices, and other related decisions in the movement simulator. Within the control and management module, different control centers may be considered and replicated. At the current state of the simulator, the service controller and the traffic management controller are operational as described below. Ongoing efforts are being made in the development of the freight controller using detailed freight and logistics data (*25*). The third component is the communication network simulator, which simulates agent-to-agent communications. The information can be passed from one agent to another by the mobile communication or by vehicle-to-vehicle communication or maybe by vehicle-to-infrastructure communication. The communication network simulator is responsible for simulating the physical communication network (e.g., a wireless network), and agents simulated within the microscopic traffic network will use this simulated network to pass information between them. This process will help the agents to get the realistic communication network, which will handle the message delivery delay or coverage.

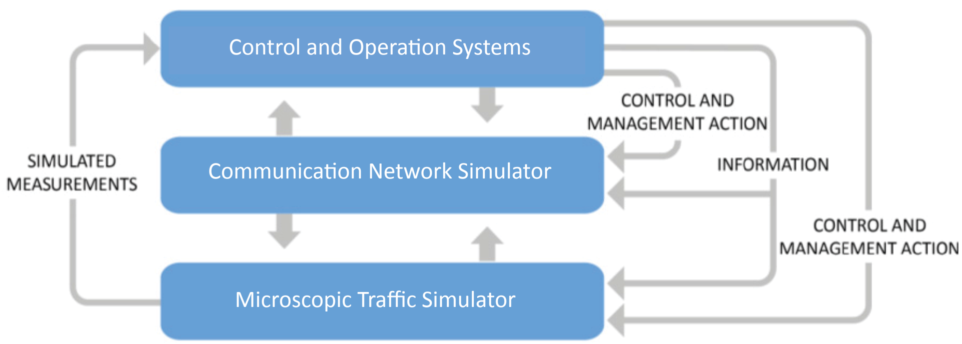
MODULES

Microscopic Traffic Simulator

The structure of the microscopic movement module is detailed in Figure 2. The virtual world is populated during the initialization phase, after which the simulation receives the control information and action plan at every time step. Two kinds of behaviors are simulated: high-level (travel) decisions, such as route choices, and lower-level (movement) decisions, such as car following and lane changing, which occur while the agent is in movement. While the agent's position is updated at every time step, the movement-related decisions only take place when specific events occur.



(a)



(b)

FIGURE 1 Framework of SimMobility and SimMobility ST: (*a*) high-level framework and (*b*) short-term framework (HH = household).

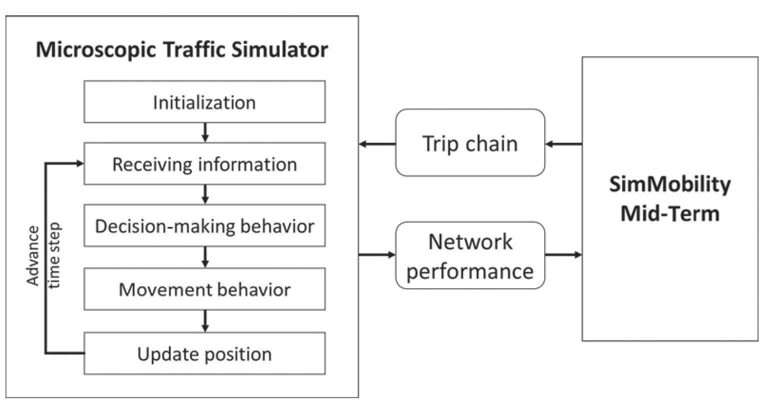


FIGURE 2 SimMobility ST traffic simulator.

Demand Input

Instead of the traditional O-D matrix definition used in the demand formulation of traffic microscopic simulation models, SimMobility ST (and its higher-level counterparts) uses an activity-based demand formulation in the form of activity schedules. In such an approach, trip chains are generated by individual daily schedules instead of aggregated traffic-specific matrices. Such data can be obtained directly using mobility and goods survey data or using a preday model, such as the one integrated in the SimMobility Mid-Term framework (*22*). The preday model consists of an activity-based modeling system formulated as interconnected discrete choice models representing choices at distinct dimensions. This preday model development follows the day activity schedule approach (*23, 26*), which focuses on decisions related to daily activity and mobility. There are three different hierarchies in the system: day pattern level, tour level, and intermediate stop level. Each level consists of several models, such as mode choice or departure tie choice. For the full specification of the preday in the SimMobility Mid-Term model, the reader is referred to Lu et al. (*22*). The output is an activity schedule and the trip chains for each agent in the simulation.

Within SimMobility ST, agents are then moved as per the planned trip chain. However, the realized trips can be changed during the simulation if specific circumstances, such as high congestion, incidents, public transportation interruptions, or any control or information provision, are observed.

For each agent's subtrip (a multimodal trip can have several subtrips) generated in the simulation, its role is assigned (pedestrian, passenger, and driver) and its role-specific characteristics are generated (e.g., aggressiveness, look-ahead distance, and reaction times). For each vehicle-based trip, an individual vehicle is generated. Ongoing work is being carried out to allow vehicle ownership and parking models to integrate a consistent vehicle generation model with unique identifiers. The generated vehicles are then assigned vehicle attributes (e.g., type and drive train) on the basis of configurable distributions.

Static Supply Input

The network in SimMobility ST is composed of a road network layer, a pedestrian network layer, and a public transportation layer. The road network layer is composed of (*a*) nodes, (*b*) links, with (*c*) segments, (*d*) polylines, and (*e*) lanes. Connectivity attributes are assured by (*a*) lane connectors, (*b*) turning groups, (*c*) turning paths, and (*d*) conflict points.

Nodes represent intersections or source and sink for trip chains. They are used for link definition in route choice and for the detailed characterization of intersections. Links are directional roads that connect nodes and are composed by segments. The latter are road sections with uniform geometric characteristics (speed limit, design speed, grade). Each segment is a fixed number of lanes, each with its specific lane rules (lane-changing regulation and use privilege). Polylines determine the shape of the segment and lane connectors define the connectivity between segments. At the node level, turning groups, turning paths, and conflict points can be defined. Turning paths connect specific lanes of two connected links, while turning groups and conflict points, respectively, define sets of turning paths connecting the same pair of links and overlapping points to two different turning paths. Additionally, the 10 road items are point-specific features that can be added to the network to represent items on the road to which drivers must respond (e.g., traffic lights and bus stops). Ongoing work is being carried out to extend this framework with parking infrastructure.

Driving Behavior

The core traffic model of SimMobility ST is based on MITSIM, an open-source microscopic traffic simulation application developed by the Massachusetts Institute of Technology. MITSIM moves vehicles according to route choice, acceleration, and lane-changing models. The acceleration model captures drivers' responses to neighboring conditions as a function of surrounding vehicles' motion parameters. The lane-changing model integrates mandatory and discretionary lane changes in a single model. Merging, drivers' responses to traffic signals, speed limits, incidents, and tollbooths are also captured. The driving behavior models implemented in MITSIM are those estimated by Ahmed (*27*) and Toledo et al. (*28*). The MITSIM lane-changing model was later enhanced by Choudhury (*17*), for the specific purpose of integrating latent plans in the lane selection process, namely in urban arterials and in freeways with a large number of lanes.

Several additional enhancements were made to the MITSIM original driving behavior: an enhanced reaction time formulation capable of explicitly modeling reaction time and perception delays for each person in a detailed and flexible manner as introduced [see Basak et al. (*29*) for further details]; lateral movement during lane change was also included. For the current implementation, the lateral speed is kept constant during the lane change, but the implementation of a sine function for lateral acceleration similar to the one proposed in Chovan et al. (*30*) has been initiated.

Finally, the design of a dedicated intersection behavior model, based on the conflicts technique, has also been implemented. The intersection behavior starts once the intersection is visible to the vehicle. The driver identifies the intersection regime (no rules, priority, or controlled). If the intersection is not controlled, the subject vehicle identifies the neighboring vehicles and the conflicting vehicles and proceeds with a gap acceptance–based model that accounts for intersection-specific priorities (if any).

Travel Behavior

Within SimMobility ST, changes in planned trip chains have to be considered. As the simulation is running, the agents need to find the routes for their trips and transform the activity schedule into effective decisions and execution plans. Agents may get involved in a multitude of decisions, not constrained to the planned set of destination, mode, path, and departure time, depending on the network and their state in the simulation cycle (*22*). In the current implementation, agents can reroute (as drivers or public transportation passengers) in the presence of congestion or the provision of control and information. Route choices are based on a probabilistic model that captures the impact of travel times and biases toward routes that use freeways over urban streets. The impact of real-time information on routing decisions is captured by a route-switching model in which informed drivers reevaluate their pretrip route choices on the basis of the traffic conditions observed en route. For perfect model integration with higher-level simulators, the route choice model used in SimMobility ST is the same as the one used in the SimMobility Mid-Term framework and its details can be found in Lu et al. (*22*).

Pedestrian Movements

The pedestrian behavior model focuses on the problem of how a pedestrian makes crossing-related decisions at different levels and at different points of times, when she or he walks along a given path to destination. Specifically, a crossing choice module is designed to determine where a pedestrian crosses the road, along a given path. A crossing timing decision module is designed to control when a pedestrian starts crossing the road, once she or he reaches a crossing point.

Commodity Movements

The movement of freight vehicles is typically considered in microscopic traffic simulation models by means of adapting driving behavior parameters for heavy vehicles (*31*) or by coupling dedicated external applications with the simulator (*32*). Teo et al. extended these traditional approaches by integrating the simulation of freight movements, logistics decisions, and traffic within an agent-based simulation (*33*). SimMobility ST is the first microscopic simulator that integrates commodity-specific movements with detailed traffic models. Similar to individual trip-chain input, SimMobility ST allows commodity-specific shipments. The commodity entity was specified for this purpose and freight drivers are assigned tours on the basis of the commodities to deliver during the simulation period. A default tour generation model was developed but this will be relaxed and linked to a freight operator controller, which typically represents a carrier. Freight vehicles and drivers will then be assigned to a specific freight operator and a set of delivery stops specified as road items in the network. Decisions on the freight vehicle tours can be made by the freight operator controller or the driver. The design and integration of all these entities within the core models of SimMobility ST are still under development at the stage of writing this document.

Control and Operation Systems

Traffic Management Controller

The traffic management controller mimics the traffic and information control system in the network under consideration. A wide range of traffic control and route guidance systems can be simulated. These systems include intersection controls, ramp control, freeway main-line control, lane control signs, variable speed limit signs, portal signals, variable message signs, and in-vehicle route guidance. The traffic management controller can represent different designs of such systems with logic at varying levels of sophistication (pretimed, actuated, or adaptive) by means of a flexible configuration input.

Control devices can be either linkwide (such as variable speed limits) or lane specific (e.g., lane use regulation). They are represented by road items and are characterized by their location, type, and visibility distance. Their logic is implemented directly by the traffic management controller and the analyst will need to code its logic through external scripting files (in Lua language). An example of tested implementation in the current state of SimMobility ST development is the Sydney Coordinated Adaptive Traffic System–like algorithm for traffic signals (*34*) and an innovative time slot–based algorithm for the coordinated management of intersections for autonomous vehicles (*35*).

Service Controllers

Service controllers are the central control point responsible for operation of a specific mobility service. They rely on static information as well as real-time information and communicate with the simulated vehicle operators and drivers to send them instruction at various situations. In the current development of SimMobility ST, the bus transportation framework and controller are already implemented. There are currently two settings of the controller: the bus controller and smart mobility controller.

The bus controller is responsible for the scheduling and dispatching of buses, for keeping track of individual arrival times, and for deciding on transit control strategy. Along with the bus controller, other features were implemented in SimMobility ST: the bus driver agent will be responsible for routing the bus on a fixed route, the bus movement near the bus stop, real-time passenger count, and dwell time calculation. The bus movement, for example, includes mandatory lane-changing maneuvers to reach the lane where the stop is, depending on a distance-to-stop threshold. When a bus is farther than bus-to-stop visibility, the driver agent may make discretionary lane changes according to the same logic that applies to other drivers with a preference toward the lane that contains the bus stops. Finally, the passenger and pedestrian agents will also interact in the bus framework because they are responsible for realizing their individual bus route choice, boarding choice, and alighting choice.

SimMobility ST also provides mechanisms for simulating emerging technologies that are yet to be widely available, for example, mobility-on-demand services. Such a feature was achieved with the development of uniform interfaces between a smart mobility controller and the different models in SimMobility ST. The smart mobility controller relies on third-party code that can run separately from SimMobility ST and mimics the operation of fleets, from regular taxi to Uber-like systems. This code interacts with agents and entities in SimMobility in run time, and a few features have been made accessible to it.

Two applications of the proposed smart mobility controller can be found already in the literature: an autonomous mobility on demand service, which provides one-way car sharing with self-driving electric vehicles and has emerged as a promising solution for autonomous urban transportation (*36*), and flexible mobility on demand, which provides personalized and optimized services to travelers in real time with flexibilities both on the operator and traveler side (*37*). These technologies are designed to deal with the recent trends that emphasize more flexibility through the use of shared-ride services and integration of multimodal mobility options.

Communication Network Simulator

Many simulators broaden their applicability by allowing customized interactions with third-party components and SimMobility is no exception. In addition to traditional library-based extension, SimMobility ST provides a transmission control protocol (TCP) socket integration layer that allows other software systems to interact with a running SimMobility ST simulation. This layer is primarily used in Hetu et al. (*38*) to overlay Android emulators running transit-related applications, or apps, onto existing SimMobility agents, thus providing accurate location information to the apps. In return, the apps provide more realistic within-day rerouting, creating a feedback loop, which should optimize the system.

The choice to communicate over TCP sockets has several advantages. First, it requires minimal changes to existing third-party software systems, and usually only a small communication module is needed. Second, it facilitates interactions between a larger number of simulators. In the example just discussed, SimMobility ST connects to a running instance of a network simulator, which it uses to provide accurate timing and packet loss information for messages sent between Android clients. Finally, the use of TCP sockets provides a stable, cross-platform means of interaction with clearly defined boundaries.

SYSTEM ARCHITECTURE

Overall Framework

SimMobility ST applies several design heuristics to make modeling and development easier for a heterogeneous user base. First, entities are isolated from each other and can only interact through properties that are shared among them. This isolation is achieved through the use of agent-based simulation techniques. Second, the simulator is location-agnostic with regard to agents. In other words, an agent's interface does not change depending on where it is in relation to other networks [except when message passing interface (MPI) is enabled]. Third, SimMobility ST's time step is indivisible; agents are assumed to all tick forward at once. Finally, SimMobility is hierarchical and provides sensible defaults. A good example of this behavior is the use of trip chains, which can be filled in with more information as the agent's trip progresses. If an agent does not have a route for a given segment of the trip chain, one can be estimated for it.

SimMobility ST is designed as a hybrid software framework, including both event-driven messaging and discrete time step simulation. Heterogeneous time steps are supported, allowing coarse-grained software agents such as traffic signals to interact efficiently with fine-grained agents such as drivers. Ultimately, SimMobility ST was designed with accuracy and performance as its two primary goals. To this end, it includes a parallel and a distributed component, which are now described in turn.

Parallel Computing

SimMobility ST features a robust, straightforward approach to massive parallel scalability that was designed to take advantage of the processing power of modern hardware. The majority of computations performed by SimMobility ST are done by entities as they update their internal state. This process is performed once per entity per time tick. Entities with similar time step resolutions are grouped together into "workers" that manage the update process for a given thread.

Entities can generally ignore the worker to which they are assigned, as all communication through other agents is done with buffer-backed variables. These variables use an internal buffer to allow lockless communication with any entity on any worker. In addition, entities will be automatically added to a worker when their start time of their first trip arrives and will be removed once the final trip's destination has been reached. Typically, entities only interact with their workers when requesting a manual migration. This action can occur when the agent crosses an MPI boundary (described in the following subsection), or if some kind of spatial optimization, such as the mid-term's conflux structure, is desired.

This use of buffers runs counter to traditional logic of using mutex-based locking for parallel communication. Ultimately, the conventional approach exhibits several systemic and nontrivial problems. Primarily, it limits the repeatability of the simulation by introducing an unacceptable amount of nondeterminism. Entities are constantly reading and reacting to the internal state of each other and performing these reads in parallel leads to different orderings for each simulation run. Attempting to solve this problem by ordering entity updates will remove some of the benefits of parallelism, which is particularly detrimental to entities that are not sensitive to update order. A secondary issue with lock-based synchronization is the heavy toll it places on many-cored systems, especially in traffic simulation with its high degree of interagent data dependency. As the number of discrete processing units on a system increases, more and more performance is lost to overhead.

An additional channel for communication is event-driven messaging. Here, entities can register for messages to be delivered for a series of given events, such as node arrivals or agent deactivations. These messages will be triggered during a time tick, gathered and sent at the end of that time tick, and received at the beginning of the next time tick. Thus messages inherently incur a one-tick delay, although it is possible to reduce this to zero if certain conditions hold (e.g., passengers on a bus can receive zero-delay messages). Although event processing is inherently single threaded, it can lead to large performance gains by allowing entities to deactivate their update phase until a given message arrives. Furthermore, entities can register for events and still maintain their update phase, thus allowing for a hybrid of event-driven and discrete time-stepped simulation.

Distributed Computing

Once simulations encompass a large enough number of entities, it is inevitable that parallel simulation will reach a point of diminishing returns. At this point, the simulation must be split and run on different machines (nodes) through SimMobility ST's MPI-based distributed computing platform. Although the same parallel setup just discussed is still run on each node, the internode communication is inherently less flexible and requires modelers to abandon the location-agnostic property of the single-node setup. In particular, entities must be located in a particular geometric space, and various spatial decomposition techniques are used to assign different spaces to different physical machines.

The global state of the simulation, including the road network and the trip chains, is split and distributed to each node, where it is loaded on an as-needed basis (to reduce memory usage). Agents are distributed to the node that contains their starting trip and will be automatically transferred to other nodes as they cross the relevant node's boundary. A node boundary is defined as a line perpendicular to a given road segment that divides the upstream and downstream halves of that segment between two nodes.

Entities on different nodes can sense each other by mirroring, but they cannot otherwise interact. Furthermore, entities that are not in a mirrored region cannot send each other messages unless they are on the same node. This is an unfortunate necessity, as it puts a lower bound on the number of nodes with which a given node must interact, in turn allowing SimMobility ST to scale efficiently up to arbitrarily complex road networks and simulation workloads. To work around these limitations, a delayed-hopping message protocol is in development that allows messages to be sent to agents on other nodes at a time cost of *N* time ticks, where *N* is the number of nodes between the originating agent and the receiving agent. This system is flexible and permits the time delay to be further reduced (to at least 1) through the use of software-defined relays.

Visualizer

To represent the simulator output graphically, an interface is developed in C++ using QT libraries. This graphical user interface is used for debugging purpose and to demonstrate traffic impact through vehicle animation (Figure 3). This application accepts the simulator output file produced in plain text format and then it displays vehicle trajectory at each frame tick (in microscopic view mode) or network statistics (mesoscopic view mode). It supports zoom-in and zoom-out operation to handle large or small area visualization and a road network entity search functionality to locate an object easily within the graphical user interface.

Data Management

SimMobility ST supports multiple data interfaces to exchange data with the other level or the data that are exogenous to the simulator. It is able to read or write data from and to XML files, PostgreSQL database, and the CSV files. This will give the user greater flexibility to run SimMobility ST with a variety of data sources depending on the experiment requirement. SimMobility ST data requirements can be grouped into configuration, input and output data, and model parameters.

SimMobility ST stores the configuration data and the model parameters in XML format. It provides greater readability for the user to configure SimMobility ST. Simulation input data can be either exogenous to the system (e.g., road network and traffic light phases) or received from another level of SimMobility ST (e.g., trip chains received from mid term to short term). SimMobility ST supports both XML and database interfaces for input data and, depending on the simulation need, the user can specify the format in the configuration file. Simulation outputs are generated in plain text format that can be used for further processing, and also the required portion is written in the database for passing to the next level.

CALIBRATION

Demand and Supply Parameters

Demand parameters are typically calibrated through tuning of the O-D flows. Because SimMobility uses activity schedules and trip chains instead of an O-D matrix, the trip chain is aggregated to generate the O-D parameter to be calibrated, then the updated O-D parameter ($\hat{\theta}_{k+1}$) is converted into trip chains by disaggregating through the so-called killing–cloning process for each iteration (k). Thus each activity schedule will be a parameter to be calibrated (killed or cloned). This means that each individual activity schedule, and therefore each agent, will be calibrated. Each activity schedule

FIGURE 3 SimMobility visualizer (Singapore simulation).

is considered as fixed, but replicates can be generated in the calibration process. The integrated calibration of mid- and short-term simulators would relax this assumption by calibrating directly activity-based model parameters as the demand.

Other parameters in demand side are route choice parameters. The route choice model for private transport corresponds to a path-size logit model (*39*). Accordingly, a driver's route choice decisions are captured in a probabilistic manner and are highly likely to result in the route that maximizes his or her utility, which is characterized by influencing factors including travel time and distance. On the supply side, all driving behavior parameters are considered (*28, 40*).

Calibration Framework

The simultaneous calibration of demand and supply parameters generates a large set. To deal with such complexity, the authors used the weighted simultaneous perturbation stochastic approximation (W-SPSA) (*41, 42*). The algorithm finds the best parameter set by iteratively updating the parameter set to the decreasing direction in goodness of fit, which may reply on existing measurements and on prior knowledge on demand–supply parameters from previous experiments. The SimMobility ST is run through the killing–cloning process to get initial simulation output and assignment matrix. The assignment matrix is the weight matrix for the measurements in W-SPSA. After the initial setting, the calibration loop runs until it reaches the convergence condition and the objective value is within an acceptable level of performance. The optimization problem over parameter space during the period of $H = \{1, 2, \ldots, H\}$ can be formulated as

$$\min_{x,\beta} z(\theta) = \sum_{h=1}^{H} \left[w_c z_c \left(C_h^O, C_h^S \right) + w_T z_T \left(T_h^O, T_h^S \right) + w_p z_x \left(x_h, x_h^a \right) \right] + w_p z_\beta \left(\beta, \beta^a \right) + w_p z_\gamma \left(\gamma, \gamma^a \right) \quad (1)$$

subject to

$$C_h^S = f_C(x_1, \ldots, x_h; \beta; \gamma)$$

$$T_h^S = f_T(x_1, \ldots, x_h; \beta; \gamma)$$

$$\text{lb}_{x_h} \le x_h \le \text{ub}_{x_h}$$

$$\text{lb}_\beta \le \beta \le \text{ub}_\beta$$

$$\text{lb}_\gamma \le \gamma \le \text{ub}_\gamma$$

where z_C and z_T respectively measure goodness of fit between externally observed (count: C_h^O; travel time: T_h^O) and simulated measurement (C_h^S, T_h^S); z_x compares estimated time-dependent O-D parameter (x_h) with the seed O-D (x_h^a) from MT; z_β and z_γ respectively evaluate the estimated parameter set in driving behavior (β) and route choice (γ) against a priori values (β^a, γ^a); and θ is the decision vector [$\theta = (x_1, \ldots, x_h, \beta, \gamma)$]. The parameters are bounded upper (ub) and lower (lb) limits. Each evaluation term includes a weighting coefficient (w_C; w_T; w_p), which is determined by the reliability on the external information.

W-SPSA selectively perturbs relevant parameters based on a weight matrix (w), which represents spatiotemporal correlations between each parameter and measurements. Readers can refer to the full structure of the W matrix in Lu et al. (*41*). To increase the applicability in ST calibration, which deals with many agents for large spatial ranges, a sparse matrix has been generated in this phase. Then, the gradient approximation can be formulated as

$$\hat{g}_{ki}\left(\hat{\theta}_k\right) = \frac{\sum_{j=1}^{D} w_{ji} \left[\left(\varepsilon_{kj}^{+} \right)^2 - \left(\varepsilon_{kj}^{-} \right)^2 \right]}{2 c_k \Delta_{ki}} \quad (2)$$

where

c_k, Δ_{ki} = perturbation amplitude and random perturbation vector (following the Bernoulli process), respectively;

$(\varepsilon_j^+)^2$ and $(\varepsilon_j^-)^2$ = deviation vectors measuring the distance between the observed and the simulated measurement with plus ($\hat{\theta}_k + c_k\Delta_k$) and minus ($\hat{\theta}_k - c_k\Delta_k$) perturbed parameter, respectively; and

$\hat{g}_{ki}$ = ith element of the approximation of the gradient vector; this gradient provides the amount of movement from current kth state ($\hat{\theta}_k$) to the next iteration ($\hat{\theta}_{k+1}$).

With Equation 2, update $\hat{\theta}_{k+1}$, which yields less cost on evaluation function,

$$\hat{\theta}_{k+1} = \text{arg.min}_{\hat{\theta}_{k+1}} \left(z\left(\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k \hat{g}_k \left(\hat{\theta}_k \right) \right), z\left(\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha_k \hat{g}_k \left(\hat{\theta}_k \right) \right) \right) \quad (3)$$

where arg.min is arguments of minima and α_k is a step size in a gain sequence. Therefore, SimMobility ST calibration includes a backward decision process that gives additional chance to consider opposite direction in the decision vector update. Also, note that the two different function evaluations on two parts (Equation 2) conducted on a parallelized way are independent of each other and the algorithmic parameters are selected as in Vaze et al. (*43*). Once the objective value satisfies the internal convergence term, the full calibration process is terminated.

Experimental Setting

SimMobility ST is calibrated by running the traffic for the extended central business district (CBD) in Singapore (Figure 4). This area contains more than 1,200 intersections, which are covered by more than 2,000 loop detectors. A smaller subnetwork with 10 intersections called Bugis, located inside the CBD, was also tested for assessing the impact of daily variability in the calibration process. The aggregated demand generated by SimMobility Mid-Term has 1,497 observed O-D pairs and a total of 48,988 trips. These trips (demand), 11 route choice parameters for demand, and 112 driving behavior parameters (supply) are the set of parameters to calibrate.

For the calibration, two types of data were available: loop count and GPS travel time data from probe vehicles, collected in August 2013 by the Land Transport Authority of Singapore and taxi data location, respectively. Counts had a resolution of 5 min, while the GPS data were structured in O-D travel time tables for each 30-min interval of the day. Counts were also preprocessed for outlier detection (*44*). A total of 360 sensors were used in the calibration. The following data were also used in network settings: Sydney Coordinated Adaptive

FIGURE 4 Extended CBD (shaded area) in Singapore (red dots = measurement locations).

Traffic System signal phases, geographic information system network configuration, Google transit network data for the buses routes and schedules, and freight (background) traffic data.

Calibration Result

The result shows the calculated root mean square normalized (RMSN) error between the simulated and observed count over all segments and time intervals. The table that follows shows RMSN for multiple days in a small Bugis network:

	Date							
	8/6	*8/7*	*8/13*	*8/14*	*8/15*	*8/20*	*8/21*	*8/22*
Initial	0.58	0.53	0.53	0.54	0.49	0.47	0.51	0.46
Calibrated	0.29	0.29	0.32	0.26	0.25	0.29	0.27	0.25

This table shows that the calibration framework is able to calibrate a simulator using different external data sets, with improvements 40%~50% in RMSN. In the extended CBD area and after 250 iterations, the fit-to-counts has been improved from 0.72 to 0.37 of RMSN (Figure 5*a*) and the calibrated counts became close to the 45° line (Figure 5*b*). The calibrated RMSN seems to be satisfactory given

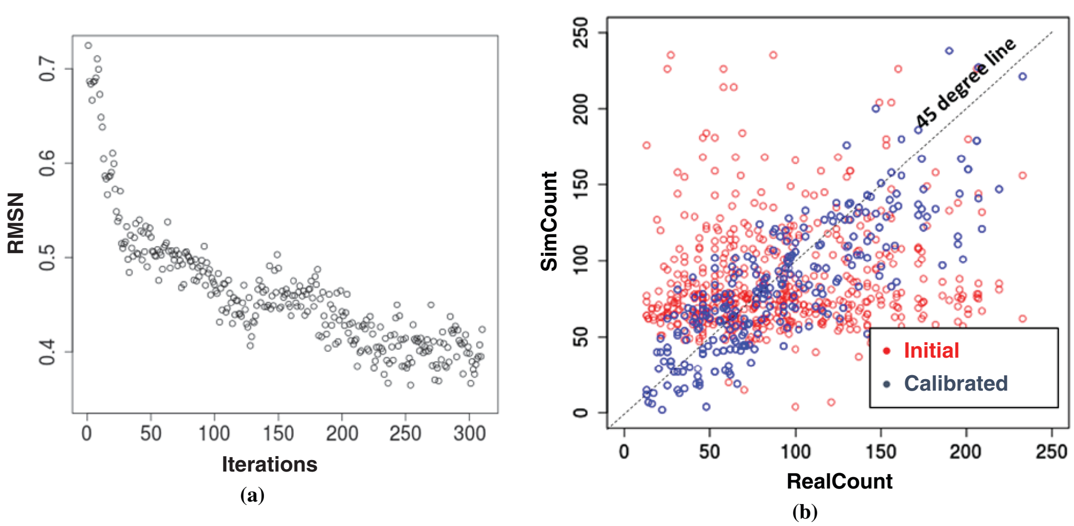


FIGURE 5 Calibration result for the Singapore extended CBD area: (*a*) RMSN over iterations and (*b*) fit-to-counts (simulated versus real counts).

the above small number of sensors and relatively large number of parameters.

CONCLUSIONS

A microscopic mobility simulator, SimMobility ST, aimed at simulating the movement of agents and the decisions and operation of control and logistic centers, is presented. SimMobility ST allows for a modular integration of specific behaviors associated with new mobility services and transportation modes. It is integrated within the SimMobility framework, a multiscale simulation platform that considers land use, transportation, and communication interactions using various behavioral models. The new simulator particularly focuses on impacts of innovative transportation services on transportation and mobility networks, thereby enabling the simulation of a portfolio of technology, policy, and investment options under alternative future scenarios. SimMobility ST has been successfully calibrated using external data in Singapore. Multiple days support the replicability of calibration capability as well. This calibrated simulator would contribute to increase simulation reliability in evaluation of new scenarios in Singapore and elsewhere.

The main ongoing development efforts have been focusing on integrating further (existing) advanced driving behavior models; designing and implementing the urban freight tour-based logic, along with its specific behaviors and logistic decisions; and implementing further smart mobility services.

Finally, the integration of a dedicated framework for simulating electrical vehicles and both environmental and safety impacts assessment modules are also three short-term key milestones. The first one will allow SimMobility ST to be used in the decision process of the design of the electrical vehicles grid and to model the associated changes in the mobility patterns at the city level. The environmental and safety impacts assessment modules will allow a more comprehensive evaluation of the technologies and services being tested within SimMobility.

ACKNOWLEDGMENTS

The authors thank Max Zhiyong and Seth Hetu for their contributions to the early development of SimMobility ST. The authors also thank the Land Transport Authority of Singapore for providing data for this research study. This research was supported by the National Research Foundation under its CREATE program and the Singapore-MIT Alliance for Research and Technology, Future Urban Mobility Interdisciplinary Research Group.

REFERENCES

1. Barceló, J. (ed.). *Fundamentals of Traffic Simulation,* 1st ed. Springer, New York, 2010. https://doi.org/10.1007/978-1-4419-6142-6.
2. *Aimsun Dynamic Simulator User's Manual, Version 7.0.* Transport Simulation Systems, Barcelona, Spain, 2011.
3. *VISSIM 5.20 User Manual.* Technical Report. Planung Transport Verkehr AG, Karlsruhe, Germany, 2009.
4. *The Paramics Manual, Version 6.6.1.* Quadstone Paramics, Edinburgh, Scotland, 2009.
5. *Traffic Simulation Software: TransModeler User's Guide.* Caliper Corporation, Newton, Mass., 2008.
6. Hidas, P. A Car-Following Model for Urban Traffic Simulation. *Traffic Engineering and Control,* Vol. 39, No. 5, 1998, pp. 300–305.
7. *CORSIM User's Guide, Version 6.* FHWA, 2006.
8. Liu, R. Traffic Simulation with DRACULA. In *Fundamentals of Traffic Simulation* (J. Barceló, ed.), Springer, New York, 2010, pp. 295–322. https://doi.org/10.1007/978-1-4419-6142-6_8.
9. Koskinen, K., I. Kosonen, T. Luttinen, A. Schirokoff, and J. Luoma. Development of a Nanoscopic Traffic Simulation Tool. *Advances in Transportation Studies,* Vol. 17, 2009, pp. 89–96.
10. Rakha, H. *INTEGRATION Release 2.40 for Windows: User's Guide, Volume I: Fundamental Model Features.* M. Van Aerde and Associates, Ltd., Blacksburg, Va., 2014.
11. Ben-Akiva, M., H.N. Koutsopoulos, T. Toledo, Q. Yang, C.F. Choudhury, C. Antoniou, and R. Balakrishna. Traffic Simulation with MITSIMLab. In *Fundamentals of Traffic Simulation* (J. Barceló, ed.), Springer, New York, 2010, pp. 233–268.
12. Behrisch, M., L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO—Simulation of Urban MObility: An Overview. In *SIMUL 2011: Proceedings of the Third International Conference on Advances in System Simulation,* IARIA, Wilmington, Del., 2011, pp. 55–60.
13. *Cube Dynasim.* http://www.citilabs.com/software/cube/cube-dynasim/. Citilabs. Accessed July 31, 2016.
14. *TU0903-Cost Action.* http://www.multitude-project.eu/. Accessed July 31, 2016.
15. Pipes, L.A. An Operational Analysis of Traffic Dynamics. *Journal of Applied Physics,* Vol. 24, No. 3, 1953, pp. 274–281. https://doi.org/10.1063/1.1721265.
16. Hranac, R., D. Gettman, T. Toledo, V. Kovvali, and V. Alexiadis. *NGSIM Task E.1-1: Core Algorithms Assessment.* Technical Report. FHWA, 2004.
17. Choudhury, C.F. *Modeling Driving Decisions with Latent Plans.* PhD thesis. Massachusetts Institute of Technology, Cambridge, 2007.
18. Krajzewicz, D., J. Erdmann, M. Behrisch, and L. Bieker. Recent Development and Applications of SUMO—Simulation of Urban Mobility. *International Journal on Advances in Systems and Measurements,* Vol. 5, No. 3–4, 2012, pp. 128–138.
19. *TRANSIMS (TRansportation ANalysis SImulation Systems).* 2014. http://ndssl.vbi.vt.edu/transims-docs.html. Accessed June to July 2014.
20. *MATSim: Multi-Agent Traffic Simulation.* 2014. http://www.matsim.org. Accessed June to July 2014.
21. Bellemans, T., B. Kochan, D. Janssens, G. Wets, T. Arentze, and H. Timmermans. Implementation Framework and Development Trajectory of the FEATHERS Activity-Based Simulation Platform. *Transportation Research Record: Journal of the Transportation Research Board,* No. 2175, 2010, pp. 111–119. https://doi.org/10.3141/2175-13.
22. Lu, Y., K. Basak, C. Carrion, H. Loganathan, M. Adnan, F.C. Pereira, V.H. Saber, and M. Ben-Akiva. SimMobility Mid-Term Simulator: A State of the Art Integrated Agent Based Demand and Supply Model. Presented at 94th Annual Meeting of the Transportation Research Board, Washington, D.C., 2015.
23. Ben-Akiva, M., J.L. Bowman, and D. Gopinath. Travel Demand Model System for the Information Era. *Transportation,* Vol. 23, 1996, pp. 241–266.
24. Le, D.T., G. Cernicchiaro, C. Zegras, and J. Ferreira. Simulation of Synthetic Establishments for Modeling Firm Behavior in SimMobility. Presented at International Scientific Conference on Mobility and Transport Transforming Urban Mobility, mobil. TUM, Munich, Germany, June 2016.
25. Teo, J., L. Cheah, Y.J. Lee, V. Marzano, J. Santos, C. L. Azevedo, F. Zhao, and M. Ben-Akiva. An Integrated Sensing-Based Urban Freight Data Collection Framework: Methodology and Pilot Projects in Singapore. Presented at URBE Conference, Rome, Oct. 2015.
26. Bowman, L., and M.E. Ben-Akiva. Activity-Based Disaggregate Travel Demand Model System with Activity Schedules. *Transportation Research Part A: Policy and Practice,* Vol. 35, No. 1, 2001, pp. 1–28. https://doi.org/10.1016/S0965-8564(99)00043-9.
27. Ahmed, K. *Modeling Drivers' Acceleration and Lane Changing Behavior.* PhD thesis. Massachusetts Institute of Technology, Cambridge, 1999.
28. Toledo, T., H. Koutsopoulos, and M.E. Ben-Akiva. Integrated Driving Behavior Modeling. *Transportation Research Part C: Emerging Technologies,* Vol. 15, No. 2, 2007, pp. 96–112. https://doi.org/10.1016/j.trc.2007.02.002.
29. Basak, K., S. Hetu, Z. Li, C. L. Azevedo, H. Loganathan, T. Toledo, R. Xu, L.-S. Peh, and M.E. Ben-Akiva. Modeling Reaction Time Within a Traffic Simulation Model. Presented at 92nd Annual Meeting of the Transportation Research Board, Washington, D.C., 2013. https://doi.org/10.1109/ITSC.2013.6728249.

30. Chovan, J., L. Tijerina, G. Alexander, and D. Hendricks. *Examination of Lane Change Crashes and Potential IVHS Countermeasures.* Technical Report. U.S. Department of Transportation, NHTSA, 1994.
31. Sarvi, M. Heavy Commercial Vehicles-Following Behavior and Interactions with Different Vehicle Classes. *Journal of Advanced Transportation,* Vol. 47, 2013, pp. 572–580.
32. Nourinejad, M., A. Wenneman, K. Nurul Habib, and M. J. Roorda. Truck Parking in Urban Areas: Application of Choice Modelling Within Traffic Microsimulation. *Transportation Research Part A: Policy and Practice,* Vol. 64, 2014, pp. 54–64. https://doi.org/10.1016/j.tra.2014.03.006.
33. Teo, J. S. E., E. Taniguchi, and A. G. Qureshi. Evaluation of Distance-Based and Cordon-Based Urban Freight Road Pricing in E-Commerce Environment with Multiagent Model. *Transportation Research Record: Journal of the Transportation Research Board,* No. 2269, 2012, pp. 127–134.
34. Daizong, L. *Comparative Evaluation of Dynamic TRANSYT and SCATS-Based Signal Control Systems Using Paramics Simulation.* MSc thesis. National University of Singapore, 2003.
35. Tachet, R., P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti. Revisiting Street Intersections Using Slot-Based Systems. *PLoS One,* Vol. 11, No. 3, 2016, p. e0149607. https://doi.org/10.1371/journal.pone.0149607.
36. Azevedo, C. L., K. Marczuk, S. Raveau, H. Soh, M. Adnan, K. Basak, H. Loganathan, N. Deshmukh, D. H. Lee, E. Frazzoli, and M. Ben-Akiva. Microsimulation of Demand and Supply of Autonomous Mobility On-Demand. *Transportation Research Record: Journal of the Transportation Research Board,* No. 2564, 2016, pp. 21–30.
37. Atasoy, B., T. Ikeda, and M. Ben-Akiva. Optimizing a Flexible Mobility on Demand System. *Transportation Research Record: Journal of the Transportation Research Board,* No. 2536, 2015, pp. 76–85. https://doi.org/10.3141/2536-10.
38. Hetu, S., S. Vahid, and L. S. Peh. Similitude: Interfacing a Traffic Simulator and Network Simulator with Emulated Android Clients. Presented at IEEE 79th Vehicular Technology Conference, Seoul, South Korea, May 2014.
39. Ramming, M. S. *Network Knowledge and Route Choice.* PhD dissertation. Massachusetts Institute of Technology, Cambridge, 2001.
40. Ciuffo, B., and C. L. Azevedo. A Sensitivity-Analysis-Based Approach for the Calibration of Traffic Simulation Models. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 15, No. 3, 2014, pp. 1298–1309. https://doi.org/10.1109/TITS.2014.2302674.
41. Lu, L., Y. Xu, C. Antoniou, and M. Ben-Akiva. An Enhanced SPSA Algorithm for the Calibration of Dynamic Traffic Assignment Models. *Transportation Research Part C: Emerging Technologies,* Vol. 51, 2015, pp. 149–166. https://doi.org/10.1016/j.trc.2014.11.006.
42. Antoniou, C., 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. *Transportation Research Part C: Emerging Technologies,* Vol. 59, 2015, pp. 129–146. https://doi.org/10.1016/j.trc.2015.04.030.
43. Vaze, V. S., C. Antoniou, Y. Wen, and M. Ben-Akiva. Calibration of Dynamic Traffic Assignment Models with Point-to-Point Traffic Surveillance. *Transportation Research Record: Journal of the Transportation Research Board,* No. 2090, 2009, pp. 1–9. https://doi.org/10.3141/2090-01.
44. Lu, X. Y., P. Varaiya, R. Horowitz, and J. Palen. Faulty Loop Data Analysis/Correction and Loop Fault Detection. In *Proceedings of 15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting,* Washington, D.C., 2008.

The Standing Committee on Traffic Flow Theory and Characteristics peer-reviewed this paper.