Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

FROM OPENSTREETMAP AND CELL PHONE DATA TO ROAD NETWORK SIMULATION MODELS

Kai Gutenschwager

Institute of Information Engineering
Ostfalia University of Applied Sciences
Am Exer 2
Wolfenbüttel, 38302, GERMANY

Robert D. McLeod
Marcia R. Friesen

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, MB, R3T 5V6, CANADA

ABSTRACT

In the field of supply chain simulation, transport relations are often modeled as transport times using distributions. Considering long-distance transport relations, this is usually a suitable approach. But, for short-distance transports within large cities, delays depend on specific roads and the time of day. Some simulation tools offer geographical data for modeling actual roads. However, in order to model time-dependent transport times, additional data are needed. In this paper, we present an approach to tackle this problem. Road networks are derived from OpenStreetMap data (including traffic signals). In order to obtain the average speed of vehicles on an hourly basis, we conduct pre-simulation runs modeling the entire inner-city traffic. The respective vehicle rides are derived from trajectory data of cell phone users, where the assignment of users to cell phone tower sections is given for each hour of the day. First results for the city of Winnipeg are presented.

1 INTRODUCTION

Supply chain simulation has become a standard method in the field of supply chain design within the last years and various specialized simulation tools have been developed (Gutenschwager et al. 2018; Terzi and Cavalieri 2004). The main focus of most tools is to provide a basis to analyze design decisions by simulating the main material flows along with the underlying operative information and decision processes including forecasting, production, and replenishment planning as well as distribution and transport planning. In this paper, we focus on the transportation processes within such models. Modeling transport processes is often based on defining fixed transport relations which connect different locations within a supply network, e.g., suppliers with production sites or distribution centers. Some simulation tools further allow for modeling multi-modal transports consisting of several transport relations, where loading units change the means of transport at respective hubs. With respect to the overall costs, the main parameters addressed in respective simulation studies are often the capacity of transport relations (for example the maximum number of means of transport per day) and the frequency of transports, sometimes modeled as more-realistic transport schedules on a daily basis. Here, also stochastic influences need to be modeled, which relate to interruptions or delays in the transport processes.

In recent years, the distribution of goods within urban areas – defining the last step of supply chains – has received an increasing research interest. Due to a higher awareness of pollution, public regulations, and the general aim to reduce traffic in inner cities, new approaches need to be considered (Rabe et al. 2016). Rabe et al. (2018) present a simulation approach to evaluate distribution structures using urban consolidation centers within the city of Athens, Greece, using the simulation tool *SimChain* (Gutenschwager and Alicke 2004). The distance for each trip is calculated as the weighted Euclidean distance. In this context, simulation models including the underlying road network should lead to a higher accuracy. The simulation

978-1-7281-3283-9/19/$31.00 ©2019 IEEE

tool *Anylogistix*, for example, offers a respective functionality using a GIS database to automatically define paths between locations of a given scenario (Ivanov 2018). Ebert and Friedrich (2018) present a simulation model for evaluating urban consolidation centers with urban access regulations using an actual road network.

Furthermore, the planning functionalities for designing tours from different distributions centers to hubs and from hubs to the final customers, need to be provided within the simulation model. The underlying optimization models typically refer to vehicle routing problems (VRP) with various additional restrictions, such as time windows for visiting customers or the maximum daily driving time for truck drivers. In some simulation tools, simple construction and improvement heuristics are implemented to achieve reasonably good solutions for dynamically occurring instances of VRP-related problems, such as the sweep or the savings algorithm. A comparison of these two algorithms is given by Paessens (1988). The inputs for such models are typically deterministic driving times. Driving times can be derived from distances and road types, historical data, or systems like Google Maps or OpenStreetMap (OSM). OSM is a non-commercial information system for geographical data. The main problem applying deterministic driving times for each transport relation in urban transports is that driving times highly depend on road sections and the time of day. As a consequence, time-dependent vehicle routing problems (TDVRP) need to be considered (Malandraki and Daskin 1992). Furthermore, reflecting the environmental sensitivity of vehicle routing problems, Green Vehicle Routing Problems (GVRP) could further be considered (Lin et al. 2014). To our knowledge, no simulation tool in the field of supply chain simulation offers such functionalities for planning tours.

In this paper, we present an approach to gather all necessary data and to automatically generate a simulation model of the underlying road network. Our approach can be considered a microscopic traffic simulation, modeling individual vehicles and the traffic infrastructure (Feldkamp and Strassburger 2014). In a first step, we generate a detailed simulation model layout on the basis of data imported from OSM. The approach includes a concept to additionally model intersections served by traffic signals. In order to obtain the time-dependent average speed for each road segment, we carry out simulation experiments for the entire inner-city traffic. The actual traffic load is derived from trajectory data of cell phone users. Here, we use the anonymized data of about 180,000 cell phone users in the greater area of Winnipeg for one week. As a result, we obtain a statistic of the average speed per road segment (edges in the underlying graph) on an hourly basis. These results are stored in a database and can be used to parameterize simpler models where only the vehicles involved for transports within a supply chain are modeled (without any traffic signals or different lanes of a road).

The rest of this paper is organized as follows. Section 2 describes our approach to generate simulation models on the basis of OSM. Section 3 describes our our approach to model the traffic load based on trajectory data of cell phone users. Section 4 presents our implementation and first results for models we have set up for the city of Winnipeg. Section 5 summarizes our approach and gives a short outlook for future research.

2 A GENERIC SIMULATION MODEL BASED ON OPENSTREETMAP DATA

In this section, we describe our approach to generate a detailed simulation model of a road network for the simulation tool *Plant Simulation*. The process consists of four steps:

1. Download and import of data from OSM into a MySQL database
2. Data preprocessing (deriving relevant edges from ways and nodes)
3. Import of nodes and edges into *Plant Simulation* and generic creation of road networks
4. Automatic creation (and parameterization) of traffic signal controls

In the literature, numerous approaches deal with using OSM data to set up models for traffic simulation (Matsumoto et al. 2016; Arellano and Mahgoub 2013; Backfrieder et al. 2013; Zilske et al. 2011). Xia et al. (2018) outline an approach to obtain models for the traffic simulation tool SUMO. The authors observed some unrealistic behavior of the initial model for the city of Shanghai due to missing or incorrect OSM data (including the number of lanes) and also due to the transformation process itself, which sometimes led to two coexisting road junctions in the SUMO model. After an update of the data, the model reached a valid state, though. Li et al. (2017) set up a road network model for Shenzhen, consisting of 86 roads, where some roads (being isolated from the main road structure) have been removed. Meyer et al. (2013) present an approach to generate models from OSM data using the simulation tool *Plant Simulation*. The approach focuses on generating correct road structures and junctions without modeling traffic signals.

OSM data mainly consist of nodes, ways, and relations. For each node, the longitude and latitude are given along with a list of possible tags to add further information, e.g., if the node represents a traffic signal. Ways consist of a sequence of nodes and may represent streets, pedestrian sidewalks, or railways. By using tags, ways can be differentiated respectively. Typical values for the tag *highway* are, e.g., *motorway*, *motorway_link primary*, *secondary*, *tertiary*, or *residential*. For downloading data we use http://overpass-turbo.eu/. Here, filter scripts can be defined to only extract necessary data. For the example of the city of Winnipeg, we have filtered 40,414 nodes and 10,424 ways of types *motorway*, *trunk*, *primary*, *secondary*, and *tertiary* along with the respective *links* of each highway type including all nodes representing a traffic signal in an XML-file.

2.1 Database Structure and Preprocessing

In the next step, the data downloaded as an XML-file are transferred into a MySQL database. The structure of the database is given in Figure 1. Table *highwayType* is parameterized manually for all highway types to be filtered. Here, also a default value for the maximum speed is defined, as the respective tag is not necessarily given for each way. Tables *node*, *way*, and *way_has_node* are directly derived from the XML-file:

- First, we include nodes of the given ways into the database. Each node is included into the database with given latitude and longitude. We further check if the tag *highway* is given and set to *traffic_signals*. In this case, the attribute *isTrafficSignal* in table *node* is set to true.
- In the next step, all ways are included into the database, where we further check for the tags *highway*, *lanes*, *oneway*, and *maxspeed* to receive values for the respective columns in table *way*.
- We also include all nodes belonging to a way into table *way_has_node* with a respective sequence number referring to the sequence the nodes are given in the list of the way. For each node, we further count to how many ways the node belongs to. This value is stored in column *numberOfWays* in table *node*. This information is needed for defining edges in a preprocessing phase.

After all data are included in tables *node*, *way*, *way_has_node*, edges for the resulting graph are generated in the database (implemented as a procedure in the underlying database). Each edge refers to a given way, but for each way more than one edge might be defined in the database. Each edge consists of two nodes (start and end), with the first node of a way and the last node of a way always belonging to an edge. If no (further) nodes occur on the way under consideration that also belong to another way, and no traffic signals are on the way either, then the entire way is mapped to one single edge. If, however, a node in the sequence belongs to more than one way or a traffic signal is given, an intersection is assumed and further edges need to be defined to model the correct road structure. The distance d of each edge consisting of n nodes in the original way is computed as the sum of the air distances from node i to the next node $i+1$ in the given sequence of nodes of the way with r being the radius of the earth:

$$d = \sum_{i=1}^{n-1} \arccos(\sin(lat_i) + \sin(lat_{i+1}) + \cos(lat_i) \cdot \cos(lat_{i+1}) \cdot \cos(lon_i - lon_{i+1})) \cdot r.$$

An example is given in Figure 2, where a way with nine nodes is given, with node 3 also belonging to another way and node 8 being a traffic signal. As a result, three edges are defined.

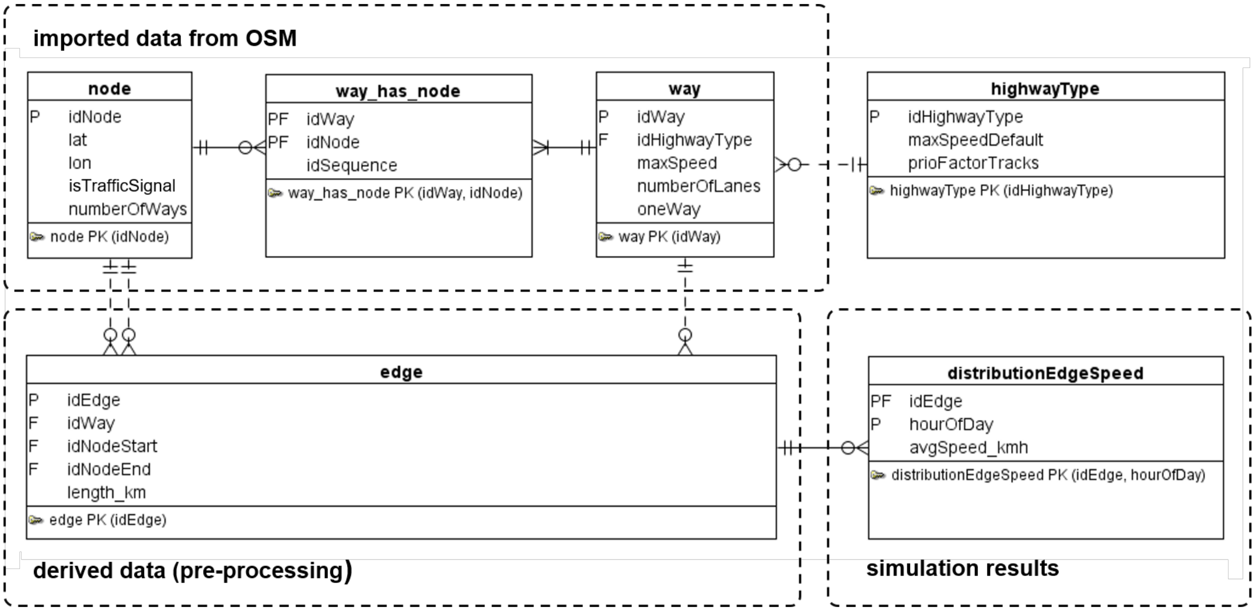

Figure 1: Data model for the road network.

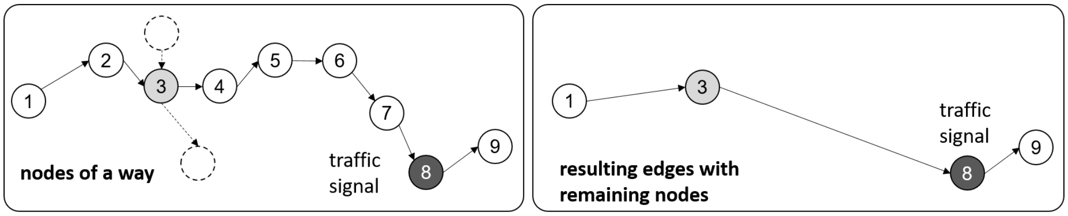

Figure 2: Transformation of ways to edges.

2.2 Generic Simulation Model

When importing data from the database to set up a road network within the simulation tool *Plant Simulation*, also partial models can be created by defining a frame with a maximum and a minimum latitude and longitude, such that only edges are imported, where at least one of the two nodes lies within the frame.

All edges with start or end node outside the frame are shortened so they fit into the given frame using the polar angle between the two nodes involved. The respective graph is then analyzed for connectivity. Here we propose the following approach, where the respective steps are illustrated in Figure 3: In a first step, we find all sub-graphs (indicated as groups) that are connected by only two-way edges, such that each node can be reached from any other node within the group. In the second step, all one-way connections (paths) between these groups are computed. In step 3, we extend each group by the nodes that belong to a path starting and ending at nodes belonging to the same group (in the example, nodes 7 and 8 of group 3). In step 4, we check for circles of paths between different groups. All groups belonging to such a circle can be merged to one group. In the example, there is a path from group 1 to group 2 and from group 2 to group 1 (via node 9), such that these two groups along with all nodes on the respective paths are merged.

The remaining paths can be used to define further start (inbound paths) or destination nodes (outbound paths) for vehicle rides. In the example, nodes 16 and 17 may be used as starting nodes with a destination node within group 1. On the other hand, there is a path from group 1 to group 4, such that nodes 8, 13, 14 and 15 may be used as destination nodes for vehicle rides that start at a node in group 1 (or the respective inbound nodes 16 and 17 of that group). In a last step, only the largest connected graph is kept. In the

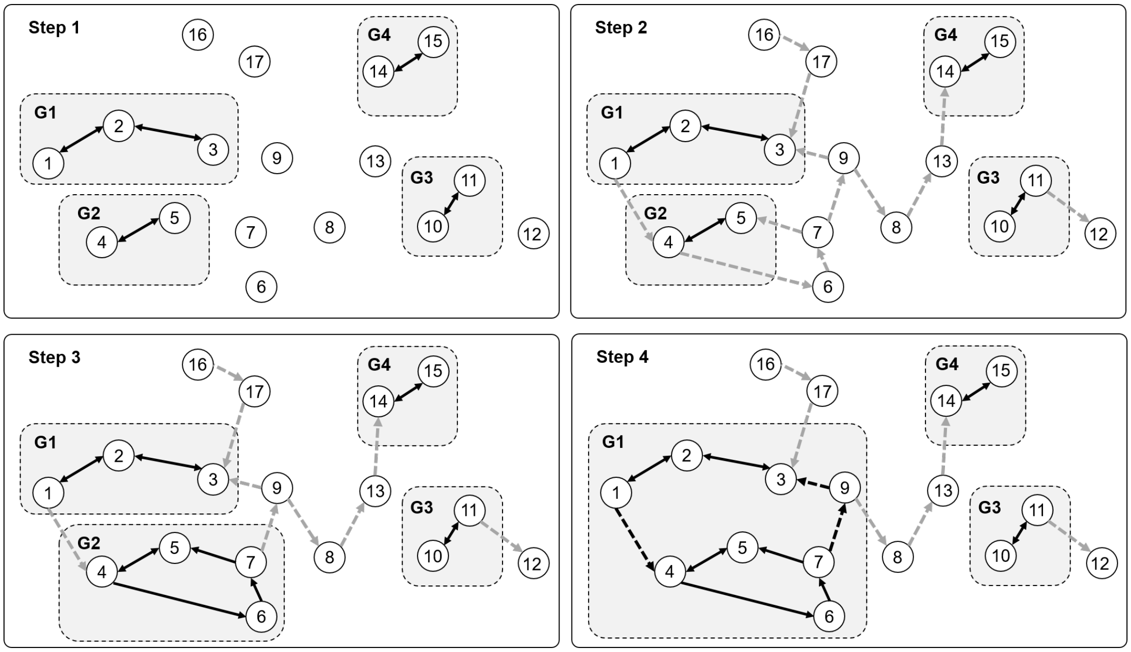

Figure 3: Graph analysis and node selection.

example, nodes 10, 11 and 12 would not not be considered anymore. The nodes belonging to the final graph (along with a classification of the inbound and outbound nodes) are used for defining the vehicle rides for the simulation model.

For generating the respective road network for the remaining edges within *Plant Simulation* we use the object class *Track*. The nodes are not modeled explicitly. However, we store all nodes in a table object. In a subtable *tabEdges* all ingoing and outgoing edges are administrated for each node. Setting up a simulation model consists of generating instances of a derived class *TrackRoadNetwork* of class *Track* for all edges (one per direction for two-way edges). Each track is connected to all its successor tracks using the standard object class *Connector*.

The road network is created in an empty frame called *map*, where the geographical position of all nodes is mapped to x- and y-coordinates (using a given maximum x- and y-coordinate) according to the given latitude and longitude of each node. Each track is initialized with the length of the edge, as well as the maximum speed, the number of lanes, and the highway type as user-defined attributes. Furthermore, the information whether the start or end node of the edge refers to a traffic signal is stored in the derived track object. *Plant Simulation* includes an automatic routing of vehicles on tracks (calculating the shortest path to a given destination track dynamically). Here, weights can be defined for each track, such that the shortest path does not have to correspond to the length of the track. We have defined a column *prioFactorTracks* corresponding to such weights for each highway type in table *highhwayType* to make it possible to, e.g., give motorways (with usually a higher maximum speed) the highest priority.

The standard track object of *Plant Simulation* does not provide several parallel lanes. In order to obtain a realistic model, we further create parallel tracks if more than one lane exists. The respective information is given as a tag in the OSM data. Including parallel lanes is implemented along with creating intersections with traffic signals, which will be described next.

2.3 Modeling Intersections with Traffic Signals

Some nodes are tagged as traffic signals in the OSM data. However, traffic signals are modeled independent from each other in OSM, i.e., we do not know, which traffic signals belong together (for example, to regulate the traffic at an intersection). We, therefore, cluster the given nodes, assuming that the traffic signals are not further than 50 meters away from each other. For each group of traffic signals, an instance of the object class *TrafficSignalControl* is created.

All edges (tracks) belonging to a traffic signal control are classified as inbound or outbound tracks or tracks that connect inbound and outbound tracks within an intersection. According to the angle of the edge (from start to end node) the nodes are further distinguished according to their (rough) direction as north-, east-, south-, or westbound. A typical configuration is given in Figure 4 (left) with seven nodes and eight tracks. The number of lanes is given as the weight of the edges. Note, that the edges (1;2) and (2;6) have been two-way ways in the original OSM data.

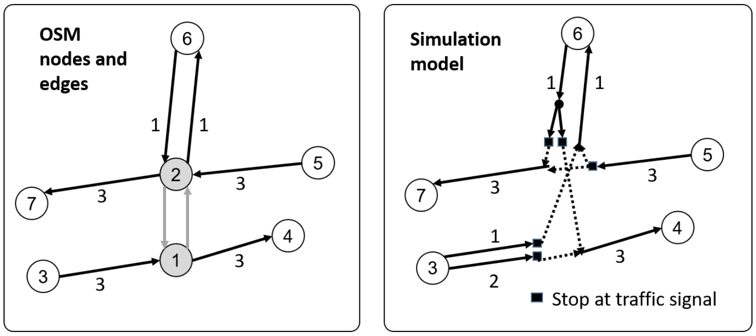

Figure 4: Example for defining an intersection served by a traffic signal control.

If a track consists of more than one lane, a specific track for left turns is created if a respective outgoing track is identified, and the edges are reconnected accordingly. In the example, edge (3;1) having three lanes is split into two tracks reducing the number of lanes by one for the main track. The tracks are connected as given in Figure 4 (right). As a left turn is also possible for vehicles entering from the north, we additionally include a short track (being long enough for one vehicle) for a left turn, even though only one lane is given. The original track is split into two parts to be able to connect the short track for left turns correctly. The main idea behind this is that in most cases it is still possible to pass a car established in the intersection for a left turn, by passing on the right. Connections allowing u-turns are not shown in the example. But, as the original edges of the OSM data allow u-turns in the example, further connections are also included in the final simulation model.

Finally, additional tracks are created for tracks having more than one lane. Here, the additional tracks are connected to all successors of the main track. The new tracks are not connected to the predecessors of the main track though. Instead, for each vehicle trying to enter the main track (being the only successor lane) a method is executed selecting the best lane to continue the ride. Here, we select the lane with the most available space.

For each traffic signal control, different phases need to be defined for coordinating the traffic flow. For each phase, vehicles on a given set of inbound tracks need to stop at the respective traffic signals. In the simulation model, the exits of the respective inbound tracks are temporarily locked, which is indicated by an attribute of the object class *Track*, so no additional implementation effort has been necessary. The traffic signal control object we have developed allows for defining different signal phase schemes. As a standard, we have defined a scheme consisting of six phases (Figure 5).

If this scheme is chosen, and the intersection combines less incoming and outgoing tracks (an example is given in Figure 4), the respective phases used exclusively for the missing directions are ignored. For the example phases 1 and 4 can be ignored. Phase 5 could still be kept for vehicles making a right turn, even though phase 6 would also cover the respective traffic flow.

A main problem is now to define the duration for each signal phase. Unfortunately, no such data can be derived from OSM either. Here, each phase duration needs to be parameterized individually for each

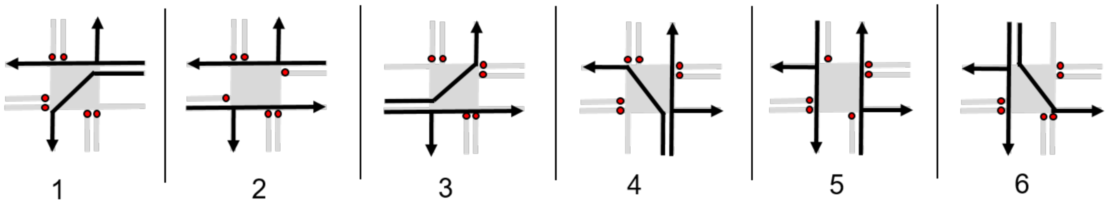

Figure 5: Example scheme for signal phases.

traffic signal control, including the definition of different time shifts. In Winnipeg, typically three shifts are differentiated:

- NIGHT: 11:00 pm - 7:00 am
- MORNING: 7:00 am - 3:00 pm
- EVENING: 3:00 pm - 11:00 pm

In order to get reasonably good values, a simulation run can be performed with the initial scheme described above having a phase duration of $d = 45$ seconds for all n (remaining) phases. A statistic keeps track of the number of vehicles passing the intersection for each inbound track including all parallel lanes (separate for each shift). This statistic can be used to re-parameterize the duration of each signal phase. Here, the following assumptions are made:

- A minimal signal phase duration d^{min} is given (d^{min} = 20 seconds)
- The total cycle time T needs to be met with $T = n \cdot 45$ seconds.

For each shift, a new scheme is created for the traffic signal control under consideration. The resulting duration d^* of each phase (for the considered shift) depends on the relative number of vehicles passing each inbound track (within the considered shift). Our approach for computing the duration for each signal phase is given as:

1. For all main inbound tracks used for going straight and making right turns we check if a respective track for left turns is defined. If so, we reduce the number of vehicles that have passed the main track (including additional lanes) by the number of vehicles that have made a left turn (coming from the same direction). We assume that when making left turns (phases 1, 3, 4, and 6), the same number of vehicles that make a left turn can also go straight in parallel.
2. For each phase we then check all incoming tracks being unlocked in the considered phase and determine the bottleneck direction, i.e., the track (including additional lanes) having the maximum number of vehicles that pass the intersection. The maximum number of vehicles of all given directions in phase i is referred to as m_i.

The relative number of vehicles p_i of phase i is used to calculate the initial duration $d^{init} = p_i \cdot T$ for each signal phase i. If the initial duration d_i^{init} is less than the minimum duration d^{min}, the duration of phase i has to be increased with $d_i^- = max(0;\ d^{min} - d_i^{init})$. In order to obtain the total cycle time T, the duration of other phases need to be reduced. We define $d_i^+ = max(0;\ d^{init} - d_i^{min})$. For each phase i with $d_i^- > 0$, we set the duration $d_i^* = d^{min}$. With a factor r defined as sum of all d_i^- divided by the sum of all d_i^+, we set the duration for all other phases to $d^* = d^{init} \cdot r$. An example is given in Table 1. First results for this approach are given in Section 4.

Table 1: Example for determining the duration of the phases of a traffic signal control.

phase	m	p	d^{init}	d^{-}	d^{+}	d^{*}
2	305	72.27%	130.09	0.00	110.09	114.09
3	61	14.45%	26.02	0.00	6.02	25.14
5	7	1.66%	2.99	17.01	0.00	20.00
6	49	11.61%	20.90	0.00	0.90	20.77
sum	422	100.00%	180.00	17.01	117.01	180.00

3 DEFINING AREAS AS SOURCES AND DESTINATIONS FOR VEHICLE RIDES

In order to generate the entire traffic load, different information sources may be taken into account. Li et al. (2017) use real-time traffic data sets available for some main roads, including the average speed of vehicle flow and congestion level. Xia et al. (2018) use data provided by a taxi company including detailed traces of 13,750 taxis for one month. Our approach is based on anonymized trajectory cell phone data of about 180,000 users, which is about a quarter of Winnipeg's population. The data have already been used in previous research activities (Wijedasa et al. 2013; Neighbour et al. 2012). In this section, we briefly describe the data preprocessing step and the definition of areas that refer to cell phone tower sectors. In our data, 166 cell tower sectors are given with latitude, longitude, and a definition of the sectors. Usually, three sectors are defined by given angles. For each node of the road network, the closest cell phone tower and the respective sector (according to the given angles that define the respective sector) are determined. It should be noted that this particular type of cell phone data is not related to a person placing a call or a text. The data used here are an example of administrative data (Authentication, Authorization, and Accounting (AAA)), recorded periodically (approximately at 15 minute intervals) so that the carrier knows where to deliver incoming calls, as such the only requirement is that the phone be powered on.

3.1 Data Preprocessing

The trajectory data of cell phone users is given as a simple list for each user stating the cell tower section the user was assigned to for each time frame (of fixed length). For each user, we check if the user has changed the location from one time frame to the next, indicated by switching from one cell tower sector to another. Each such change of a cell phone sector can be interpreted as a vehicle driving from a track belonging to the start cell tower sector to a track belonging to the target cell tower sector.

Therefore, we evaluate how many users have switched from cell tower sector i to cell tower sector j per time frame. The resulting list, referred to as the *travel matrix*, is the basis for defining vehicle rides within the given time frames. However, not all cell phone tower sections involved need to be within the road network under consideration. Therefore, we first define areas that serve as sources and sinks for vehicle rides: All cell phone towers outside the frame of the road network under consideration are mapped to a geographical location at the given frame of the road network. Next, areas are defined, where cell tower sectors are aggregated in case the distance is less then 500 meters. This is especially useful for cell tower sectors that have been mapped to the frame of the road network under consideration. Figure 6 shows two examples for road networks within the city of Winnipeg along with the areas defined by cell tower sectors (given as green circles). The light green circles outside the frame are areas defined by cell towers that have been mapped to the frame of the road network. The sizes of the areas refer to the temporary number of vehicles parked in that area.

3.2 Definition of Vehicle Rides

Vehicle rides are defined according to the computed travel matrix, where the original list is adopted to vehicle rides between areas. In order to increase or decrease the total number of vehicle rides for a given

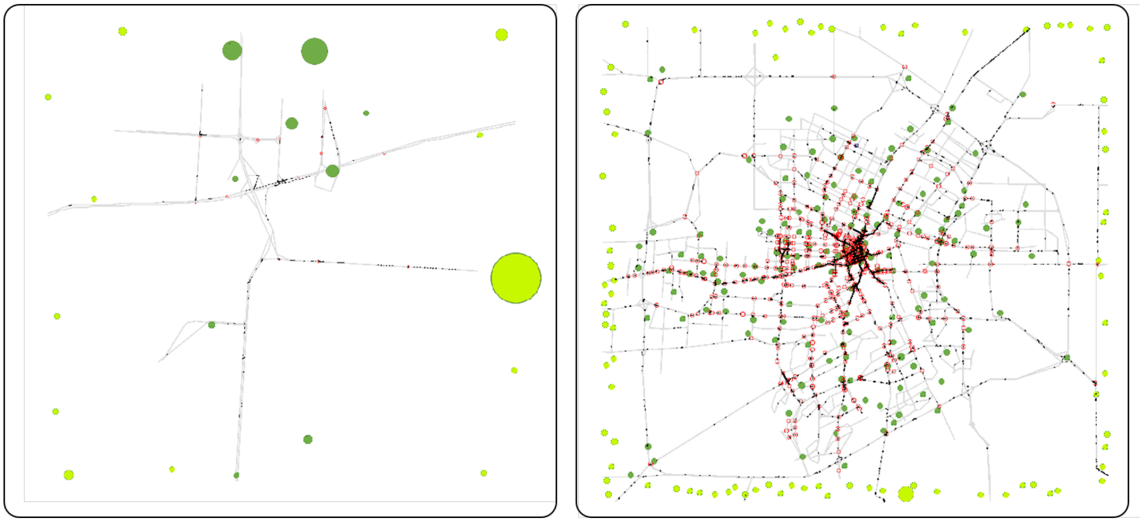

Figure 6: Simulation model of the main road network of Winnipeg.

scenario, a factor is introduced to compute the number of vehicles that are represented by a single cell phone user changing a cell phone sector (or a respective area on the map).

For each vehicle ride, a starting track and a destination track are chosen randomly from the list of tracks that are assigned to the area, also using the results from the test of connectivity of the underlying graph (see Section 2.2) to ensure that a path between the start and destination track actually exists. The time of departure is selected randomly within the given time frame for each vehicle using a uniform distribution. The areas are objects within the *Plant Simulation* model. As the entire traffic load is computed in advance, the expected number of vehicles leaving and entering each area can be computed per time frame in advance, also. For each area, we can further compute the minimum number of initial vehicles. In each simulation run, the minimum number of vehicles is created in the initialization phase in each area. The number of vehicles within each area is observed as a time series.

4 IMPLEMENTATION AND FIRST RESULTS

The overall generic simulation model, implemented as an object library in *Plant Simulation* consists of the following objects:

- *LOAD_MAP_DB* contains all methods and tables to import the OSM data from the database and to create the road network in object *MAP*, (following the approach presented in Section 2).
- *CELL_TOWER* comprises all tables and methods necessary to derive the traffic load from the cell phone data and the creation of areas in object *MAP*.
- *MAP* only contains tracks that are connected by the standard connector, areas and traffic signal controls (see Figure 6).
- *SIM* contains the entry and exit controls for the tracks within the object *MAP*. In the entry method the maximum speed of the vehicles is set according to the maximum speed of the track. Here, also first implementations for detecting and solving possible deadlock situations are given.
- *StatCounter* and *StatMinMaxMean* serve as containers for respective key figures. The main key figures implemented are the number of rides per hour, the average speed (per hour and edge), the total number of cars on the road per hour.

Detailed statistics of the average speed of the vehicles are stored in an attribute of each track object, the subtable *statHourlySpeed*. For each hour, the average travel time of all vehicles using the track object (edge) within the respective hour is computed and stored in *statHourlySpeed*. These values can be exported to the underlying database in table *distributionEdgeSpeed*, which is input for simpler supply chain simulation models, where the maximum speed is adjusted dynamically on an hourly basis for all vehicles. The given data can also used as input data for solving the TDVRP.

We have set the acceleration of the vehicles to 3.5 m/s^2 and their length to 5.1 m for all experiments. The maximum speed of the vehicles is set to the respective value of the track dynamically. *Plant Simulation* further offers methods to control the distance between vehicles, but we have not implemented any sophisticated approaches. The aim of the experiments is to get first insights of the model behavior and to analyze the impact of changing the duration of signal phases with the approach described in Section 2.3.

We have tested our approaches for several road networks. An example of a rather small model is shown in Figure 6 (left), which consists of 218 nodes, 245 edges, and 17 traffic signal controls, nine areas within the given road network (referring to actual cell towers) and 12 areas outside the actual road network for modeling traffic flows leading into or out of the road network under consideration. A total number of nearly 160,000 vehicle rides have been created. For this model we have compared the average speed for the initial model, where all signal phases have a length of 45 seconds. After the initial simulation run has been conducted, our approach to significantly improve the duration of the signal phases is applied for all traffic signal controls. A comparison for the resulting average speed of the vehicles for a simulation time of four days (starting at 3pm) is given in Figure 7. The average speed is increased from 30 km/h in the initial simulation run to about 38 km/h after improving the duration of all signal phases. For the roads under consideration this refers to a rather realistic value. The average runtime of the model is about 105 seconds using an Intel Core i5-6440HQ CPU with 2.6 GHz.

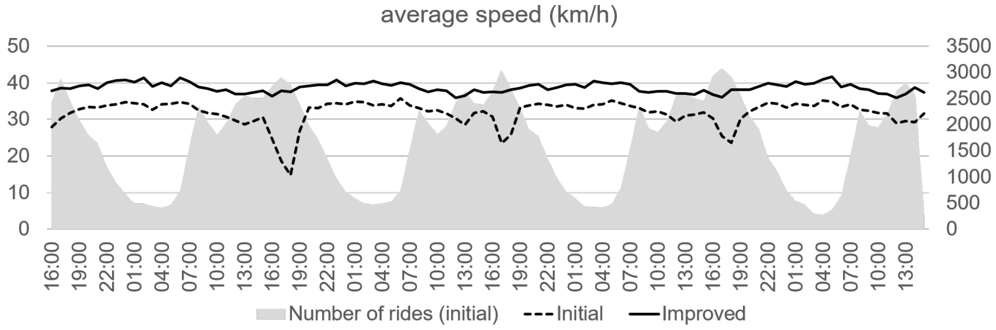

Figure 7: Comparison of the average speed before and after the recalculation of the duration of the phases of traffic signal schemes.

Furthermore, a model of the entire city of Winnipeg has been created (see Figure 6, right). This model consists of 5,428 nodes, 7,942 edges, 540 traffic signal controls, 152 areas within the road network, and 85 areas outside the road network, with a traffic load of about 1,550,000 vehicle rides in four days. For this model, the computational time to simulate an hour with a high traffic load is about 110 seconds, which is still acceptable. However, the model gets stuck in deadlock situations, where vehicles cannot move any further due to high traffic load with vehicles blocking the way for other vehicles to leave respective circles of road segments. As a consequence, no further results have been obtained so far. We have implemented methods to detect deadlock situations (which also occur in the real world) and are currently working on realistic approaches to resolve them.

Due to these deadlock situations, also the validation of the large model is not completed. A validation of larger models can be conducted by comparing typical travel times from other information sources, at least for the main roads. If the model shows unusual differences in the travel times, a few issues might have to be considered. One problem is that the number of lanes for left or right turns is not given in the OSM data for all intersections (Feldkamp and Strassburger 2014). A hint that a further analysis of the real-world layout of intersections is necessary is, e.g., that the number of vehicles making left turns at a respective intersection is much higher than for vehicles going straight. A validation of traffic signal schemes might also be useful. However, our approach to compute the respective signal times on the basis of the traffic load and general given settings should relate to the real-world situation quite well.

5 CONCLUSION AND FURTHER RESEARCH

In this paper, we have proposed an approach for automatically generating large road network models from OSM data, including the creation and parameterization of traffic signal controls. The traffic load is derived from trajectory cell phone data. As a result, detailed road network simulation models are given to obtain information about the average speed for the tracks of the road network on an hourly basis. This is needed to solve time-dependent vehicle routing problems, as they occur in the field of distributing goods within urban areas. Here, the respective statistical values are intended to be used for parameterizing tracks for supply chain simulation or optimization models that deal with the distribution of goods in urban areas. Of course, these models can also be further used for traffic simulation in general. The novelty of the work is derived from the integration of disparate and non-obvious data sets, and the results set a framework for future work. Our intention is also to highlight the emerging opportunities within a data culture to the multidisciplinary research associated with modeling without requiring a formal background in telecommunications or intelligent transportation systems. Other sources of vehicular traffic patterns that would be more appropriately termed crowd-sourced could be extracted through evolving taxi or ride sharing services as well as other delivery services. These are likely to be rich sources of data that will make supply chain simulation and modeling more robust and could be used to validate models.

Our approach could be adopted to other regions if respective cell phone data are available. However, the step of transforming respective raw data to the data structures needed will most likely need some preprocessing effort.

Our future research will concentrate on modeling sufficient recovery strategies for deadlocks, which currently occur in large models. Also, different schemes for traffic signal controls could be added with an automatic selection depending on the type of intersection and the expected traffic loads of the different directions. After a more thorough validation of larger models, we intend to return to classical supply chain simulation models and will integrate TDVRP formulations to make use of the generated data describing the time-dependent behavior of the underlying road network.

REFERENCES

Arellano, W. and I. Mahgoub. 2013. "TrafficModeler Extensions: A Case for Rapid VANET Simulation using, OMNET++, SUMO, and VEINS". In *2013 High Capacity Optical Networks and Emerging/Enabling Technologies*. December 11th-13th, Magosa, Cyprus, 109-115.

Backfrieder, C., C. F. Mecklenbruker, and G. Ostermayer. 2013. "TraffSim – A Traffic Simulator for Investigating Benefits Ensuing from Intelligent Traffic Management". In *Proceedings of the 2013 European Modelling Symposium*. November 20th - 22nd, Manchester, United Kingdom, 451-456.

Ebert, R. and C. Friedrich. 2018. "Simulation-based Evaluation of Urban Consolidation Centers Considering Urban Access Regulations". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 2827–2838. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Feldkamp, N. and S. Strassburger. 2014. "Automatic Generation of Route Networks for Microscopic Traffic Simulations". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 2848–2859. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gutenschwager, K. and K. Alicke. 2004. "Supply Chain Simulation mit ICON-SimChain". In *Logistik Management*, edited by T. Spengler, S. Voß, and H. Kopfer, 161–178. Heidelberg: Physica.

Gutenschwager, K., M. Rabe, M. Theile, and B. Wilhelm. 2018. "Comparison of Approaches to Encrypt Data for Supply Chain Simulation". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3084–3095. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ivanov, D. 2018. *Supply Chain Simulation and Optimization with AnyLogistix*. 2nd ed. Berlin: Berlin School of Economics and Law.

Li, Y., L. Zhao, Z. Yu, and S. Wang. 2017. "Traffic Flow Prediction With Big Data: A Learning Approach Based on SIS-Complex Networks". In *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. December 12th-15th, Chengdu, China, 550-554.

Lin, C., K. Choy, G. Ho, S. Chung, and H. Lam. 2014. "Survey of Green Vehicle Routing Problem: Past and Future Trends". *Expert Systems with Applications* 41(4/1):1118–1138.

Malandraki, C. and M. S. Daskin. 1992. "Time dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms". *Transportation Science* 26(3):185–200.

Matsumoto, T., K. Sakakibara, M. Ohara, and H. Tamaki. 2016. "Multi-agent Simulation with Mathematical Optimization of Urban Traffic Using Open Geographic Data". In *5th IIAI International Congress on Advanced Applied Informatics*. July 10th-14th, Kumamoto, Japan, 1109-1113.

Meyer, T., M. Trojahn, and S. Strassburger. 2013. "Using Crowdsourced Geographic Information from OpenStreetMap for Discrete Event Simulation of Logistic Systems". In *Proceedings of the 46th Annual Simulation Symposium*. 7th-10th April, San Diego, California, 2:1-2:8.

Neighbour, R., S. Mukhi, M. R. Friesen, R. D. McLeod, and M. Crowley. 2012. "Vehicular Traffic Modeling Governed by Cellular Phone Trajectories". In *2012 IEEE Vehicular Technology Conference (VTC Fall)*. 3rd-6th September 2012, Quebec City, Canada, 1-5.

Paessens, H. 1988. "The Savings Algorithm for the Vehicle Routing Problem". *European Journal of Operational Research* 34(3):336–344.

Rabe, M., A. Klueter, U. Clausen, and M. Poeting. 2016. "An Approach for Modeling Collaborative Route Planning in Supply Chain Simulation". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 2228–2238. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rabe, M., A. Klueter, and A. Wuttke. 2018. "Evaluating the Consolidation of Distribution Flows Using a Discrete Event Supply Chain Simulation Tool: Application to a Case Study in Greece". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 2815–2826. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Terzi, S. and S. Cavalieri. 2004. "Simulation in the Supply Chain Context: A Survey". *Computers in Industry* 53(1):3–16.

Wijedasa, S., C. Gunasekara, M. Laskowsk, M. R. Friesen, and R. D. McLeod. 2013. "Smartphone and Vehicular Trajectories as Data Sources for Agent-based Infection Spread Modelling". *Health Systems* 2(2):120–133.

Xia, F., A. Rahim, X. Kong, M. Wang, Y. Cai, and J. Wang. 2018. "Modeling and Analysis of Large-Scale Urban Mobility for Green Transportation". *IEEE Transactions on Industrial Informatics* 14(4):1469–1481.

Zilske, M., A. Neumann, and K. Nagel. 2011. "OpenStreetMap for Traffic Simulation". In *Proceedings of the 1st European State of the Map Conference*. July 15th-17th, Vienna, Austria, 126-134.

AUTHOR BIOGRAPHIES

KAI GUTENSCHWAGER is head of the Institute of Information Engineering at the Ostfalia University of Applied Sciences, Wolfenbüttel. He studied Business Informatics and received his doctoral degree from the Technical University of Braunschweig. He worked for SimPlan as a head of the office being a simulation expert in the field of logistics and supply chain management. From 2009 he was a professor for information systems in logistics at Ulm University of Applied Sciences and since 2013 he is a professor at Ostfalia University of Applied Sciences. His research is focused on IT-based methods for the simulation and optimization of production and logistics systems. His e-mail address is k.gutenschwager@ostfalia.de.

ROBERT D. MCLEOD is a Professor in Computer Engineering at the University of Manitoba. He received his doctoral degree also from the University of Manitoba in Electrical and Computer Engineering. His research interests include internet and smartphone application development, agent-based modeling and simulations, statistical inferencing, sensors, and systems. He has supervised over 60 M.Sc. and 13 Ph.D. students to completion in various areas. His email address is robert.mcleod@umanitoba.ca.

MARCIA R. FRIESEN is an Associate Professor in the Faculty of Engineering and Director, Centre for Engineering Professional Practice and Engineering Education. She is an NSERC Chair in Design Engineering for Sustainable Development and Enhanced Design Integration. Her research program focusses on engineering qualifications recognition for engineering newcomers, the scholarship of teaching and learning in engineering education, mHealth applications, and modeling and simulation for healthcare applications. Her email address is marcia.friesen@umanitoba.ca.