Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 256 (2025) 781–788

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2024

Implementing Dynamic Travel Time Calculation in EMS Simulations: Impacts on Prehospital Stroke Care and Transportation

Saeid Amouzad Mahdiraji[a,*], Marcus Juninger[a], Nicholas Narvell[a], Johan Holmgren[a], Radu-Casian Mihailescu[a], Jesper Petersson[b,c]

[a]*Department of Computer Science and Media Technology, Malmö University, Malmö 21119, Sweden*
[b]*Department of Healthcare Management, Region Skåne, Malmö 21428, Sweden*
[c]*Department of Neurology, Lund University, Lund 22242, Sweden*

Abstract

Preparing travel time data can be a time-consuming process, which greatly limits the flexibility of transport simulation models. In the current paper, we present an approach to integrate a routing engine locally in an existing modeling framework, hence enabling to dynamically calculate travel times in the constructed emergency medical services (EMS) simulation models. This integration eliminates the need for the pre-calculation typically required to prepare travel time data. Using the extended framework, we developed an EMS simulation model for stroke patients, which we applied in a scenario study to southern Sweden. This allowed us to evaluate the potential benefits of using dynamic travel time calculations in prehospital stroke care. The experimental results, supported by comparisons with pre-calculated travel times, confirm the effectiveness of our approach in integrating dynamic travel time calculations into the framework. Moreover, the results of our evaluation indicate that including this functionality in simulation models can provide more realistic results. Finally, our approach for local implementation of dynamic travel time calculations is faster and less restricted compared to using online services.

* Corresponding author. Tel.: +46-40-6657110.
E-mail address: saeid.amouzad-mahdiraji@mau.se

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies
10.1016/j.procs.2025.02.179

Keywords: Framework, Dynamic travel time, EMS, Travel data calculation, Simulation model.

1. Introduction

Transport simulation is an important tool for analyzing and planning transport systems, enabling, for example, the prediction of traffic patterns and the assessment of infrastructure projects. By simulating different transport scenarios, decision-makers can use simulation models to evaluate the impact of different policies and changes before real-world implementation. The dynamic nature of transport systems, characterized by changes in traffic conditions and unexpected road closures, necessitates the use of dynamic data to ensure the accuracy and relevance of simulation outcomes.

Before or during the simulation, different types of data, such as travel times, may need to be provided to the transport simulation model. Some methods for preparing travel times in transport simulation models involve calculating travel times as origin-destination matrices in advance for all potential locations within the study region. These pre-calculated matrices are then used in the simulation model during execution. However, such travel time data preparation approaches are complex and time-consuming since some of the data preparation tasks typically need to be performed manually. Moreover, due to the complex and long data preparation process, the travel data cannot be frequently updated and are instead used as static data. An important limitation of static travel times data is that they fail to account for real-time traffic conditions in the simulation models.

In response to these challenges, dynamic travel time calculations have been introduced to simulation models to provide more accurate and dynamic data that reflect the current transport conditions. The integration of dynamic travel time calculations into the transport simulation models allows the model to efficiently adapt and respond to variations in input, for example, resource availability and changes in traffic patterns, and to simplify the data preparation process. Another advantage is that, by using the dynamic approach, only the required travel data will be calculated. This integration would also improve the predictive accuracy of simulations [1]. In this regard, Hajinasab et al. [2] propose using online services to outsource the generation and calculation of data for transport simulation. This approach is advantageous as it eliminates the need to pre-calculate vast amounts of travel data. Huber and Rust [3] introduce an approach to locally calculate thousands of travel time requests within seconds. As these calculations are performed offline, their approach can effectively use the full computational power of the host system and support a large number of requests.

The need for dynamic travel data is crucial in emergency medical services (EMS) simulations, where the speed and efficiency of prehospital interventions can critically impact patient outcomes, for example, for acute conditions like stroke. EMS simulation models, used to optimize operations and improve patient care, may use pre-calculated, static, travel times. While these static times are useful for basic scenario planning, they fail to reflect the unpredictable and dynamic nature of real-world conditions that emergency vehicles (EVs) encounter. The integration of dynamic travel time calculations into EMS simulation models can be also positive, showing significant improvements in the accuracy of the models and their applicability for policy-making. Patel et al. [4] and Zhen et al. [5] demonstrate the limitations of using Geographic Information System and other static data sources for modeling pre-hospital and ambulance deployment times, which often lead to suboptimal resource positioning during peak traffic periods or other dynamic events. Juninger and Narvell [6] evaluate routing engines for dynamic travel time calculations used in the simulation of EV transportation. Juninger and Narvell demonstrate that the most suitable routing engine is the *Open Source Routing Machine with Contraction Hierarchies* (OSRM-CH)[1]. They also realize that locally deploying the instances of the routing engines leads to better performance than running them online.

Some studies use pre-calculated travel data to simulate prehospital stroke care and transportation [7, 8]. However, the impact of including dynamic travel times in EMS simulations, particularly for prehospital stroke care, remains largely unexplored. In this paper, we build on the approach by Juninger and Narvell [6] for dynamic travel time calculations. In particular, we integrate the OSRM-CH routing engine into an established EMS modeling framework,

1 https://github.com/Project-OSRM/osrm-backend/

used to construct EMS simulation models for different medical conditions, including stroke [7]. This integration enhances the framework's capability to calculate travel times dynamically, removes the need to pre-calculate travel data, and it aims to better align the simulation outcomes with actual EMS operations. We explore the benefits and limitations of our proposed approach through a scenario study applied to southern Sweden.

The rest of the paper is structured as follows. In Section 2, we present how we integrate dynamic travel time calculations into the modeling framework and then use them to construct a stroke simulation model [7]. Section 3 describes the scenario study, followed by the results and a discussion. Eventually, Section 4 concludes the paper.

2. Integrating Dynamic Travel Time Calculations in a Modeling Framework

In this section, we present our framework for building EMS simulation models, which followed by a description of the integration of the dynamically calculated travel times into the framework.

2.1. Modeling framework

In a prior study [7], we introduced a modeling framework designed to construct EMS simulation models. As illustrated in Fig. 1, the framework comprises two phases: model construction and model execution. In the construction phase, the framework uses different inputs, including geographical data, patient data, EV data, hospital data, and a care chain specification, to build a discrete event simulation model tailored to a specific EMS scenario. During the execution phase, the constructed model is executed using the input data, and the specified care chain is applied to each simulated patient. In the original version of the framework, pre-calculated travel times of EVs are given as a static input to the framework. Considering the potential advantages of using dynamic travel time calculations during simulation, we below describe how we incorporate this functionality into the framework.

There are five different types of transport activities in the framework that require travel time calculations: 1) travel from the current location of an EV to the patient's location, 2) travel from the patient's location to a hospital, 3) travel from a hospital to an EV site, 4) travel from the patient's location to an EV site, and 5) travel from a hospital to a special clinic. Each transport activity type requires a unique way of calculating travel times. Transport activity type 1 involves an initial step to determine which EV(s) should be dispatched to the patient's location. This decision-making process requires calculating the travel times from all EV sites to identify which EV is expected to arrive fastest at the patient location. However, performing such calculations from the current locations of all EVs to the patient's location using pre-calculated ambulance driving times would be complicated. Transport activity type 2 necessitates calculating travel times from the patient's location to all hospitals or special clinics to determine the nearest one. These two activities require the calculations of many-to-one and one-to-many trips, respectively. Transport activity types 3, 4, and 5 only require one-to-one travel time calculations. For example, for transport activity types 3 and 4, each EV is typically stationed at and returns to the same EV site. Also, for transport activity type 5, there is generally only one special clinic in each region. However, there could be different special clinics for different diagnoses and treatments.

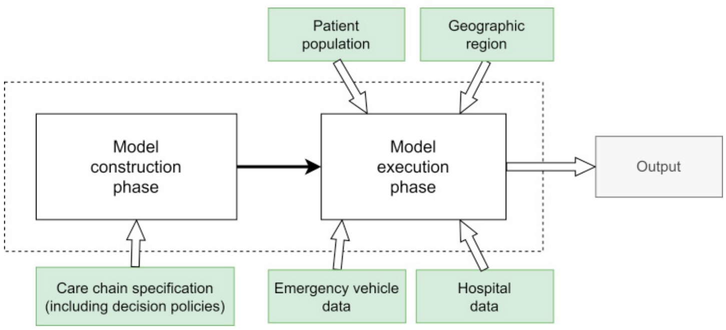

Fig. 1. Overview of the modeling framework proposed by Amouzad Mahdiraji et al. [7], where the provided input data is used to construct and to run the model.

2.2. Integrating dynamic travel time calculations into the framework

In the original version of the framework [7], we pre-calculated the travel times of EVs using the driving time generation feature of OpenStreetMap (OSM), which we accessed through the Openrouteservice toolbox in QGIS. OSM, created by the OpenStreetMap Foundation, is based on geographic data and provides a free editable world map [9]. Due to the limited number of requests allowed by OSM per run, we needed to manually divide the input data, including EV data, hospital data, and patient data, before processing it through the Openrouteservice toolbox. Subsequently, we merged all the calculated travel data in QGIS to compile the travel time data set. This method of calculating travel times proved to be inconvenient, time-consuming, and prone to errors, particularly when adapting the framework to new regions. In addition, using the pre-calculated travel data limits the framework in several ways. For example, it requires that EVs are always dispatched from their sites to the patients' locations. Hence, we could not account for the possibility of dispatching an EV from any location within the region. It also requires that an EV must return to its site from the hospital or patient's location to be ready for the next call. A dynamic approach for travel data calculation can overcome these limitations by allowing for more flexible and accurate travel time calculations. Due to the large number of travel time calculations required by the framework, the chosen approach should not restrict the number of travel times to be calculated.

To enable the framework to calculate the EV travel time dynamically during simulation, we include the approach by Juninger and Narvell [6] into the framework. Juninger and Narvell identify and evaluate a set of routing engines based on several criteria, such as license price, request limitations, matrix search ability, performance, and usability. Among the available routine engines, the authors identify that ORS, OpenSourceRoutingMachine (OSRM), and Valhalla offer unlimited requests for local deployment. For evaluation, the authors set up two so-called Docker instances for each routing engine: one hosted locally and one hosted at a separate location. Docker[1] is a platform that uses lightweight virtualization to package applications into containers, allowing them to run consistently across different computing environments. This containerization simplifies deployment and supports scalability by enabling multiple instances of the same application to operate simultaneously. They also develop a performance test suite to log statistics to calculate and return requests for each routine engine. The experimental results demonstrate that OSRM-CH outperforms other routing engines for local deployment. OSRM-CH is a C++-based routing engine that uses Contraction Hierarchies as its traversal algorithm. Furthermore, they integrate the OSRM-CH into a module[2] that makes it possible to dynamically calculate travel times.

In this paper, we utilize the module by Juninger and Narvell to locally integrate the OSRM-CH into the framework. In Fig. 2, we illustrate how the framework and the module interact to calculate the travel times during the simulation. The OSM data for the region of interest is downloaded from Geofabrik[3]. To enable the module to function, an instance of OSRM_CH, which is preloaded with OSM data, is configured and run locally in Docker. The module restructures the input data to align with the routing engine's interface before sending HTTP requests to the Docker container where the routing engine is running.

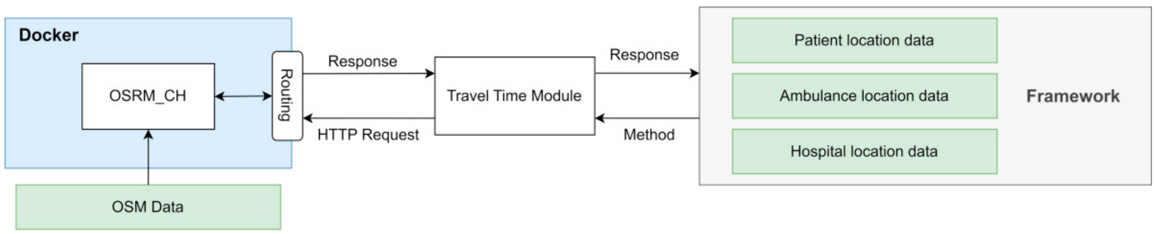

Fig. 2. An overview of the data flow and relationships between the framework and the module [6] for the dynamic travel time calculations.

1 https://www.docker.com/
2 https://github.com/Juninger/python-travel-time-module/
3 https://download.geofabrik.de/

The patient data, hospital data, and EV data, fed into the framework as input data, should include either a single coordinate (a pair of longitude and latitude) or a list of coordinates. The module handles three types of trip requests based on the transport activity types, as described in Table 1. Each request type has two input parameters: *origin* and *destination*, and the return value is a tuple containing the fastest travel time and the corresponding coordinates for the origin and destination, that is, *{travel_time, [long, lat], [long, lat]}*. Considering the transport activity types included in the framework, transport activity type 1 is modeled using the many-to-one request type, transport activity type 2 using the one-to-many request type, and transport activity types 3-5 using the one-to-one request type.

The proposed approach can dynamically calculate travel times for different scenarios applicable to the framework. All these calculations can occur during the framework's runtime based on the requested trip type, eliminating the need for any pre-calculated travel times.

Table 1. Description of each type of trip request available in the module by Juninger and Narvell [6].

Request type	Description
Many-to-one	The origin parameter takes a list of coordinates, that is, *origins: [[long, lat], [long, lat], ...]*. The destination parameter takes a list of a single coordinate, that is, *destination: [long, lat]*.
One-to-many	The origin parameter takes a list of a single coordinate, that is, *origin: [long, lat]*. The destinations parameter takes a list of coordinates, that is, *destinations: [[long, lat], [long, lat], ...]*.
One-to-one	The origin parameter takes a list of a single coordinate, that is, *origin: [long, lat]*. The destination parameter takes a list of a single coordinate, that is, *destination: [long, lat]*.

3. Scenario Study

We evaluated the extended framework by constructing a simulation model for prehospital stroke care, which we applied to Sweden's southern healthcare region (SHR). In the current section, we provide a brief introduction to stroke and describe the scenario study, as well as the framework configurations, before presenting the experiments.

Stroke is known as one of the main causes of death and disability worldwide. Ischemic stroke is the most common type of stroke, where thrombolysis is the standard treatment. Timely treatment is essential for the successful recovery of stroke patients; however, logistical challenges often delay the delivery of appropriate treatment. In addition, stroke patients are typically transported to an acute hospital for diagnosis and treatment, which can be time-consuming. However, earlier studies demonstrate that the use of mobile stroke units (MSUs) has the potential to significantly reduce the time to treatment for stroke patients. MSUs are specialized EVs equipped with a CT scanner that enables both the diagnosis and administration of thrombolysis directly on-site [10]. An MSU may be deployed either alone or alongside a regular ambulance (RA) in response to a stroke incident.

In the scenario study, we applied the proposed approach of integrating dynamic travel time calculations into the framework to SHR, illustrated in Fig. 3. As of 2023, the total population of SHR was 1,926,100 [11]. According to the Swedish Stroke Register (Riksstroke) [12], Sweden recorded over 20,000 annual stroke incidents in 2022, including 3,900 cases within SHR. Fig. 3 illustrates the locations of ambulance sites and acute hospitals in the SHR, marked by green triangles and purple circles, respectively, with each ambulance site assigned a unique ID. Currently, SHR does not have any MSUs in operation.

We conducted several simulation runs using the constructed simulation model for prehospital stroke care on a computer equipped with 32 gigabytes of RAM and an Intel(R) Core(TM) i7-8650U CPU operating at 1.90 gigahertz. The framework settings allow users to run experiments with or without using the dynamic travel time calculation functionality. In the simulation runs, we considered decision policies concerning: 1) the selection of which EV to be dispatched to the patient's location, 2) the choice between single dispatch and co-dispatch, and 3) the selection of the destination hospital. For EV selection, we adopted the *time to diagnosis* policy, where one or two EVs—referred to as single dispatch and co-dispatch, respectively—are chosen to minimize the expected time until diagnosis. For hospital selection, our choice was the *fusion* policy, allowing the patient to be transported either to the nearest hospital or to a special clinic based on the on-scene assessment (and diagnosis) and the distance from the patient's location to the special clinic. Regarding the dispatch type, the user can choose between *single dispatch* and *co-dispatch*. The single dispatch policy refers to the situation where either an RA or an MSU is dispatched to a stroke incident. Co-dispatching, on the other hand, refers to when an RA and an MSU collaborate until the suspected stroke patient receives appropriate treatment.

We utilized the same input data, MSU locations, and assumptions as in our previous study [13]. The geographic region, that is, the SHR, is divided into non-overlapping 1×1 km^2 subregions to facilitate the localization of patients, hospitals, and ambulance sites. The stroke patient data, generated using a Poisson distribution, represents a synthetic population distributed across the SHR (details are provided in a previous study [14]). Each patient in this population is characterized by specific attributes including incident time, location (coordinate), age, symptoms, diagnoses, and preferred treatment. We assumed that all patients represented in the simulation are ischemic stroke patients requiring diagnosis and, potentially, treatment either inside the MSU or at a hospital. Regardless of the diagnosis and treatment location, the patients would ultimately be transported to a hospital for further care. In all cases, the time to diagnosis or treatment is defined as the expected duration from the onset of a stroke until the patient receives diagnosis or treatment, either inside the MSU or at a hospital.

We took into account two scenarios that correspond to the dispatch policies: 1) single dispatch using RA only, which reflects the current situation in the SHR, and 2) co-dispatch. We evaluated these scenarios in relation to other decision policies, specifically ambulance selection and hospital selection, and under both dynamic and pre-calculated travel time conditions. In our experiments, the locations of the MSUs in the SHR were based on a previous study [15], in which we proposed an optimization model for the MSU placement problem to make a tradeoff between assuring equitable service and maximizing population coverage. We therefore set up one MSU in Alvesta (ID: 3), two MSUs in Malmö (ID: 38) and Alvesta (ID: 3), or three MSUs in Ängelholm (ID: 2), Alvesta (ID: 3), and Malmö (ID: 38) to conduct the experiments.

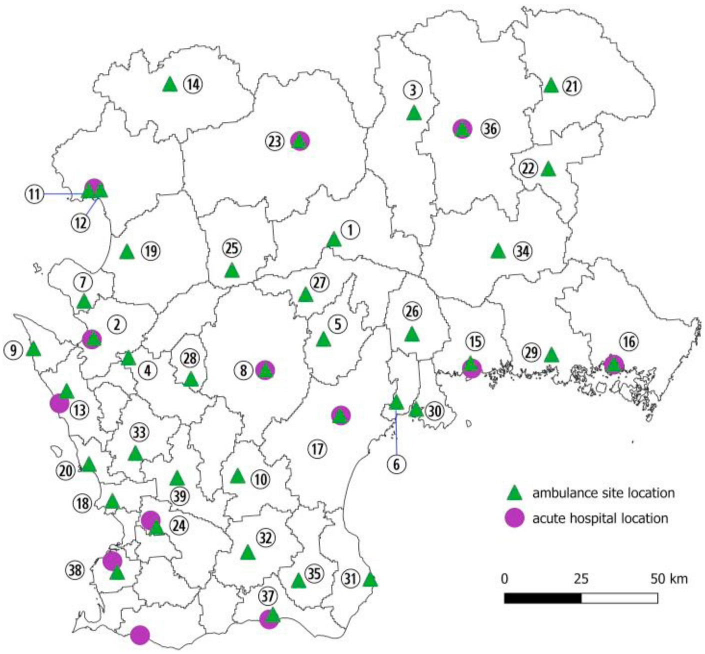

Fig. 3. Overview of Sweden's southern healthcare region, reproduced from [15]. The purple circles and green triangles represent the locations of acute hospitals and ambulance sites, respectively. The circled numbers indicate the corresponding ambulance site IDs.

3.1. Experimental results and discussion

In Table 2, we present the simulation results for the constructed model with and without dynamic travel time calculations concerning the average time to diagnosis and treatment for the described scenarios and policies. The results show that the use of co-dispatching, compared to the RA-dispatch, that is, the current situation in the SHR, is expected to reduce the time to diagnosis and treatment for stroke patients in the SHR, particularly when three MSUs are involved in co-dispatch operations, the expected reduction is 15.6 and 24.6 minutes for time to diagnosis and treatment, respectively.

According to Table 2, the time to diagnosis and treatment are approximately equal for dynamic and pre-calculated travel times, demonstrating that the proposed approach for integrating the dynamic travel time calculations of EVs into the framework functions as intended. However, using the dynamic travel time data results in a slightly longer time to diagnosis and treatment for all scenarios. The slight differences in the results arise because the dynamic travel times calculated using OSRM_CH reflect the most recent changes in road networks. In addition, the pre-calculated travel data was created using a different tool, namely the Openrouteservice toolbox in QGIS.

Furthermore, in Table 2, we compare the execution time of the simulation model applied to the SHR both with and without dynamic travel time calculations. The comparison shows that the running time using dynamic travel time calculations is higher since the framework, for calculating each travel activity, interacts with the module, which sends the requests to the Docker container. This process would take a few milliseconds for each request. Conversely, the time spent to calculate travel times data in advance for simulation is enormous, especially when the region under study is large, restricting the model's flexibility to consider different simulation scenarios. In addition, loading the pre-calculated travel data of SHR to the framework before starting the simulation would take about 22 minutes on average, which also needs to be taken into consideration. In this condition, calculating travel times data for a larger region and loading it into the model would be even more time-consuming. Hence, the use of pre-calculated data for real-world EMS situations is not beneficial. As a result, while adding dynamic travel time calculation functionality to the framework slightly increases the running time, it eliminates the need to prepare travel data for the simulation, which is a very time-consuming task.

To further evaluate our approach for dynamically calculating travel times in an EMS simulation model for stroke patients, we compare the running time of our approach, which is based on a local routing engine, with that of an online service, that is, the ORS Web API[1]. ORS Web API provides a free online routing service with a limited number of requests per minute and day, that is, 500 matrix route calculations. The volume of requests does not support our scenario study. However, to make a comparison between the performance of ORS Web API and our proposed approach, we randomly chose 500 pairs of coordinates between patient locations and ambulance locations in the SHR. Then, we divided these 500 requests into 13 batches, where each batch of 40 requests is sent to ORS Web API per minute to calculate the travel times. The experiment reveals that calculating 13 batches of a total of 500 requests sent to the ORS Web API took 316.60 seconds. In contrast, executing the same number of requests

Table 2. Comparison of running time (in hours), loading time for pre-calculated data (in hours), and the average time to diagnosis and treatment (in hours) for the considered dispatch policies both with and without dynamic travel times. SD: single dispatch, CD: co-dispatch. The numbers within the curly brackets show the ambulance site IDs (see Fig. 3).

MSU locations	Dispatch policy	Travel time calculation	Average time to diagnosis (h)	Average time to treatment (h)	Running time (h)	Loading time of pre-calculated data (h)
-	SD (RA)	Static	1.01	1.59	0.44	0.37
-	SD (RA)	Dynamic	1.02	1.60	1.05	-
{3}	CD	Static	0.98	1.54	0.43	0.36
{3}	CD	Dynamic	0.99	1.55	1.13	-
{3, 38}	CD	Static	0.83	1.29	0.45	0.36
{3, 38}	CD	Dynamic	0.84	1.31	1.15	-
{2, 3, 38}	CD	Static	0.74	1.18	0.46	0.36
{2, 3, 38}	CD	Dynamic	0.76	1.19	1.16	-

[1] https://openrouteservice.org/plans/

using the local routing engine used in this paper required 5.11 seconds. Consequently, the online service is 44 times slower than the local routing engine integrated into the framework.

4. Conclusions

We have explored the potential advantages of dynamic calculation of travel times in EMS simulation models. For this aim, we used the proposed approach by Juninger and Narvell [6] for dynamic travel time calculations and integrated it into a modeling framework for construction of EMS simulation models. The chosen routing engine was OSRM_CH. In our scenario study, we created an EMS simulation model for the stroke care chain, which we applied to southern Sweden, and compared the effectiveness of our approach with or without dynamic travel data.

The experimental results, supported by the results using pre-calculated travel data, showed that the proposed approach for dynamic travel data calculations works successfully in the EMS simulation model for stroke patients. The simulation output analysis demonstrated that although the use of pre-calculated travel data can lead to shorter execution times, the time to calculate travel data and to load it during simulation would be considerable. However, when using dynamic travel data calculations during simulation, we do not need to calculate the travel times in advance, which is a time-consuming task, especially when the region of study is vast. Also, it is more straightforward to adapt the framework to new regions only by providing the required input data, including ambulance locations, patient locations, and hospital locations, and a care chain specification. Comparing the local-based and online services for dynamic travel time calculations showed that the local implementation, used in our approach, is faster and less restricted regarding the number of requests compared to online services.

References

[1] Fujimoto, R., Barjis, J., Blasch, E., Cai, W., Jin, D., Lee, S., and Son, Y.-J. (2018) "Dynamic data driven application systems: research challenges and opportunities." *In 2018 Winter Simulation Conference (WSC)* 664-678 IEEE.

[2] Hajinasab, B., Davidsson, P., Holmgren, J., and Persson, J. A. (2017) "On the use of on-line services in transport simulation." *Transportation Research Procedia* 21: 208-15.

[3] Huber, S., and Rust, C. (2016) "Calculate travel time and distance with OpenStreetMap data using the Open Source Routing Machine (OSRM)." *The Stata Journal* 16 (2): 416-23.

[4] Patel, A. B., Waters, N. M., Blanchard, I. E., Doig, C. J., and Ghali, W. A. (2012) "A validation of ground ambulance pre-hospital times modeled using geographic information systems." *International journal of health geographics* 11: 1-10.

[5] Zhen, L., Wang, K., Hu, H., and Chang, D. (2014) "A simulation optimization framework for ambulance deployment and relocation problems." *Computers & Industrial Engineering* 72: 12-23.

[6] Juninger, M., and Narvell, N., (2023) "On the use of routing engines for dynamic travel time calculation within emergency vehicle transport simulation," *bachelor thesis* Malmö University.

[7] Amouzad Mahdiraji, S., Holmgren, J., Alshaban, A. a., Mihailescu, R.-C., Petersson, J., and Al Fatah, J. (2022) "A Framework for Constructing Discrete Event Simulation Models for Emergency Medical Service Policy Analysis." *Procedia Computer Science* 210: 133-140.

[8] Al Fatah, J., Alshaban, A. a., Holmgren, J., and Petersson, J. (2018) "An agent-based simulation model for assessment of prehospital triage policies concerning destination of stroke patients." *Procedia Computer Science* 141: 405-412.

[9] Open Street Map (2024) About OpenStreetMap. Available at: https://wiki.openstreetmap.org/wiki/About_OpenStreetMap (Accessed: April 1, 2024).

[10] Ebinger, M., Winter, B., Wendt, M., Weber, J. E., Waldschmidt, C., Rozanski, M., Kunz, A., Koch, P., Kellner, P. A., and Gierhake, D. (2014) "Effect of the use of ambulance-based thrombolysis on time to thrombolysis in acute ischemic stroke: a randomized clinical trial." *Jama* 311 (16): 1622-1631.

[11] Statistics Sweden (2024) demographic data 2023. Available at: https://www.scb.se/ (Accessed: April 1, 2024).

[12] The Swedish Stroke Register (Riksstroke) (2023) Annual Report From Rikssstrok: Stroke and TIA. Available at: https://www.riksstroke.org/sve/forskning-statistik-och-verksamhetsutveckling/rapporter/arsrapporter/ (Accessed: September 15, 2023).

[13] Amouzad Mahdiraji, S., Holmgren, J., Mihailescu, R.-C., and Petersson, J. (2024) "Simulation-based Analysis of Co-dispatching in Prehospital Stroke Care. " *In 15th International Conference on Ambient Systems, Networks and Technologies (ANT 2024)*.

[14] Amouzad Mahdiraji, S., Holmgren, J., Mihailescu, R.-C., and Petersson, J. (2022) "A Micro-Level Simulation Model for Analyzing the Use of MSUs in Southern Sweden." *Procedia Computer Science* 198: 132-139.

[15] Amouzad Mahdiraji, S., Holmgren, J., Mihailescu, R.-C., and Petersson, J. (2021) "An Optimization Model for the Tradeoff Between Efficiency and Equity for Mobile Stroke Unit Placement." *Innovation in Medicine and Healthcare: Proceedings of 9th KES-InMed 2021* 183-193.