The Korean Institute of
Communications and
Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 7 (2021) 481–486

www.elsevier.com/locate/icte

Improving emergency response in the era of ADAS vehicles in the Smart City

Nicola Capodieci*, Roberto Cavicchioli, Filippo Muzzini, Leonard Montagna

University of Modena And Reggio Emilia, Department of Physics, Informatics and Mathematics, Modena, Italy

Received 10 July 2020; received in revised form 23 February 2021; accepted 4 March 2021
Available online 26 March 2021

Abstract

Management of emergency vehicles can be fostered within a Smart City, i.e. an urban environment in which many IoT devices are orchestrated by a distributed intelligence able to suggest to road users the best course of action in different traffic situations. By extending MATSim (Multi-Agent Transport Simulation Software), we design and test appropriate mitigation strategies when traffic accidents occur within an existing urban area augmented with V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure) capabilities and Advanced Driving Assisted cars (ADAS). Further, we propose traffic congestion models and related mechanisms for improving the necessary time for emergency vehicles to respond to accidents.

Keywords: Smart City; Multi-agent systems; Simulation

1. Introduction

Nowadays, the automotive industry is heavily investing in self-driving vehicles and Advanced Driving Assistance Systems (ADAS). The automotive sector development must match the parallel evolution of information and communication technologies applied to urban development, i.e. the *Smart City*. A Smart City is an urban area enhanced with a pervasive technological infrastructure able to exploit the compute continuum among *edge*, *cloud* and *fog* [1] computing nodes. Such a processing power is applied to the large amount of data collected by IoT-powered devices such as smartphones, next-generation vehicles and embedded platforms installed in the surrounding city infrastructure [2]. Such a distributed system allows the system engineer to define the interaction mechanisms among ADAS-enabled vehicles and the city itself, so to improve the overall urban viability and quality of life. For instance, emergency vehicles response time is a paramount aspect for urban quality of life; it is defined as the interval between the time at which an emergency vehicle is dispatched and its arrival time at the location of the accident [3]. Minimizing such an interval can drastically increase the chance of survival for the individuals involved.

In such a context, the contribution of this paper is twofold: first, we propose re-routing algorithms aimed at improving the travel times for emergency services while responding to traffic accidents. Secondly, we assess the impact of such strategies on the travel times experienced by the other (non-emergency) road users. For this latter category, we will also study the impact of the variable percentage of ADAS-equipped vehicles over the total population of road users, as such vehicles are the ones capable of efficiently communicating with the surrounding smart city infrastructure.

Trivially, changes to real world urban viability must be carefully planned and tested before their actual deployment. Therefore, our experiments on emergency vehicle re-routing are evaluated through simulations. More specifically, we used MATsim to perform our tests. The MATSim simulator [4] is a mesoscopic simulation platform for urban transportation that takes inspiration from the Multi-Agent System (MAS) literature [5]. The baseline version of MATSim is only capable of simulating road users who cannot communicate to nearby cars or to the surrounding infrastructure. For this reason, in this paper we also present an extension to MATsim able to allow the system engineer to simulate the interactions among sets of connected vehicles within a Smart City.

* Corresponding author.
E-mail addresses: nicola.capodieci@unimore.it (N. Capodieci), roberto.cavicchioli@unimore.it (R. Cavicchioli), filippo.muzzini@unimore.it (F. Muzzini), 165689@studenti.unimore.it (L. Montagna).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.03.005

Our Smart City MATSim extension is composed of multiple modules to simulate cameras, sensors, servers and communication systems so that IoT-equipped vehicles are able to react to unexpected events such as traffic accidents. More specifically, emergency vehicles can communicate with the city infrastructure so to suggest rerouting plans to the connected vehicles. This approach drastically reduces traffic congestion that occurs within the ambulances' paths.

To prove the validity of our simulated traffic models and related routing strategies, we discuss an extensive set of experiments aimed at assessing the emergency vehicles' response time when moving from/to the accidents' locations.

2. Related work

MATSim [4] (Multi Agent Transport Simulation) is a multi-agent simulation software for traffic management in urban cities. Each agent is a software autonomous entity that represents a pedestrian, a passenger vehicle, an emergency vehicle or a public transportation vehicle. MATSim has been extensively used in multi-agent system research literature: the multi-agent system approach is a design pattern that focuses on the autonomicity of software entities able to act on behalf of or for simulating the needs of human users. In this context, MATSim serves as framework for transportation scenarios simulated with multi-agent systems methodologies. Others eligible simulators for urban mobility are present in literature [6]; some of them are open-source projects (e.g., MATSim, SUMO), whereas platforms like Vissim and AIMSUM are commercial software. Due to the nature of our experiments, we prefer the extensibility offered by open-source solutions. A plethora of documentation materials, examples and research applications are available for both SUMO and MATSim. Nevertheless, we argue that MATSim is able to provide a higher degree of flexibility, easiness to customize and use with respect to SUMO; our considerations closely match the conclusion reported in the previous work (e.g. [7,8]). We also highlight that MATSim performs better on large networks than SUMO and, given our specific needs, MATSim routing module takes into account a more suitable set of parameters for the road entities [9]. In summary, the open-source nature of MATSim, its extensible structure and its ability to perform simulations of large urban areas have convinced us to choose this platform over similar software.

The previous work on MATSim and its applications relates to study the efficiency of legacy transportation infrastructures, such as taxi services [10], car sharing [11] and obviously public transit systems [12]. Our work differs from the cited literature by focusing on the future of transport, hence by accounting for partially autonomous vehicles that are connected with a really existing infrastructure of smart sensors inside an average sized municipality in Northern Italy. Such an urban area is located in the city of Modena and it is called MASA (Modena Automotive Smart Area, depicted in Fig. 1(a)).

2.1. MASA: Modena Automotive Smart Area

The MASA project aims at becoming a pioneering nation-wide research effort on control, standardization and certification methodologies for Smart-Cities.[1] MASA, which started in 2017, is exploited by many European Projects as a test bed for interconnected road users and infrastructure, artificial intelligence on edge devices and ADAS capabilities for innovative vehicles. A private 4G-LTE Antenna is installed in the area: this allows for low latency communication among vehicles, edge and fog nodes. MASA is also equipped with 50 cameras, both traditional (20) and smart (30): a smart-camera allows for local processing of artificial vision algorithms, such as object detection and vehicle counting. The European projects that relates to the MASA area made available a small fleet of ADAS vehicles. These vehicles are equipped with cameras, LiDARs, GPS and a computing board: these are full fledged mobile platforms that perceive the environment and communicate with the computing infrastructure of the city, namely 4 fog nodes and 6 cloud servers. The CLASS project[2] is building a software architecture able to answer in real time the road user needs, therefore the communication protocol that has been deployed in the area is based on a low latency UDP transmission over 4G using binary compressed data.[3] A schematic of the actors involved and the communication infrastructure is shown in Fig. 1(b).

We exploit the open-source nature of MATSim to extend it in order to account for the complex V2V and V2I interactions within the MASA that are made possible by the CLASS software architecture. More specifically, we tune our simulations within the MASA aiming to design and test real-time rerouting strategies for minimizing emergency vehicles response time during traffic accidents.

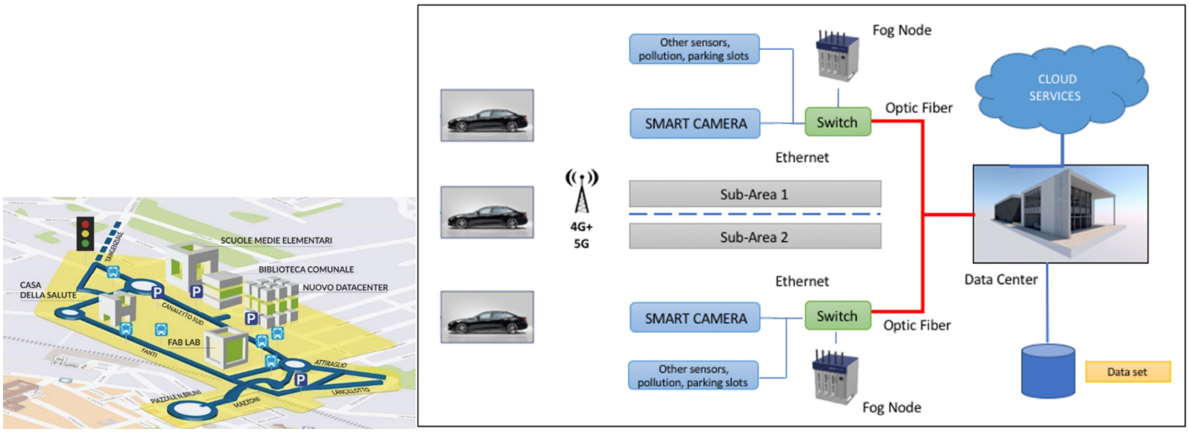
2.2. Extending MATSim for simulating smart cities

The baseline version of MATSim does not enable the interaction among vehicles and the city infrastructure; moreover, vehicles' daily plans and corresponding routes are fixed at initialization time and do not change during the simulation runtime. Since this is a limitation that would prevent developers from testing emergency response mechanics, a MATSim extension [13] has been used for allowing the developers to implement dynamic agent routing. This extension enables MATSim to exploit dynamic plans in all those situations in which the dynamic management of vehicle's routes [14] is a necessary aspect for the simulation. Starting from the MATSim extension detailed in [14], we further developed the dynamic agent planning module so as to be able to dynamically adjust agents' routes according to messages exchanged within the city infrastructure. The city infrastructure is implemented as an extension of MATSim; it is able to listen to the simulation's

[1] https://www.comune.modena.it/modena-smart-community/smart-mobility/masa-modena-automotive-smart-area.

[2] https://class-project.eu/.

[3] https://git.hipert.unimore.it/rcavicchioli/masa_protocol/tree/cmakefix.



(a) MASA within the city of Modena

(b) MASA connectivity setup

Fig. 1. Details of the Modena Automotive Smart Area: road network, communication and computing infrastructures.

events in order to obtain a global view of the city state. Position and communication capabilities of the involved vehicles are part of this observed state. In summary our developed extension allows the vehicles to communicate with the city infrastructure and thus, dynamically change their behaviour based on the information given by city infrastructure itself; a first version of our extension has been successfully deployed in the previous work on Smart City simulations in MATSim [15, 16].

3. Simulation model

The interested Smart City area is represented as a network Γ of connected nodes: formally, this is represented as graph $G = (V, E)$, in which vertices represent traffic intersections, and edges are the links representing the connecting roads. The graph is directed so that we are able to simulate the permitted flow directions. In our MATSim extension, we were able to enhance the baseline urban area representation by adding turning bans and one-way streets in accordance to the really existing area we elect for our simulation. For each link in the road network we associate a FIFO queue, so that whenever an agent enters that link, it is automatically inserted in the corresponding queue. It is trivial to understand that a link characterized by an accident will soon fill its queue, causing a congestion that will reduce or even set to zero the speed of the vehicles involved (see Section 3.2).

3.1. Road users as agents

Several kinds of agents are present in our model and each agent features a starting location, one or more activities (in terms of places to reach in specific time intervals), and an ending location. More formally, to each agent A we associate a path $P \subset \Gamma$.

The *classic agent* is the MATSim standard road user who is unable to communicate with the other vehicles or with the surrounding infrastructure. Such an agent can be aware of a traffic accident only when it is located in close proximity to the accident's location. In our model, the classic agent further specializes to be either an *autochthonous* or an *allochthonous* road user. The first type describes the behaviour of a road user that is familiar with the city: hence, in case of a traffic accident it is able to find an optimal rerouting strategy without needing help from the Smart City infrastructure. The optimal route is computed by exploiting the Dijkstra algorithm on the road network, provided that the accident link has been removed from the road network. In contrast, the allochthonous agent will attempt to find alternative routes by randomly try nearby roads. More specifically, if an allochthonous agent with a specified ending $destination$ sees an accident at location $siteOfAccident$, a list of connected links from that site ($listOfConnectedLinks$) is then constructed. A random outgoing link is therefore extracted from that list and checked against the link corresponding to the accident's site. In case the selected link is different from the accident link, the agent will move towards the extracted link and checks if the new arrival link belongs to its original route to destination ($originalPath$). In this case, the rerouting plan is complete. Otherwise this process is iterated. This is explained in Algorithm 1.

```
if $destination == siteOfAccident$ then
    enqueue($siteOfAccident$)
    return
end
while $true$ do
    if $siteOfAccident$ is the only member of $listOfConnectedLinks$ then
        enqueue($siteOfAccident$)
        return
    end
    else
        $agentPosition \leftarrow$ link $\leftarrow$ pickRnd($listOfConnectedLinks$)
        update($listOfConnectedLinks$)
        if $link \in originalPath$ then
            return reroutingPlan
        end
    end
end
```

Algorithm 1: Allochthonous agent rerouting plan

The *smart agent* represents a road user driving an ADAS-capable vehicle, hence able to communicate with other ADAS vehicles and the Smart City infrastructure. More specifically, this kind of vehicle informs a central authority about its route. The central authority is then able to immediately inform the agent that an accident occurred in a link within the route previously communicated by the agent. The agent is therefore able to compute the optimal route even if that accident is far from the agent line of sight.

The *emergency vehicle* represents an agent acting on behalf of a critical emergency response organization, e.g. police, fire brigade or ambulance. In this work, we aim to prioritize their ability to move within the city during accidents. An emergency vehicle entering a queue of a link is modelled as an occupant of a *Seepage queue*. Instead of lining up [17], a Seepage queue model allows emergency response vehicles to seep through the queue of both smart and classic vehicles. Also, we assume that this agent can communicate with other smart agents and with the city infrastructure.

Even if a smart vehicle can immediately reroute as soon as a central authority is aware of an accident, V2V communication can be exploited to avoid smart agent routes to conflict with the emergency vehicle routes to/from the accident site. Given that the path of the emergency vehicle ($emergencyPath$) is composed of N links, we define a *Look Ahead Window* (LAW) of links represented by the next M links within the emergency agent path starting from its current position. The LAW is communicated to the nearby smart agents, so that they can compare this newly received list of links with a subset (P^*) of their current plan (P); such a subset is composed of the first K subsequent element starting from the current position of the smart agent. If a potential conflict is detected then the smart agent will reroute (i.e. by removing the overlapping links and recompute Dijkstra). Implementation wise, constraints in Eq. (1) apply.

$$\begin{aligned}
&M = |LAW|,\ \ N = |emergencyPath| \\
&LAW \subset emergencyPath \\
&K = |P^*|,\ \ P^* \subset P \qquad (1)\\
&K \leq M \ll N, M > \epsilon \\
&if\ \ P^* \cap LAW\ \ \neq \emptyset\ then\ conflict\ is\ detected
\end{aligned}$$

The formulae in (1) constraint the Look Ahead Window to have a considerably small cardinality compared to the whole emergency vehicle path. This is because a big granularity for path conflict detection would imply unnecessary large rerouting operations for the smart agent. On the other hand, LAW cardinality must not be excessively small: if, for instance, $M = 1$ this would imply having a precise knowledge on when vehicles will reach a specific position. Although this is feasible in a simulated environment, such a precise information will be next to impossible to be collected in a real-life traffic scenario.

3.2. Agent speed calculation

In order to model the speed of the cars within a queue, we exploit the Greenshield Linear model (Eq. (2)):

$$v = max\left(v_m - \frac{v_m\ d}{c}, min_{speed}\right) \qquad (2)$$

In Eq. (2), v is the current velocity, v_m is the maximum velocity for that category of vehicle, d is the current vehicle density of that link [vehicles\mile or kilometre] and c is the link capacity (as in vehicle count). Eq. (2) implies the vehicles to set their velocities to zero ($min_{speed} = 0$) if the link's capacity is exceeded (i.e. congestion). v_m is set according to the maximum speed allowed by the local traffic authority, which implies allowing emergency agents to reach higher speeds than regular road users. In order to emulate the seeping behaviour of the emergency vehicles, for such a class of agents we impose $min_{speed} > 0$. Moreover, we also decrease the non-emergency vehicle speed in non-congested links when an emergency vehicle is passing through that road: this simulates the likely behaviour of the road users to decrease their speed and/or to temporarily move to the side of the road to let the emergency responder to pass more easily. This v^* speed is therefore calculated as in Eq. (3).

$$v^* = min\left(R\ e^{-\frac{R}{l_{link}}}, v\right) \qquad (3)$$

In Eq. (3) we define $v*$ as the minimum between v as in Eq. (2) and a decreasing exponential function that depends on R (a constant walking-pace speed) and l_{link}, which is the length of the interested link. Hence, this slowdown is inversely proportional to the length of the road to simulate that the emergency vehicle effect does not necessarily propagate to the whole interested street.

4. Experiments and simulation

First, a representation of the MASA area street network was created. We used several plugins to import to MATSim the original map from the OpenStreetMap (OSM)[4] database and then we augmented the network with additional information such as capacities for the road links, turning bans and parking spaces. The MASA area is a 1-square Kilometre wide area centred around coordinates (44.65632, 10.93150) in OSM. The size of the simulated population was estimated using regional traffic flow data made available online[5]; using such data we estimated a rush-hour road users population between 10 000 and 20 000 agents. Constants in Eqs. (2) and (3) are set to $v_m = 50$ km/h, $min_{speed} = 0$ for non emergency vehicles, $v_m = 70$ km/h, $min_{speed} = 10$ km/h for emergency vehicles and $R = 5$ km/h. We highlight that maximum speed values are mandated by the Italian law. LAW cardinality for smart agent rerouting is set to 15, whereas $K = 8$ from Eq. (1).

For each experimental scenario we provide a XML file as required as input by MATSim. In such a file, we describe the initial plans for the agents, i.e. their departure and arrival locations. Plan are initially set for all the interested agents related to all the different experimental scenarios so to that the agents plans remain the same while we simulate different behaviours. Agents' plans are constructed taking two random locations on the MASA map: these two places represent home and work locations. The agents perform two trips in a day, first from home to work, then from work to home. The departure time from home and work is chosen following a normal distribution with peak on typical rush hours (8:00 AM–9:00 AM for home departure and 06:00 PM–07:00 PM

[4] https://www.openstreetmap.org/.

[5] https://servizissiir.regione.emilia-romagna.it/FlussiMTS/.

Table 1
Ambulance average travel times with different kinds of agents.

	Smart agent percentage	Ambulance travel time [s]	% difference from Scenario 1
Scenario 1	0	109	–
Scenario 2	33	102	6.4
Scenario 3	50	89	18.3
Scenario 4	66	80	26.6
Scenario 5	100	69	36.7

Table 2
Normalized travel times of different kinds of agents.

Accident scenario	Classic	Allochthonous	Autochthonous	Smart
Prev. intersection has multiple outgoing links	1	0.7385	0.6906	0.6724
Prev. intersection has no other outgoing links	1	1	1	0.6392

for departing from work). With regard to traffic modelling, simulated vehicles' routes undergo a process of calibration for depicting realistic traffic situations [18,19]. It is important to highlight, however, that the purpose of this paper is not to provide a realistic reconstruction of the traffic in the MASA area. More specifically, we are interested in the mechanisms behind emergency response while varying the capabilities of the involved vehicles within a relatively restricted city area. Therefore, we argue that once one-way streets, turning bans, street lane directions/capacities extracted from OpenStreetMap and a reasonable distribution of peak/rush hours are set for the simulation, this constitutes a proper environment for studying our proposed emergency response strategies.

4.1. Emergency vehicles response time

In this set of experiments, we are interested in observing how fast an ambulance (modelled as an emergency response agent) can travel to/from the site of an accident within the MASA. From the model described in Section 3 it is trivial to infer that ambulances travel time will be influenced by the percentage of smart agents over the total number of road users involved in our simulations. Considering this ratio as an experiment parameter is also useful for studying urban viability in the transitional period from exclusively human-driven vehicles to exclusively ADAS vehicles [15]. It is reasonable to assume that both smart and non-smart vehicles will coexist until next-generation cars will completely overtake the classic transportation means.

To determine the ambulance travel time with an increasing number of smart agents, multiple scenarios were run. In all these scenarios, we report the average travel time over a set of 30 ambulance trips. The first scenario consisted of 12 000 agents with a scaling number of smart agents and results are depicted in Table 1.

In Table 1 it is shown that the average response time of the emergency vehicles dramatically improves when the percentage of smart agents exceeds the 50% over the total number of road users. If all agents where representing ADAS-capable vehicles, decrease in response time would amount to 36.7%. Another key variable in determining the average response time is the population size of road users within the interested area. We performed additional tests by increasing the number of agents to 16 000, hence getting closer to the inevitable congestion of the MASA area we modelled in our simulation. In such a scenario, the response time improvement within a population characterized by 75% of smart agents would range from 4.3% to 8.33% with respect to the baseline scenario of no smart agents at all.

4.2. Accidents' impact on non-emergency agents

In this section we observe the impact on travel times on the agents when accidents and related road blocks occur within the MASA. As highlighted in Section 3.1, regular road users are modelled as *classic*, *allochthonous*, *autochthonous* and *smart* agents. The first ones are the MATSim baseline implementation of the road users who will make no attempts to avoid the road block caused by the accident. This kind of agent will serve as a baseline for our proposed model. In this experiment, we force accidents in random spots within the MASA and in the agents' travel plans. We then record their average travel times according to the category of agent they belong. In these experiments, we assume the totality of the agent population to belong to the indicated categories. Another aspect to consider in such an experiment is whether the link preceding the blocked road has more than one outgoing links. Having no outgoing links other than the accident site will likely force most of the road users to get stuck in the interested blocked road, whereas having the possibility to reroute to other links once the accident is in the driver's line of sight can help the road user to avoid being blocked. Experiments' results are visible in Table 2.

In Table 2 we can see that, when it is possible to reroute at the intersection before the accident link, travel times can improve from 26.1% (allochthonous agents) up to 33.6% (smart agent). The improvements for the allochthonous agents tell us that a more realistic approach to model an accident will lead to less pessimistic traffic estimation compared to the MATSim baseline approach. More specifically, the regular driver will attempt to find alternative routes without the help of the smart city; it does that even by accepting the risk to lose its way. While the performance difference between an allochthonous and an autochthonous agent can be explained by highlighting that this last category of agents cannot lose its way, in the multiple outgoing link scenario, there is little difference in improvements over travel times by an autochthonous agent and a smart vehicle (about 2%). The smart agent advantage gets more dramatic in the second scenario, in which the intersection before the accident site does not allow the agents to perform last minute rerouting operations. More specifically, allochthonous and autochthonous agents are able to react to the accident when it is too late, hence their performance matches the one of the classic MATSim agent, whereas the smart agent is immediately made aware of the accident by the city infrastructure. This leads to noticeable improvements for the ADAS-capable vehicles we modelled in our simulations.

5. Conclusion and future work

In this paper we extended the MATSim urban traffic simulator so as to be able to test and experiment different viability models in case of traffic accidents in the Smart City. While the previous traffic congestion mitigation techniques [20,21] assumed that every road users can be informed by traffic disruptions so as to immediately reroute to avoid congestion, we investigated the impact of different kinds of agents with different capabilities to communicate with nearby vehicles and the surrounding city infrastructure. Our proposed mechanisms managed to reduce emergency vehicle response time up to 36.7%. Reducing the time taken for ambulances to/from the accident sites is paramount to increase the chance of survival for the involved road users [22,23]. Moreover, our MATSim extension is also able to simulate other common scenarios in transportation. For instance, in this recently published article [16] our MATSim extension is used to simulate algorithms for improving waiting times at street intersections. This is made possible by implementing a novel approach to traffic lights control in which smart vehicles can communicate to nearby vehicles and traffic lights. We are also in the process of further extending our work for simulating and proposing mechanisms for vehicle pollution monitoring and smart parking systems. We argue that having a traffic simulator able to implement V2V and V2I communication will enable the user to test and simulate any transportation scenario that could benefit from a direct interaction with nearby vehicles and the city infrastructure.

CRediT authorship contribution statement

Nicola Capodieci: Conceptualization, Methodology, Writing. **Roberto Cavicchioli:** Writing - review & editing, Visualization, Writing - original draft. **Filippo Muzzini:** Software, Investigation, Supervision. **Leonard Montagna:** Software, Investigation, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the EU H2020 program under the CLASS project, Grant No. 780622.

References

[1] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, G.-J. Ren, Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems, IEEE Wirel. Commun. 23 (5) (2016) 120–128.

[2] E. Quiñones, M. Bertogna, E. Hadad, A.J. Ferrer, L. Chiantore, A. Reboa, Big data analytics for smart cities: the h2020 class project, in: Proceedings of the 11th ACM International Systems and Storage Conference, 2018, pp. 130.

[3] J.D. Mayer, Response time and its significance in medical emergencies, Geogr. Rev. (1980) 79–87.

[4] K. W. Axhausen, A. Horni, K. Nagel, The Multi-Agent Transport Simulation MATsim, Ubiquity Press, 2016.

[5] J. Ferber, G. Weiss, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Vol. 1, Addison-Wesley Reading, 1999.

[6] P.M. Ejercito, K.G.E. Nebrija, R.P. Feria, L.L. Lara-Figueroa, Traffic simulation software review, 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA) (2017) 1–4.

[7] M. Saidallah, A. El Fergougui, A.E. Elalaoui, A comparative study of urban road traffic simulators, in: MATEC Web of Conferences, Vol. 81, EDP Sciences, 2016, p. 05002.

[8] M. Maciejewski, K. Nagel, Towards multi-agent simulation of the dynamic vehicle routing problem in matsim, International Conference on Parallel Processing and Applied Mathematics (2011) 551–560.

[9] D.F. Allan, A.M. Farid, A benchmark analysis of open source transportation-electrification simulation tools, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015, pp. 1202–1208.

[10] M. Maciejewski, K. Nagel, Simulation and Dynamic Optimization of Taxi Services in MATSim, VSP Working Paper 13-0, TU Berlin, 2013, Transport Systems Planning and Transport Telematics, 2013.

[11] F. Ciari, M. Balac, K.W. Axhausen, Modeling carsharing with the agent-based simulation matsim: state of the art, applications, and future developments, Transp. Res. Rec. 2564 (1) (2016) 14–20.

[12] M.M. Oliveros, K. Nagel, Automatic calibration of agent-based public transit assignment path choice to count data, Transp. Res. C 64 (2016) 58–71.

[13] M. Maciejewski, K. Nagel, Towards multi-agent simulation of the dynamic vehicle routing problem in matsim, in: International Conference on Parallel Processing and Applied Mathematics, Springer, 2011, pp. 551–560.

[14] F. Zwick, K.W. Axhausen, Analysis of ridepooling strategies with matsim, in: 20th Swiss Transport Research Conference (STRC 2020)(Virtual), IVT, ETH Zurich, 2020.

[15] G. Cabri, L. Gherardini, M. Montangero, F. Muzzini, About auction strategies for intersection management when human-driven and autonomous vehicles coexist, Multimedia Tools Appl. (2021) 1–16.

[16] F. Muzzini, N. Capodieci, M. Montangero, Exploiting traffic lights to manage auction-based crossings, in: Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good, 2020, pp. 199–204.

[17] A. Agarwal, G. Lämmel, Modeling seepage behavior of smaller vehicles in mixed traffic conditions using an agent based simulation, Transp. Dev. Econ. 2 (2) (2016) 8.

[18] T. Toledo, M.E. Ben-Akiva, D. Darda, M. Jha, H.N. Koutsopoulos, Calibration of microscopic traffic simulation models with aggregate data, Transp. Res. Rec. 1876 (1) (2004) 10–19.

[19] R. Balakrishna, C. Antoniou, M. Ben-Akiva, H.N. Koutsopoulos, Y. Wen, Calibration of microscopic traffic simulation models: Methods and application, Transp. Res. Rec. 1999 (1) (2007) 198–207.

[20] T. Cox, P. Thulasiraman, A zone-based traffic assignment algorithm for scalable congestion reduction, ICT Express 3 (4) (2017) 204–208.

[21] J. Li, N. Ferguson, A multi-dimensional rescheduling model in disrupted transport network using rule-based decision making, Procedia Comput. Sci. 170 (2020) 90–97.

[22] R. Sánchez-Mangas, A. García-Ferrrer, A. De Juan, A.M. Arroyo, The probability of death in road traffic accidents. How important is a quick medical response?, Accid. Anal. Prev. 42 (4) (2010) 1048–1056.

[23] C.S. Lim, R. Mamat, T. Braunl, Impact of ambulance dispatch policies on performance of emergency medical services, IEEE Trans. Intell. Transp. Syst. 12 (2) (2011) 624–632.