

Lecture Notes in Networks and Systems 253

Juan M. Corchado
Saber Trabelsi *Editors*

Sustainable Smart Cities and Territories

 Springer

Lecture Notes in Networks and Systems

Volume 253

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA,
School of Electrical and Computer Engineering—FEEC, University of Campinas—
UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering,
Bogazici University, Istanbul, Turkey

Derong Liu, Department of Electrical and Computer Engineering, University
of Illinois at Chicago, Chicago, USA, Institute of Automation, Chinese Academy
of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering,
University of Alberta, Alberta, Canada, Systems Research Institute,
Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering,
KIOS Research Center for Intelligent Systems and Networks, University of Cyprus,
Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong,
Kowloon, Hong Kong

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at <http://www.springer.com/series/15179>

Juan M. Corchado · Saber Trabelsi
Editors

Sustainable Smart Cities and Territories

 Springer

Editors

Juan M. Corchado
Department of Computing Science
Universidad Salamanca
Salamanca, Spain

Saber Trabelsi
Texas A&M University at Qatar
Doha, Qatar

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-030-78900-8 ISBN 978-3-030-78901-5 (eBook)
<https://doi.org/10.1007/978-3-030-78901-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

A Novel Model for Detection and Classification Coronavirus (COVID-19) Based on Chest X-Ray Images Using CNN-CapsNet 187
 Dahdouh Yousra, Anouar Boudhir Abdelhakim, and Ben Ahmed Mohamed

Distributed Platform for the Extraction and Analysis of Information 200
 Francisco Pinto-Santos, Niloufar Shoeibi, Alberto Rivas, Guillermo Hernández, Pablo Chamoso, and Fernando De La Prieta

Intelligent Development of Smart Cities: Deepint.net Case Studies 211
 Juan M. Corchado, Francisco Pinto-Santos, Otman Aghmou, and Saber Trabelsi

Applications of AI systems in Smart Cities (APAISC)

Intelligent System for Switching Modes Detection and Classification of a Half-Bridge Buck Converter 229
 Luis-Alfonso Fernandez-Serantes, José-Luis Casteleiro-Roca, Paulo Novais, and José Luis Calvo-Rolle

A Virtual Sensor for a Cell Voltage Prediction of a Proton-Exchange Membranes Based on Intelligent Techniques 240
 Esteban Jove, Antonio Lozano, Ángel Pérez Manso, Félix Barreras, Ramon Costa-Castelló, and José Luis Calvo-Rolle

Intrusion Detection System for MQTT Protocol Based on Intelligent One-Class Classifiers 249
 Esteban Jove, Jose Aveleira-Mata, Héctor Alaiz-Moretón, José-Luis Casteleiro-Roca, David Yeregui Marcos del Blanco, Francico Zayas-Gato, Héctor Quintián, and José Luis Calvo-Rolle

Smart Mobility for Smart Cities (SMSC)

Infrastructure for the Enhancement of Urban Fleet Simulation 263
 Pasqual Martí, Jaume Jordán, Fernando De la Prieta, Holger Billhardt, and Vicente Julian

Modern Integrated Development Environment (IDEs) 274
 Zakieh Alizadehsani, Enrique Goyenechea Gomez, Hadi Ghaemi, Sara Rodríguez González, Jaume Jordan, Alberto Fernández, and Belén Pérez-Lancho

Smart Cyber Victimization Discovery on Twitter 289
 Niloufar Shoeibi, Nastaran Shoeibi, Vicente Julian, Sascha Ossowski, Angelica González Arrieta, and Pablo Chamoso



Infrastructure for the Enhancement of Urban Fleet Simulation

Pasqual Martí¹(✉), Jaume Jordán¹, Fernando De la Prieta², Holger Billhardt³,
and Vicente Julian¹

¹ Valencian Research Institute for Artificial Intelligence (VRAIN),
Universitat Politècnica de València, Camino de Vera S/n, 46022 Valencia, Spain
pasmargi@vrain.upv.es, {jjordan,vinglada}@dsic.upv.es

² BISITE Research Group, University of Salamanca, Calle Espejo s/n. Edificio
Multiusos I+D+i, 37007 Salamanca, Spain
fer@usal.es

³ Centre for Intelligent Information Technologies (CETINIA),
University Rey Juan Carlos, 28933 Móstoles, Madrid, Spain
holger.billhardt@urjc.es

Abstract. When it comes to urban fleet simulation, there are many factors which determine the quality of the outcome. Without real-world data on which to ground the setup, the results are not guaranteed to be useful. In addition, the coordination mechanisms for agents must be flexible and give the chance to agents to act following their own interests, as most of the urban traffic system users do. In this work we present an infrastructure for the simulation of urban fleets which deals with two challenges: realistic data generation, and self-interested agent coordination. Our infrastructure aims to ease the setup and execution of more realistic simulations in the urban traffic domain.

Keywords: Simulation · Transportation · Electric vehicle · Planning · Smart city · Urban fleets

1 Introduction

With more than half of the world's population living in cities, the list of challenges for keeping them sustainable has grown. “A smart sustainable city is an innovative city that uses ICTs (Information and Communication Technologies) to improve quality of life, the efficiency of urban operations and services and competitiveness while ensuring that it meets the needs of present and future generations concerning economic, social, environmental and cultural aspects”¹. In this line, new concerns have awakened among citizens and city councils. On the one hand, they want to reduce air pollution by promoting the use of bicycles,

¹ This definition was provided by the International Telecommunication Union (ITU) and United Nations Economic Commission for Europe (UNECE) in 2015.

public transport and even electric vehicles instead of the conventional gasoline-powered car. On the other hand, the existence of green areas throughout the city is valued; areas that beautify the appearance of the city and are related to a better quality of life. All of this seems to have influenced urban traffic, making it evolve to focus on the people rather than the vehicles. For instance, many municipalities are restricting the traffic inside their town's center, increasing the space available for pedestrians to walk, as well as air quality.

The urban traffic system, which was already complex, is therefore becoming more entangled. The traffic interactions that arise are not trivially sorted out which causes experts to be constantly researching for new ways of optimizing the traffic flow in urban settlements. Meanwhile, cities are evolving into smart cities that control parameters such as traffic status, the influx of people in different areas or on public transport and even the quality of their air, in real time. Then, as more city services become intelligent, we have more and more data that we can use both to better characterize the urban traffic problems and to advance in their resolution. Through the use of artificial intelligence, we can give more potential to such data by using it in a more problem-oriented way. While the handling of such data can lead to solutions, the changes these imply cannot be applied without considering the impact they may have on the city's inhabitants. In addition, the variety of scenarios that urban traffic offers is massive, since the urban domain counts with many different users. From fleets of taxis to privately owned vehicles, all elements must be taken into account when researching urban traffic optimization. Through the use of simulators we manage to reproduce a small part of the real world in a virtual way, which allows us to modify it as we wish. All kinds of changes or improvements can be tested without the need to implement them or the risk of causing negative effects on people's lives. This offers a perfect working area for exploring solutions to the problem of urban traffic, as these are often expensive and costly to implement.

For an accurate simulation, however, the interactions between the elements of the system to replicate must be accurately reproduced. For this, we make use of multi-agent systems. Agents are pieces of software that are inspired by human reasoning: capturing signals from their environment and reacting appropriately, communicating with other agents, making their own decisions, etc. This makes the agents suitable for modeling the different users of a city. All users can be represented by an intelligent agent that adapts its actions and its way of interacting, both with the environment and with other users, accordingly.

For this work we use *SimFleet* [15], an agent-based fleet simulator. This software combines the possibilities offered by simulators with the flexibility of multi-agent systems, offering an ideal framework for the development and testing of solutions for improving urban traffic. However, *SimFleet* has a lot of room for improvement, as it is still in an early stage of development. Motivated by this, we propose different modules that improve *SimFleet* and, from a general perspective, the urban fleet simulation.

This paper approaches the enhancement of urban fleet simulations from two perspectives. On the one hand, the creation of more realistic simulation scenarios by basing element allocation and movement on real-world data. On the other

hand, a more accurate reproduction of human behavior by modeling with rational self-interested agents. These two techniques enrich the simulations set in urban traffic domains and allow them to report more informed and interesting data and measurements.

The structure of the rest of the paper is as follows. Section 2 discusses about the strengths of multi-agent simulation and the use of self-interested agents. Section 3 describes *SimFleet*, the simulator used in our work. Section 4 presents the proposed infrastructure, describing the motivation and each of its modules. Finally, Sect. 5 assesses our work and comments on future extensions.

2 Related Work

In recent times, agent-based simulation has been crucial for creating more practical simulations with high scalability. There are several works that aim to conduct simulations in the urban mobility context to research issues such as traffic, citizen activity, crowds, emergency conditions, or the best placement of diverse facilities. For instance, in [14] authors research the optimal size of a carsharing system so as to maximize client satisfaction, evaluating different configurations through a simulator. Other works such as [7] compare various transportation services employing their own agent-based system.

Different methods have been developed to promote the modelling and creation of these simulations, enabling the implementation of experiments for the study of mobility within and between cities. The work in [2] includes a study of agent-based simulation methods for traffic and shipment.

Focusing on the urban traffic domain, there are many notable simulation tools that have aided in research activities. One of them is SUMO [3], an open-source traffic simulator which can be used to explore route choice, agent communication with different infrastructure, traffic management and even autonomous driving. SUMO uses a an origin/destination matrix to assign movement between zones of the city. Such a movement is described in terms of number of vehicles per time. In this line we also find MATSim [17], a framework for the implementation of demand-modeling and traffic flow simulations. Another example would be SIMmobility [1] which focuses on mobility demand impact prediction for smart shipment services. Finally, there exist also commercial tools such as VISSIM [5] which offers an array of technologies that can be combined to address multiple mobility and transportation problems.

In our work we make use of *SimFleet* [15], which is also an agent-based simulator, focused mainly on the development of strategies for the diverse vehicles of urban fleets. *SimFleet* allows complex simulations over cities with a large number of agents that can interact both among them as well as with certain city infrastructure, such as charging stations. We used *SimFleet* in the proposal of our infrastructure because of its flexibility, which allowed us to communicate it with several external modules. A more detailed description of *SimFleet* is presented in Sect. 3.

Self-interested agent interaction is a deeply researched topic in the field of game theory. From a theoretical perspective, authors in [6] study how cooperation emerges in different situations with self-interested individuals. There are other works with an approach similar to ours, like [11] in which authors consider multi-agent simulation to explore the effects of self-interested drivers on traffic when they act in a completely selfish manner.

With regard to self-interested agent coordination in urban traffic scenarios, techniques such as the ones in [8] and [10] could be applied. Those works present a two-game approach in which agents' possible plans are listed to obtain Nash equilibria that guarantee Pareto optimality and fairness by avoiding conflicts (which assume $-\infty$ utility). Another less computationally complex approach that obtains an equilibrium is the so-called best-response dynamics, presented in [13], and used to inspire the work in [9]. In this work we will present an agent coordination module that deals with the use of many self-interested agents in a single simulation scenario and sorts their actions through the techniques introduced in the two aforementioned papers.

3 SimFleet

SimFleet [15] is an agent-based urban fleet simulator built on the SPADE platform [4]. This fleet simulator was built to allow complex simulations over cities where a large number of agents interact to perform fleet coordination and management algorithms. SimFleet uses the multi-agent systems paradigm to allow the user to get access to autonomous, pro-active, intelligent, communicative entities. SPADE is a multi-agent systems platform that provides these features using a simple but powerful interface, which is why it was chosen for the development of SimFleet.

SPADE allows the creation of autonomous agents that communicate using the open XMPP instant messaging protocol [16]. This protocol is the standard protocol of the IETF and W3C for instant messaging communication and it (or some variant of it) is used in such important communication platforms as WhatsApp, Facebook or Google Talk. SPADE agents have also a web-based interface to create custom app frontends for agents, which is also used by SimFleet to show how every agent is moving through the city in a map that represents all routes made by agents. For the movement of such agents, SimFleet makes use of OSRM², a routing engine for finding shortest paths in road networks. Querying an OSRM server specifying the service *route* and passing origin and destination points returns the shortest route between those two points.

Finally, SimFleet is based on the Strategy design pattern, which allows the user to introduce new behaviors to the SimFleet agents without the need of modifying the simulator. This design pattern is used to introduce new algorithms that follow a common interface. In this case, introducing new coordination algorithms to an agent is as simple as building a *StrategyBehaviour* and loading it at SimFleet startup.

² <http://project-osrm.org/>.

SimFleet also provides some common agent classes that can be used (or inherit from them) to create a simulation. These agents represent the entities involved in a fleet simulator: fleet managers, transports, customers, and service directory. Next, we shortly describe these agent classes and how they interact during the simulation.

Fleet Manager Agents. **SimFleet** simulates an environment where there can be different kinds of fleets that provide services in a city. Each fleet has a fleet manager who takes care of the fleet, allows transports to be accepted in the fleet and puts customers and carriers in touch with each other to provide a service. An example of a fleet manager is a taxi company call center or a goods transport operator with trucks.

Transport Agents. These agents represent the vehicles that are moving through the city providing services. **SimFleet** supports any kind of city transport such as cars, trucks, taxis, electric cars, skateboards or even drones. However, the user can customize the kind of transport for its simulations. Transport may or may not belong to a fleet, but belonging to a fleet brings them some benefits like being found more easily and having a coordinator for its operations. Transport agents receive transport requests from customers and, if free, they will pick the customer up (or the package) and drive to its destination. However, before attending a request, a transport agent will make sure it has enough autonomy to do the whole trip. If not, the agent drops the request and goes to recharge its batteries or refuel to the nearest station. After serving one request, the agent awaits for more requests until the simulation is finished.

Customer Agents. Customers are the entities that want to perform an operation: calling a taxi to move from one place to another, send a package to a destination, etc. This entity is represented by the customer agent. In **SimFleet**, customers do not have the ability to move. They call a transport service which goes to the customer's position, picks up the package (or customer in case of a taxi, a bus, etc.), and transports the goods to a destination. Customer agents depend completely on the transport agents. To get a transport service the customer looks for an appropriate fleet in the directory and contacts to its fleet manager to get a transport service for the customer. The fleet manager broadcasts the requests to some or all of their registered transports (depending on its strategy) and any transport interested in attending it will send a proposal to the customer, who has to accept or refuse it (depending on the customer's strategy too). The customer waits until the transport agent picks it up and, once they arrive at the destination, it stops its execution.

4 Proposed Infrastructure

Our work revolves around **SimFleet** and its potential to aid in the improvement of urban traffic, providing accurate simulations of urban fleets which can be used both for research and testing. However, the current version of **SimFleet** has some limitations which we encountered while working on it. Therefore, we propose

the enhancement of urban fleet simulations in SimFleet by introducing two main improvements: realistic simulation data generation, and self-interested agents for modeling the users of the urban traffic system.

The first improvement refers to the simulation data; the data that is feed to the simulator to characterize the simulation. In the context of urban simulation, examples of such data would be agent type, the amount of agents in a determined date and time, the areas of the city in which agents spawn and move around, the movement they perform, etc. All of these variables can be defined before the simulation execution so as to make it more realistic. Basing the values of these parameters on real-world data ensures that the scenario we are creating is a better representation of the system we want to analyze or experiment on. In addition, the real-world data can be replicated through different techniques, which would ensure the availability of new data to feed the simulator.

The second improvement is the introduction of self-interested agents in the simulations. These agents are selfish and take decisions based on their own private goals. Hence, they offer a better representation of some of the users of the urban traffic system such as drivers of private vehicles. When dealing with self-interested agents, coordination mechanisms involve game theory. The agents will propose their desired actions to every other agent in the simulation. At the same time, each agent will adapt their desired actions to the proposals of every other agent, aiming to avoid conflicts. This process obtains an equilibrium, a solution from which no agent will deviate, as all of them would be doing their best actions with respect to other agents' best actions.

Finally, we propose feeding both the generated data and the agent equilibrium to the simulator. The simulator illustrates the experiment by providing motion to the agents. With it, we can study and collect data about agent interaction, both with other agents as well as the elements of the scenario.

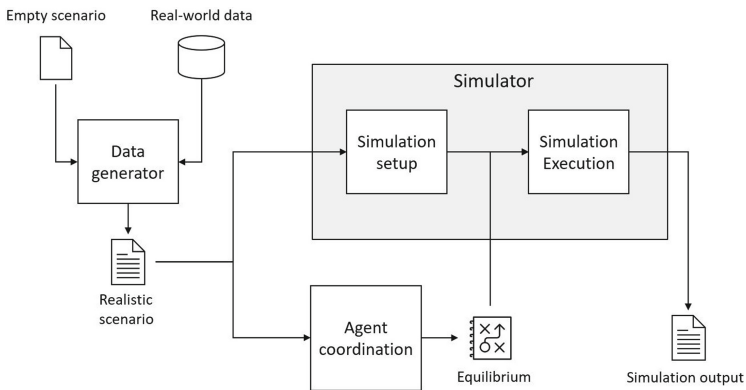


Fig. 1. Architecture of the proposed system

The proposed infrastructure can be seen in Fig. 1. The data generator module would take charge of the simulation data generation. It receives an empty simulation scenario and automatically fills it with the corresponding parameters. In addition, by using databases of real-world data, the obtained simulation scenarios can be more complex and realistic. Such scenario is then passed to the simulator for its setup. If self-interested agents want to be used, it would also be passed to the agent coordination module. This module outputs an equilibrium that coordinates the agents actions. Finally the equilibrium is read by the simulator to replicate the corresponding agent movements. All in all, our infrastructure pretends to ease the creation and execution of more lifelike simulations which, in turn, will report better and more interesting results from the experimentation. Following, we explain in detail each one of the modules of the infrastructure.

4.1 Simulation Data Generators

During our research, we identified the need of a system for the automatic generation of simulation scenarios. This means to provide the simulator users with a tool to test various distributions of elements and agents over the simulation area. In the context of urban simulations, we could use such a system to test different distributions of charging stations on a city. In addition, it could be used to locate vehicles and people with particular goals and create their associated traffic flow. With it, new opportunities arise to simulate different types of configurations, which can be very useful for the research community and even public organisms like city halls that want to test, for instance, the efficacy of charging stations in their towns.

However, one of SimFleet's main disadvantage arises when defining a simulation. A simulation is described in SimFleet by a JSON configuration file, which has to be manually written, including all agents and their attributes. This becomes specially troubling when trying to define big simulations, with a great number of agents.

To solve such issue, we propose two so-called generators [12], programs which, given a series of parameters, fill or create a new simulation scenario, leaving it ready for execution. Besides that, the generators would obtain more realistic simulations and agent distributions, being able to use cadastral, traffic, and Twitter information to obtain more accurate scenarios.

Firstly, we developed a *charging stations generator*, which locates the specified number of charging stations in the urban area where the simulation takes place. With it, various distributions of charging stations can be tested, seeing which ones achieve better traffic flow in the city.

Secondly, we introduced a *load generator* of movements in the city. It locates customer and/or transport agents in the urban area and creates routes for them; i.e.: defines their spawning and destination points. The movement of agents it creates can be random or informed by real-city data (extracted from open data portals) which make the routes of customer agents more realistic by choosing origin and destination points according to factors such as *population density*, *traffic*

information and *Twitter activity*. The influence of each of these parameters is pondered by weights that are assigned to them.

4.2 Self-interested Agents Coordination

Multi-agent systems, as any system that is composed of many units, require coordination in order to correctly function. It is usual that such coordination is performed by an omniscient entity or centralized algorithm which informs each agent about what it has to do, thus removing the agents' own free will. As this approach can be fitting for elements of the urban traffic such as traffic lights or ambulance fleets, it is not realistic for modeling the majority of users. Private vehicle owners, pedestrians and many more are autonomous and take their decisions according to their own interests. Taking this into account, we propose the use of self-interested agents for modeling users of the urban traffic system.

Rational, self-interested agents have their own objectives and make their decisions accordingly to complete them. This "selfish" behavior can be a more realistic approach for representing certain types of users of an urban traffic system, such as taxis. Taxis (or other types of chauffeur-driven rental vehicles) are interested in serving the maximum number of customers possible, as they report a certain benefit. These vehicles usually belong to a fleet which, in turn, may belong to a company. However, generally, taxi drivers will give more importance to their own benefit rather than the overall benefit of their fleet, thus adopting a self-interested behavior. In order to preserve their free will, these agents are not coordinated by any centralized entity but rather by adapting their actions to the ones of every other agent in the system.

A rational entity will always prefer to obtain a reduced benefit than (to obtain) none because of a conflict. Although completely motivated by their private interests, autonomous agents are aware that the interests of another agent may be in conflict with their own. Therefore, the most rational behavior is to modify and adapt the desired actions, so as to still obtain the maximum possible benefit while taking into account the actions of others. This leads to avoid any arising conflict and, eventually, to an equilibrium.

In game theory, an equilibrium is a solution or set of agent actions from which no agent is incentivized to deviate. In other words, every agent has decided on a set of actions which reports them the maximum possible benefit with respect to the actions of every other participant agent. Applying this concept to our urban traffic simulator, with an equilibrium we obtain a coordinated simulation, as no conflict will originate during the agents' execution. As we commented on Sect. 2, the approach of [8] and [10] could be used to obtain a Pareto-optimal equilibrium. However, this approach is computationally expensive since listing the possible strategies of the agents is only possible for a reduced number of agents.

To obtain an equilibrium with lower computational cost for a considerable set of agents, we could approach it by means of best-response dynamics [13]

and inspired by the work in [9]. This would require the definition of the coordination task as a multi-agent planning task. Therefore, the actions that each agent intends to perform during the simulation would be encoded in an agent plan. During the best-response process the agents propose their best plan (the one that reports them more benefits) in turns. After a whole round, the agents reevaluate the plan they proposed taking into account the plan of every other agent. If the actions of another agent are in conflict with theirs, the agent will propose a different plan which (1) avoids any conflict, and (2) is its current best plan. This process repeats iteratively until after a whole round no agent has modified its plan. This means that each agent is proposing their best plan with respect to every other agent's plan. Consequently, as no agent will obtain a benefit from switching plans, an equilibrium has been reached. The agents are coordinated, no conflicts will arise from the execution of their plans and their private interests have been preserved.

As can be seen in Fig. 1, the simulation scenario is inputted to the agent coordination module, as the agent actions are completely related to the elements of the simulation as well as their location, spawning time and other attributes. The module works by loading the simulation scenario, defining the agents, and running the coordination process. Once the equilibrium is obtained, it will be passed to the main simulator, which will make use of it to recreate conflictless agent movement.

4.3 Simulation Execution

The simulation task of our proposal would be performed by *SimFleet*. Nonetheless, any other simulator could be used as long as the integration between the different modules presented in this work is implemented.

SimFleet, as we have commented in Sect. 3, is oriented to the design and testing of agent strategies. As a complete simulator, it is perfectly capable of executing a simulation from the simulation scenario file, coordinating the actions and movements of the different agents. However, with our proposal, the coordination task falls on the agent coordination module and all that is left in the simulator is the visualization of the simulation as well as the capture of system interactions for the subsequent data analysis.

For this to be possible, it is necessary to integrate the different modules with *SimFleet*. The integration with the simulation data generators is trivial, as this module generates realistic simulation scenarios in a format that *SimFleet* is able to receive as input. To connect the agent coordination module, *SimFleet* must be able to receive the file describing the equilibrium that has been reached. By reading such file, *SimFleet* can assign each agent its actions and have them executed. As we have already mentioned, no conflicts should appear during execution since the best-response process avoids them.

Thus, with this integration, we complete the proposed infrastructure. An infrastructure which provides simulations in more realistic scenarios, based on real-world data, and preserves the private interests of the agents during the simulation, representing more accurately the urban traffic.

5 Conclusions

In this work we have proposed a different approach to urban fleet simulation which, integrated with *SimFleet* simulator, enhances its properties.

On the one hand, with the introduction of the simulation data generators, we have provided a tool for *SimFleet* users that allows them to easily define complex simulation configurations, providing a means to create different distributions of agents in the scenario. In addition, the use of the generators can make the simulations more realistic by giving the agents origin and destination points as well as movement based on real-world data. Nevertheless, we would like to improve the generators by studying better ways of distributing the agents and taking into account more parameters to take more advantage of the real-world data.

On the other hand, we have researched the use of rational, self-interested agents ultimately presenting an infrastructure which, together with *SimFleet*, solves urban simulations using transport agents that follow their private objectives. This involved the proposal of an agent coordination module that ensures self-interested agents both avoid conflicts and maximize their utility. For a correct completion of the coordination an equilibrium must be reached, that is, a stable solution from which no agent had incentive to deviate. For that, we proposed the use of the best-response dynamics, following the work in [9]. The equilibrium obtained by this process can easily be integrated with *SimFleet* or other simulators. In this way, we achieved the inclusion of self-interested agents in *SimFleet* simulations which accomplishes the goal of making the simulation more realistic.

In future works we want to create an instantiated version of the infrastructure presented here. Our interest is focused, most of all, on the agent coordination module, as we think self-interested agents are a feature worth exploring when it comes to urban traffic simulations. As we introduced in this work, for the coordination of self-interested agents game theory techniques must be developed and implemented. In addition, we would like to introduce planning so that every agent could decide on which actions are better for itself in every situation.

Acknowledgments. This work was partially supported by MINECO/FEDER RTI2018-095390-B-C31 project of the Spanish government.

References

1. Adnan, M., et al.: *Simmobility: a multi-scale integrated agent-based simulation platform*. In: 95th Annual Meeting of the Transportation Research Board Forthcoming in Transportation Research Record (2016)
2. Bazzan, A.L., Klügl, F.: A review on agent-based technology for traffic and transportation. *Knowl. Eng. Rev.* **29**(3), 375 (2014)
3. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: *Sumo-simulation of urban mobility: an overview*. In: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind (2011)

4. Escrivà, M., Palanca, J., Aranda, G.: A jabber-based multi-agent system platform. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2006, pp. 1282–1284. Association for Computing Machinery, New York (2006). DOI <https://doi.org/10.1145/1160633.1160866>
5. Fellendorf, M.: Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority. In: 64th Institute of Transportation Engineers Annual Meeting, vol. 32, pp. 1–9. Springer, Cham (1994)
6. Hofmann, L.M., Chakraborty, N., Sycara, K.: The evolution of cooperation in self-interested agent societies: a critical study. In: The 10th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 685–692 (2011)
7. Inturri, G., et al.: Taxi vs. demand responsive shared transport systems: an agent-based simulation approach. *Transp. Policy* **103**, 116–126 (2021). <https://doi.org/10.1016/j.tranpol.2021.01.002>
8. Jordán, J., Onaindía, E.: Game-theoretic approach for non-cooperative planning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI), pp. 1357–1363 (2015)
9. Jordán, J., Torreño, A., de Weerd, M., Onaindía, E.: A better-response strategy for self-interested planning agents. *Appl. Intell.* **48**(4), 1020–1040 (2017). <https://doi.org/10.1007/s10489-017-1046-5>
10. Jordán, J., Torreño, A., De Weerd, M., Onaindía, E.: A non-cooperative game-theoretic approach for conflict resolution in multi-agent planning. *Group Decis. Negot.* (2020). <https://doi.org/10.1007/s10726-020-09703-0>
11. Lin, R., Kraus, S., Shavitt, Y.: On the benefits of cheating by self-interested agents in vehicular networks. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, pp. 1–8 (2007)
12. Martí, P., Jordán, J., Palanca, J., Julian, V.: Load generators for automatic simulation of urban fleets. In: De La Prieta, F., et al. (eds.) PAAMS 2020. CCIS, vol. 1233, pp. 394–405. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51999-5_33
13. Monderer, D., Shapley, L.S.: Potential games. *Games Econ Behav.* **14**(1), 124–143 (1996)
14. Monteiro, C.M., et al.: Optimization of carsharing fleet size to maximize the number of clients served. *Comput. Environ. Urban Syst.* **87**, 101,623 (2021). <https://doi.org/10.1016/j.compenvurbsys.2021.101623>
15. Palanca, J., Terrasa, A., Carrascosa, C., Julián, V.: SimFleet: a new transport fleet simulator based on MAS. In: De La Prieta, F., et al. (eds.) PAAMS 2019. CCIS, vol. 1047, pp. 257–264. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24299-2_22
16. Saint-Andre, P.: Extensible messaging and presence protocol (xmpp): Core. RFC 6120, RFC Editor (2011). <http://www.rfc-editor.org/rfc/rfc6120.txt>
17. W Axhausen, K., Horni, A., Nagel, K.: The Multi-Agent Transport Simulation MATSim. Ubiquity Press, London (2016)