A Hybrid Multi-agent Architecture for Modeling in MATSim with an Alternative Scoring Strategy

Youssef Inedjaren[1], Besma Zeddini[1], Mohamed Maachaoui[1] and Jean-Pierre Barbot[2]
[1]*Laboratoire Quartz, COMUE Paris Seine, EISTI, France*
[2]*Laboratoire Quartz, COMUE Paris Seine, ENSEA, France*

Keywords: Autonomous Vehicles, Agent, Multi-agent System, Micro-simulation, MATSim, Scoring, OTFVis (On The Fly Visualizer).

Abstract: The development of new information and communication technologies is contributing to the emergence of a new generation of real-time services in various fields of application. In the area of intelligent transport systems, these new services also include connected vehicles that enable vehicles to collect and disseminate information, safety alerts and make driving smarter and more environmentally friendly. More and more, they concern power-assisted or fully autonomous vehicles. In this paper we propose an architecture of agents and we project it on a multi-agent transport simulator (MATSim). In order to improve the performance of the DriverAgent in the simulation an alternative approach to score the DriverAgent plans is proposed. The results show that the proposed scoring function is able to ensure that agents improve their plans at each iteration performing on the same if not better level than the current scoring function.

1 INTRODUCTION

The simulation of road traffic is booming nowadays. It is increasingly used in the context of traffic management. It is proving to be an effective tool for analyzing a large number of problems that cannot be solved by analytical methods. It is important to note that the first simulation techniques were developed in the early 50's in the transport sector. But nowadays, simulation is considered a necessary step for people working on road development projects or flow control. The inability of road networks to meet the demands of an exponential number of vehicles is the main reason for problems of congestion, pollution, etc. These are a big problem for the road transport system. Yet this remains the most used means of transportation for citizens. The problems of road transport are due to the inadequate demand for the number of vehicles and the saturation capacity of the road, this means the imbalance between the demand and the offer of users. To mitigate these problems, the solution would be to make the road smart, using vehicle-vehicle and vehicle-infrastructure communication. Given these facts, many researchers have embarked on work to find solutions to solve this problems. A system will be called multi-agent if there is more than one agent and these agents interact in the same environment. In general, these agents will be able to cooperate, negotiate or compete, depending on the purpose of the system to be attained and the interests of each agent. There are three types of Multi-agent simulations: The macro-simulation approach involves modelling general aspects of the system such as, density, the average speed of vehicles on the road, etc. A meso-simulation is an intermediate level between the micro and the macro. In the case of road traffic, global traffic can be considered as a collective behavior of the different vehicles. Finally, a micro-simulation approach allows to model each of the vehicles with specific characteristics such as the width of the vehicle, the maximum speed allowed, etc. From these three levels of simulation that we have just presented, we retain the microscopic approach, because it adapts better to the concept of multi-agent. For the simulation, we use the traffic simulator MATSim (Multi-agent Transport Simulation) (Horni et al., 2016) that is a framework for agent-based micro-simulations, and its main role is to optimize the travel demand in large scenarios. The work presented in this paper focuses on proposing a hybrid architecture with different agents with different roles each, then we try to improve the performance of the DriverAgent using a methodology for scoring that we injected in MATSim for this purpose. The paper is organized as follows: Firstly, we pro-

Inedjaren, Y., Zeddini, B., Maachaoui, M. and Barbot, J.
A Hybrid Multi-agent Architecture for Modeling in MATSim with an Alternative Scoring Strategy.
DOI: 10.5220/0007385401860193
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 186-193
ISBN: 978-989-758-350-6

posed a global architecture with three different types of agents (DriverAgent, RSU (Roadside Unit) Agent, and TrafficControlAgent), after that we propose a model of the DriverAgent with custom modules. Secondly, we discuss a comparative study between our approach (Hybrid) and the literature related to agents coordination using centralized and decentralized approaches respectively. In the second part of the paper, we explain the current scoring methodology in MATSim and the proposed scoring function is presented. Finally, a comparison between the results of DriverAgent's plans improvements, using the current and the proposed scoring function is outlined (Results).

2 RELATED WORK

As the document is divided in two important sections, the first part is about the different types of agents (Ferber and Weiss, 1999), but the main idea is to use these agents to propose a simple architecture of agents. The second section is based on the use of the Multi-agent traffic simulator MATSim (Horni et al., 2016), to implement this architecture and to execute the simulation with our experimental needs. We organize our discussion around three topics: Global Architecture, Coordination approach and scoring module in MATSim.

2.1 Global Architecture

In this section we focus our discussion on the global architectures that are proposed to facilitate interactions (vehicle-to-vehicle or vehicle-to-infrastructure), and here we have the notion of cooperative systems. Below some works that proposed different architectures of cooperative systems: In (Wenjie et al., 2005a), the authors proposed a network of wireless sensors of three types of nodes: vehicles, electromagnetic sensors, and intersection controllers. Road sensors continuously broadcast information containing their position: vehicles receiving data from more than three different sensors will then be able to calculate their position by triangularization and send the result and their speed to the controller, who will be able to take decisions on changing traffic lights on intersection (V2I vehicle to infrastructure). Based on the previous work, in (Wenjie et al., 2005b) the authors even proposed the following prototype: The WITS (Wireless Sensor Network for Intelligent Transportation Systems) system is used for information gathering and data transfer. There are 3 types of WITS nodes installed in this system: the vehicle unit on the individual vehicle; the roadside unit along both sides of a road; and the intersection unit on the intersection. The vehicle unit measures vehicle parameters and transfers them to roadside units. The road unit collects vehicle information and transfers it to the intersection unit. The intersection unit receives and analyses the information from other units and forwards it to the policy subsystem (calculates an appropriate scheme based on the predefined optimization target). In (Wiering, 2000) the authors proposed learning methods that allow vehicles to move by minimizing the waiting time at intersections by exchanging information with the traffic lights. Concerning our architecture, we built it on the basis of cooperative systems, drawing on the researches already presented, by implementing roadside units at intersections with a central controller to manage the network.

2.2 Coordination Approaches

Several research works have focused on centralized or decentralized approaches focusing on the coordination of autonomous vehicles. In this section, we consider an approach as centralized if at least one system decision is globally decided for all vehicles by a single central controller. For decentralized approaches, vehicles are treated as autonomous agents that try to maximize their cooperative capacity. In this context, each agent driver receives information from other driver agents and RSU agents to optimize specific performance criteria (efficiency, travel time) while satisfying the physical constraints of the transport system (traffic lights).

2.2.1 Centralized Approach

In terms of security, a centralized architecture is particularly vulnerable. It offers only one gateway, its centralized controller, which is the main weakness of the entire network. It would effectively block this controller to disconnect all users and stop the operation of the entire network.
Reservation Scheme. In this approach, there is an intersection manager (central controller) that coordinates the reservation based on requests and information received from the driver agents in the communication range. The intersection is divided into cells to be assigned for a single vehicle at each moment to avoid collisions. In (Dresner and Stone, 2004) the authors proposed the use of the reservation scheme to control an intersection with vehicles traveling with similar speed on a single direction on each road. Each vehicle is treated as a driver agent which request the reservation of space-time cells to cross the intersection during a particular segment of time defined between estimated arrival hour and the intersection. When the

central reservation system receive the request, it verify if there is no conflict with the already accepted reservations. If the request is rejected, the driver agent must decelerate and send a new reservation request. In the case, each driver agent has the choice to decide its trajectory to fulfill the crossing time interval. To test the proposed system, the authors taken in consideration the delay incurred by the vehicles due to the deceleration required until the reservation request is accepted.

2.2.2 Decentralized Approach

The main challenge faced in the implementation of decentralized approaches is the possibility of having deadlocks in the solutions as a consequence of the use of local information, and also the broadcast series on the network can have the effect of polluting and therefore slowing the data exchange between different entities of the network.

Fuzzy Logic. In (Milanés et al., 2010) The authors Designed a controller based on fuzzy logic, which permit to a fully autonomous vehicle to yield to an entering vehicle in the merging zone, or to cross if it is possible and do not cause a lateral collision. The fuzzy controller controls the accelerator and brake pedals of the autonomous vehicle. This work was extended in (Onieva et al., 2012). The proposed control scheme consists of a fuzzy control system of three layers. The first layer, detects whether a turn or a straight trajectory through the intersection is required. The second layer, precise a feasible speed value to cross the intersection safely, in this layer the fuzzy algorithm is optimized by a genetic algorithm. The third layer, determines the required accelerator and brakes commands to follow the speed reference given by the second layer. Simulation results showed that the system was able to coordinate the vehicles without collisions.

2.3 Scoring in MATSim

In our discussion on simulator features, we focus on the scoring module, that is one of the main elements of the simulation in MATSim, there are many documents already published treating this part of MATSim, such as (Balac et al., 2018) That presented and tested the performance of an alternative activity scoring function in MATSim which can be used to observe induced/suppressed demand effects. The results show that the proposed piece-wise linear function represents the behavior of people on the same if not better level than the logarithmic form. Piece-wise linear functions avoid several limitations of the logarithmic form. They avoid that even small performance duration leads to high utility scores. They better represent the behavior of people in case of shortage or access time by prioritizing different activities which is controlled by the slope of the function. Another enhancement of MATSims utility function is proposed in (Feil et al., 2009). This paper proposed a new MATSim utility function for the performance of activities, based on an asymmetric S-shaped curve with an inflection point as presented by (Joh, 2004) the new function can cope with a flexible number of activities in an activity-travel schedule as it formulates an optimal activity duration by its functional form. In our proposition, the utility function is based only on the time spent while traveling, in order to optimize the maximum possible the agents itinerary, which implies the improvement of the performance achieved by the plans generated by such new utility function.

3 OPERATION OF AN AGENT

There are two types of agents: Reactive agents and deliberative agents (Russell and Norvig, 2016).

3.1 Reactive Agents

This type of agent perceives the state of the environment and decides immediately on the corresponding action such as for example those based on Brookss Subsumption Architecture (Brooks, 2014). This type of agents can be divided into two: Simple reflex agent illustrated in figure 1 and reflex agent with model illustrated in figure 2.

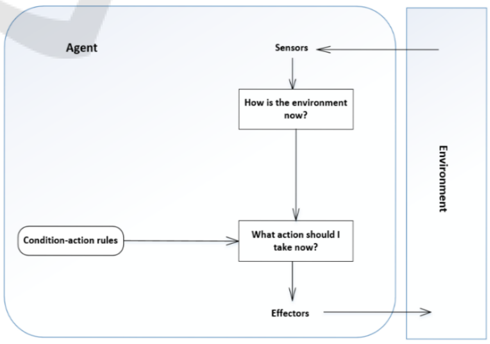

Figure 1: Simple reflex agent.

3.2 Deliberative Agents

In many situations the action to be performed by an artificial agent has to be computed not only on the basis of the state of the environment but also on the basis of its expected effects on it, that is, the agent reasons

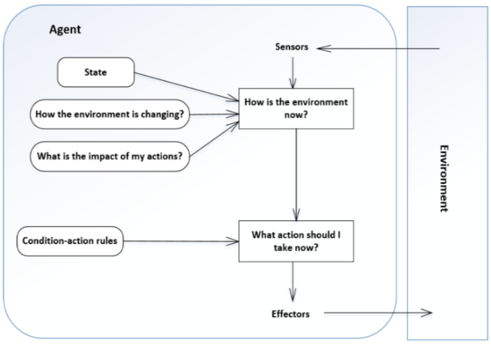

Figure 2: Reflex agent with model.

about its actions. This type of agent needs to have a model of the dynamics of the environment and of the effects of its actions on it. Deliberative agents may appear less efficient than reactive agents because they have to reason about the action to perform but they are far more autonomous and flexible than reactive agents (Balmer, 2007). In literature there are three types of deliberative agents: Agent with goals illustrated in figure 3, agent with utilities illustrated in figure 4, and the learner agent illustrated in figure 5.

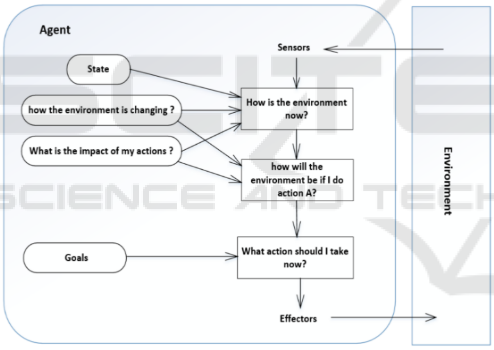

Figure 3: Agent with goals.

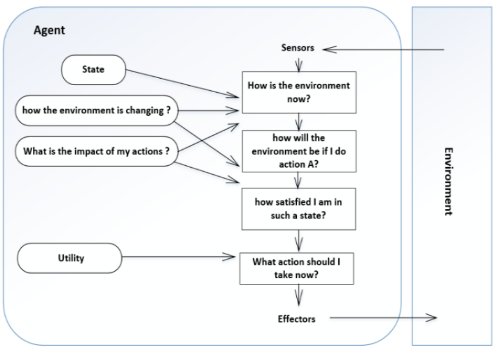

Figure 4: Agent with utilities.

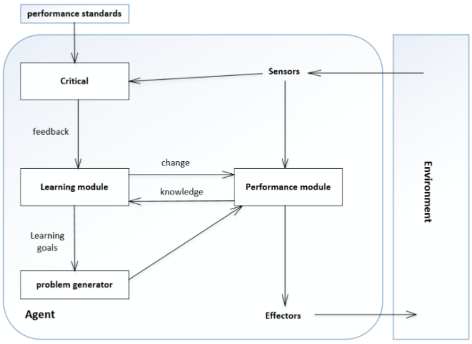

Figure 5: Learner agent.

4 DRIVERAGENT MODELLING

Our model is based on a deliberative behavior. We use a deliberative behavior, as a type of behavior that is linked to a cycle, that is to say, when an agent receives information from the environment via its sensor, it does not directly trigger an action, this information goes through a set of states, before the agent makes the decision on the consequences of his actions on the environment. The figure 6 illustrates this approach.

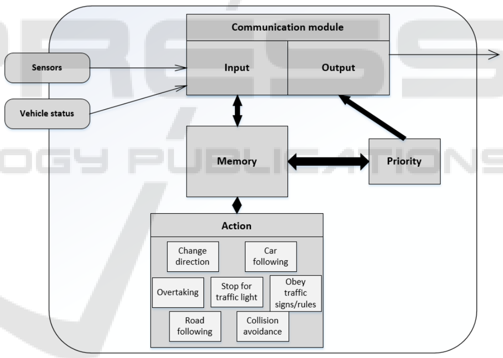

Figure 6: DriverAgent Model.

DriverAgents communicate through messages exchanged via the communication module. Each of the agents has a sensor that not only provides information on its state, but also on the state of the environment. This information is sent to the communication layer that processes it and then routes the results to the agent's memory. The module Priority chooses the highest priority action based on the prediction of the position of other vehicles on the environment, and sends it to the communication module.

5 PRESENTATION OF THE ARCHITECTURE OF OUR MODEL

The basic organizational structure of our system contain three types of agents having each one characteristics that distinguish it from others: driver agent, RSU agent, traffic control agent (see figure 7).

5.1 Driver Agent

This only dynamic component drives the road network. Each vehicle is equipped with sensor information allowing the agent to perceive its environment. Driver agents collaborate to monitor the road network. Indeed, they produce information contributing to knowledge in real time traffic conditions or predict traffic conditions. All events observed by the vehicle during its movement must be transmitted to other vehicles, which transmit this message to the nearest RSU agent. In turn, the RSU agent may send a message to a stub of vehicles. Generally the exchanged data are intended to produce alerts and inform drivers of an event occurring.

5.2 RSU Agent

This static component represents a specific point of the transport network: at each intersection, we introduce an RSU agent. It produces, in real time, information about the current state of traffic within the road segments that it manages. The RSU agent must handle messages emitted by vehicles in the first place to inform the traffic control agent of disruptions to normal traffic situation. Second, he warns drivers on their way to the place of accident for example. The RSU agent can exploit the average speeds, the time inter-vehicles to take information on the status of traffic: fluid, dense or blocked.

5.3 Traffic Control Agent

Communication vehicle-to-vehicle and communication road side units-to-vehicle used to exchange a couple of accurate information on traffic conditions. Following these exchanges, the traffic control agent will have a global vision on the road network of the area it oversees. The main goal of this supervision is to maximize road safety and minimize the time spent on the roads. The traffic control agent collects the information on road conditions from RSU agents then synthesizes the data. Once the traffic control agent has a more complete view of the road network, it broadcasts information to RSU agents.

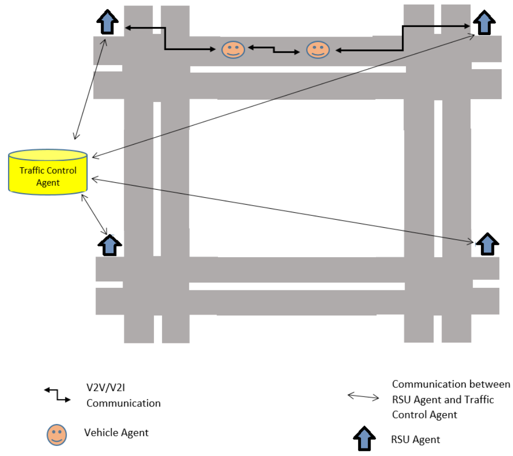

Figure 7: Global Architecture.

6 HYBRID COORDINATION APPROACH

We have seen that the centralized architecture poses problems of security, robustness, and limitation of the bandwidth. The problems come directly from the use of a central controller. So to remove the central controller it is necessary to distribute the treatment that must be done at the level of each vehicle, then to make them communicate. It is on these mechanisms that decentralized Peer to Peer networks are based. So there are no more central controllers, it is all the elements of the network that will play this role (Driver Agents, RSU Agents). Each vehicle in its roles is identical to another, which is why these types of pure Peer to Peer networks are called. Nevertheless, it explodes communication costs: the number of messages exchanged between the vehicle agents is of quadratic complexity, as well as the broadcast series which are broadcast on the network and which have the effect of polluting and therefore slowing the exchanges of data (V2V or V2I). This is why for our simulation well use our proposition, the hybrid approach, where all agents containing in the Multi-agent system participate in the coordination of traffic at intersection.

The hybrid approach (see Figure 8) is a compromise between the centralized and decentralized approach. The RSU agent is inserted between the driver agent and the traffic control agent and has the role of producing, in real time, information about the current state of traffic within the road segments that it manages. And must handle messages in the first place to inform the traffic control of disruptions to normal traffic situation. Even if communication vehicle-to-vehicle communication is used to perceive the environment to produce information contributing to knowledge in real time traffic conditions or predict traffic conditions, the roadside units-to-vehicle communication is also used to exchange a couple of accurate information on traffic conditions. Following these exchanges, the traffic control agent will have a global vision on the road network of the area it oversees. The main goal of this supervision is to maximize road safety and minimize the time spent on the roads. The traffic control agent collects the information on the RSU agents then synthesizes the data. Once the traffic control agent has a more complete view of the road network, it broadcasts information to RSU agents.

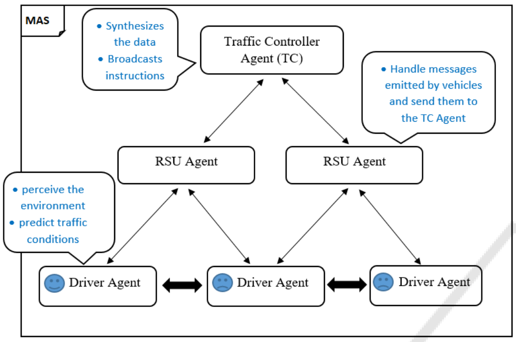

Figure 8: Hybrid Approach.

7 MATSim

MATSim (Multi-Agent Transport Simulation) (Balmer et al., 2009) is a framework for agent-based microsimulations (traffic simulator). It consists of several modules that can be used independently. It is developed by the teams at ETH Zrich(Swiss Federal Institute of Technology in Zrich) and TU Berlin(Technical University). (Balmer, 2007) Provides a detailed description of the frame. Because of its micro agent-based approach, each individual in the system is modeled as an individual agent, and each of these agents has custom parameters such as available modes of transportation and scheduled daily activities. As the simulation framework has a modular structure, the agent's parameters can be easily extended with new parameters. MATSim tries to optimize the travel demand in large scenarios. The optimization of the travel demand is done following an evolutionary algorithm, by going through different modules (execution, scoring and replanning) iteratively.

In MATSim, each agent has his own plan, which contains both the activities planned for the agent and the modes of transportation (legs) linking these activities. The Modes of transport and activities may contain several attributes, describing the route from one activity to another, such as departure time, estimated time of arrival, links connecting activities, etc. thus the plans are introduced so that they are optimized by the iterative process described above. To do this, the system iterates between the generation of the plane (the mental layer) and the simulation of the traffic flow (the physical layer). MATSim remembers several plans per agent and scores each plan based on its performance measured by a scoring function. The re-planning mechanism is continued until the plans reach an approximate balance. A formal definition of this mechanism can be found in (Nagel and Flötteröd, 2012).

8 EXTENDING MATSim

The contribution in the second section of this paper is to provide MATSim with a new scoring function so that the execution of the simulation will be controlled by the parameters that we need to implement. In this section we describe the modifications that have been introduced in the simulator.

8.1 Scoring

As a result of the traffic flow simulation, events (MATSim events) are produced to calculate the effective utility of each daily plan, taking into account the effects of interaction between agents. A good daily plan is specified by a utility function. MATSim currently uses an effective utility function described in (Charypar and Nagel, 2005). Without going into details, the elements of the utility function are:

- A positive contribution for the (usually) positive utility earned by performing an activity.
- A negative contribution (penalty) for traveling.
- A negative contribution for being late.

The utility function induces the behavior of the agent, because the agent searches in the solution space of the utility function for the best possible score, which implies the best possible daily plan.

In general, the scoring function calculates the score for one plan of an agent. As long as we work on autonomous vehicles, our main goal is to ensure users safety and to save his money, thats why we must avoid possible congestions, and subsequently the loss of time and money. Thats why we proposed a custom scoring function (1) by adding and deleting some factors to meet our experimental need:

$$U_{plan} = \sum_{i=1}^{n} (u_{travel,i}) \quad (1)$$

where n is the number of activities an agent has in his daily plan. In general, traveling decreases the score (negative utility).

$$U_{travel,i} = \alpha_{mode} + \beta_{TT,mode}.TT + \beta_{cost,mode}.Dist \quad (2)$$

$$+ \beta_{CongT}.CongT \quad (3)$$

where α_{mode} is a constant depends on the mode used (car, bike, bus...), $\beta_{TT,mode}$ is the marginal utility of traveling, $\beta_{cost,mode}$ is the marginal utility of cost for the specific mode, β_{CongT} is the marginal utility of the congestion.

The novelty is that our new scoring function depends only on the utility of traveling (1), so we overlooked the utility of exercising an activity. Then, we added a new factor inside the utility of traveling formula (2), a factor that reacts on driver agents who enter a link and find it with a level of congestion bigger than a fixed threshold (3). The Algorithm 1 illustrates the congestion engine injected in MATSim.

```
Begin
Static class CongestionEngine{
  private EventsManager eventsManager;
  private Map<Id<Vehicle>,
  Id<Person>> vehicle2driver=new HashMap<>();
  [...]
  Public void handleEvent(LinkEnterEvent event){
    If (congestion()){
     EventsManager.processEvent(new
     CongestionEvent(event.getTime(),
     vehicle2driver.get(event.getVehicleId())));
    } }
  Private Boolean congestion(){
    If (Road.carsOnTheRoad.size() >=
                Road.maxNumberOfCarsOnRoad) {
    Return true;
   } else {
    Return false;
   } } }
End
```

8.2 Case Study

8.2.1 Berlin Example

We executed the simulation using MATSim, and we applied it on the example of Berlin (Planning and group of Technische Universitt Berlin, 2018). For each simulation on MATSim you need at least three types of files: configuration file, network file and population file. So to run the simulation we will loaded the configuration file directly, the latter which will contain the path to the network file and the population file. Regarding the population file, the number of agents in this example is 15931. We executed the simulation twice: The first results were with the current scoring function of MATSim, while the second results were with the new scoring function.

8.2.2 OTFvis Visualizer

The short term for On the Fly Visualizer, OTFVis was designed to support actual visualization of live simulation runs with MATSim. Therefore, the purpose of the OTFVis is the debugging of MATSim (input) data. The OTFVis is written in Java and available as source code to extend for different MATSim projects special needs (Horni et al., 2016). We will use the OTFVis to view the traffic on the Berlin network. The OTFVis will display OSM-maps(Open Street Maps) as background, by setting option mapOverlayMode in OTFVisConfigGroup class to true, and the coordinate system of our scenario (Berlin scenario) in the configuration file (global section). in the figure 9, we illustrated the Berlin network visualization on OTFVis, before and during the execution of the simulation.

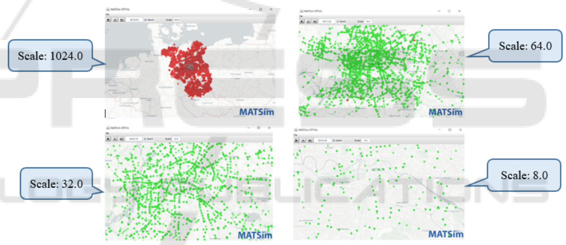

Figure 9: Visualization on OTFVis of the Berlin scenario.

8.2.3 Results

At the end of all iterations we analyzed the output plans file, we extracted the results and we represented them on a histogram (see figure 10) with agents on the abscissa and the number of replannings done for each agent as ordinates. So, after the analysis we found out that more replannings are done with the new scoring function, which means that the plans generated by our utility function are more improved in terms of the time spent while traveling. Moreover, the more replannings are done at each execution, the more optimization and a better itinerary is calculated in order to gain a better score, which impacts positively the performance of a given plan. So, the new scoring methodology that we proposed is more effective, especially with congestion scenarios.

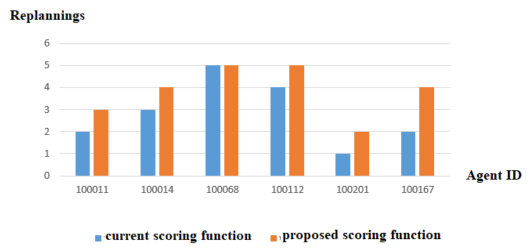

Figure 10: The effect of the new scoring function on agent's plans improvement.

9 CONCLUSION

The main goal of this paper was to propose an architecture of agents (Driver agent, RSU agent, Traffic control agent) to model the transportation system, then project it on a traffic simulator to run the simulation where we can observe and optimize the routes and the time spent travelling by the DriverAgent. For that we used a special traffic simulator that called MATSim. Our addition to the simulator is to propose an alternative scoring function which can be used to observe induced congestion effects. The results show that the proposed function improves agents plans better than the current scoring function. Future works will be headed to propose a platform that represents communications between a wide range of autonomous transport systems, and to deploy a large number of scenarios highlighting the vulnerabilities of autonomous transport systems, particularly in a context with a large number of interactions between vehicles in real traffic situation.

REFERENCES

Balac, M., Janzen, M., and Axhausen, K. W. (2018). Alternative approach to scoring in matsim and how it affects activity rescheduling. In *Proceedings of the 97th Annual Meeting of the Transportation Research Board (TRB 2018)*. The National Academies of Sciences, Engineering, and Medicine.

Balmer, M. (2007). *Travel demand modeling for multi-agent transport simulations: Algorithms and systems*. PhD thesis, ETH Zurich.

Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., and Nagel, K. (2009). Matsim-t: Architecture and simulation times. In *Multi-agent systems for traffic and transportation engineering*, pages 57–78. IGI Global.

Brooks, R. A. (2014). How to build complete creatures rather than isolated cognitive simulators. In *Architectures for intelligence*, pages 239–254. Psychology Press.

Charypar, D. and Nagel, K. (2005). Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32(4):369–397.

Dresner, K. and Stone, P. (2004). Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 530–537. IEEE Computer Society.

Feil, M., Balmer, M., and Axhausen, K. W. (2009). Enhancement and empirical estimation of matsims utility function. In *9th STRC Swiss Transport Research Conference: Proceedings*. Swiss Transport Research Conference.

Ferber, J. and Weiss, G. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading.

Horni, A., Nagel, K., and Axhausen, K., editors (2016). *Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London.

Joh, C.-H. (2004). *Measuring and predicting adaptation in multidimensional activity-travel patterns*. PhD thesis, Technische Universiteit Eindhoven.

Milanés, V., Pérez, J., Onieva, E., and González, C. (2010). Controller for urban intersections based on wireless communications and fuzzy logic. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):243–248.

Nagel, K. and Flötteröd, G. (2012). Agent-based traffic assignment: Going from trips to behavioural travelers. In *Travel Behaviour Research in an Evolving World–Selected papers from the 12th international conference on travel behaviour research*, pages 261–294. International Association for Travel Behaviour Research.

Onieva, E., Milanés, V., Villagra, J., Pérez, J., and Godoy, J. (2012). Genetic optimization of a vehicle fuzzy decision system for intersections. *Expert Systems with Applications*, 39(18):13148–13157.

Planning, T. S. and group of Technische Universitt Berlin, T. T. (2018). The matsim open berlin scenario. https://github.com/matsim-vsp/matsim-berlin.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Wenjie, C., Lifeng, C., Zhanglong, C., and Shiliang, T. (2005a). A realtime dynamic traffic control system based on wireless sensor network. In *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*, pages 258–264. IEEE.

Wenjie, C., Liqiang, G., Zhilei, C., Zhanglong, C., and Shiliang, T. (2005b). An intelligent guiding and controlling system for transportation network based on wireless sensor network technology. In *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*, pages 810–814. IEEE.

Wiering, M. (2000). Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pages 1151–1158.