Simulation of a carpool system in an artificial road traffic network using NetLogo

António Pedro Fraga, Luís Oliveira and Pedro Martins[1]

***Abstract*— Carpooling (or car-sharing) is the sharing of car journeys so that more than one person travels in cars. This type of transportation method has several advantages, being however hard to implement such a system in a real life situation because of the lack of research involving this area. The project that we describe in this paper has the purpose of addressing the problem about the integration of a carpool system in an artificial road traffic network using NetLogo. By simulating all the needed variables in order to achieve said system, and by plotting and documenting the gathered results, we found several conclusions which we deemed interesting, such as how more densely populated areas have a higher relationship between the number of people wanting a carpool and the number of people actually offering a carpool, or how accidents heavily influence the road traffic network.**

I. INTRODUCTION

A. Context of application

Carpooling can be defined as a way of traveling with more than one person with similar needs. Its main advantages are cost reduction among its participants as well as a more rewarding social experience with less lonely travelers. Its also considered eco-friendly because of the carbon emissions and traffic congestion reductions associated with it, being associated with a reduction in the number of traffic collisions as well.

Our aim was to create a simulation of a carpool system in an artificial road traffic network following an agent based model using NetLogo, which is a a multi-agent programming language and modeling environment for simulating natural and social phenomena. The provided inputs must be changeable by the user and also valuable to the problem itself. The main goal is to observe, in an artificial society, which carpool model better serves those who want to use such a system, and in which ways that model can be further improved.

Solving this problem required a great deal of effort of all the people involved that were directly involved in the development of this project. Simulation techniques and other related technical terms were learned throughout the whole development phase, and the users feel like this study can be further expanded in the future with other inputs from different sources.

*This work was not supported by any organization

[1]António Fraga, Luís Oliveira e Pedro Martins are students within the Faculty of Engineering of University of Porto, Master in Informatics and Computing Engineering, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

This paper is organized as follows (each item being a different section):

- Introduction themes with approach to the motivations behind this project.
- Related, already developed work by other sources.
- All the methodological approaches to the problem.
- The main components and the implementation of the system.
- Results insights and analysis, with a proper discussion.
- Conclusions and future work.

There is also an appendix which addresses developed work work which is unrelated to the other sections of the paper, as well as areas for acknowledgments and references to used sources.

B. Problem definition

This project intends to answer whether or not, and in which conditions, does a carpooling system works among a society which uses the same roads to travel on their everyday lives. The problem we want to answer might be defined as the following question:

Under which conditions is a society capable of efficiently serve its citizens with a carpooling system?

In the context of our problem, an efficient result is considered every time the number of users served by the carpool system are greater than or equal to the number of users not served. For example, considering the total of people who want to use the carpool system (whether as drivers or users alone) as 400 people in total, such system is considered competent enough if at least 200 people are served by it (that is, higher than 50%).

The other issue besides the actual problem is which parameters, constraints and/or variables we must define in our system in order to mirror an existing society. In later sections of this paper, we describe every choice, assumption and absence of our model, and how they are related with each other in the context of the entire system.

We envisioned the whole simulation to always be as real as possible, and one of our primary goals was to create a world which mirrored ours as much as possible. As such, this paper also documents our tries to encounter resemblances to the world we live in, and how we prioritized some similarities and discarded others we figured wouldn't add much to the final result.

C. Motivation to solve the problem

As millennials that we, the three creators of this paper, are we like to keep up with everything that means innovation and improve the world through technology. As such, this theme called to us from the start. It seemed to us like a perfect opportunity to further develop our knowledge in the simulation field while tackling a very relevant theme nowadays. A lot of our fellow students use the carpooling system to get to the faculty and sometimes encounter problems in how to make the most efficient ride. Since this is a problem that affects many of us students, we figured that a contribution to this subject would be very positive.

Another motive behind our decision to tackle this issue was the research we made regarding carpooling and how smartphones and applications are involved in implementing this type of system in societies. We found out that there's many recent, pioneer investigations and research going on which studies the influence of environmental variables in carpooling society integration. This work is reviewed in a later section of this paper.

D. Research questions

Some of hypothesis we want to clear out with our problem can be defined as simple sentences, such as the following:

Are areas with higher concentration of people more suitable for carpooling?

How does automobile accidents affect the roads and the traffic flow?

Does a carpooling system heavily reduces the CO2 emissions to the atmosphere?

Are automated vehicles viable in our city?

Can a carpooling way of thinking be contagious to other people?

Although interesting at first, our model behaves in such a way that, unfortunately, we are not able to clearly answer all these questions. The main reason behind these limitations were time restrictions regarding the completeness of the model itself. However, with the amount of information we acquired, we have all the available tools to formulate more powerful, insightful questions about how a carpool system can positively influence society. This information will be presented in later sections of this paper.

E. Expected contributions

The overall results of this project might be useful in several areas, and potentially reusable by other future researchers looking forward to extend our work. Among the possibilities, we expect that our work will eventually be picked by:

- Web-development companies which might want to create projects related with roads, cars or driving in general.
- GPS companies which might want to cross results between the passengers interested in participating in carpooling initiatives and the information about the network.
- Academical purposes, for people who also want to learn more about simulation techniques as well as NetLogo or other multi-agent technologies.
- Environmental companies, which might look under which conditions does carpooling greatly decreases atmosphere pollution. ...

II. Literature review

A. Background

This type of project has been discussed before in areas such as environment and GPS developers in order to include the best possible route when carpoolers are available. Nowadays, carpooling systems are not so well distributed (although there's several startups on the matter) because the studies have recently been done in this area which allow more people to easily develop solutions to it.

Traditional carpooling consists of a person wanting to go to a place and displaying its intention to do so by going to a road and wait. This usually implies the person somehow knows the selected road is well known to be a common path to the destination he or she is trying to reach. Without other type of information, however, such as availability of drivers willing to offer their car for carpooling, the person may wait for a very long time, eventually giving up its intention for another way of transportation.

With the introduction of smartphones, GPS and general purpose applications, it's possible to greatly improve the performance of a carpooling system, solving the aforementioned problems regarding unavailability of drivers, displaying distributed information about the location of both the driver and the user requiring carpooling, and maybe even include networking features available in other service transportation applications such as driver evaluation.

B. Related work

There are several papers which studied many aspects of this area.

A team of five people from the University of Vigo, working as Ph.D students in the Department of Telematic Engineering, wanted to work on a project that aimed at increasing the performance and reliability of the wireless transport and to provide general traffic safety improvements to all its users. They developed what they called the WiSafeCar (Wireless Traffic Safety Network between Cars), and made efforts on several different aspects on a NetLogo environment, designing an interface and studying very similar characteristics to the ones we propose in our work. [2]

Their work was further extended with more information regarding how a carpooling system might improve citizens mobility, maintaining their developed simulation in NetLogo but acquiring more social aspects about carpooling, essentially improving the whole concept they were trying to create. [3]

Another project that demonstrates how there is a growing interest in this type of work is one developed in the University of Vigo as well, but this time by a different team. This project presented SinCity, a pedagogical testbed to compare multi-agent learning strategies. The work was developed in NetLogo as well and served the community by explaining the capabilities of the language. This happened because, by creating a SinCity simulation that worked as an extension of the predator-prey pursuit problem, and by showing how new and powerful algorithms and learning techniques are useful, the user could understand how a multi-agent language can be used to solve pretty much any existing social issue. [4]

NetLogo is also widely used for other purposes, in a multitude of natural or social phenomena, with a wide collection of extensions and libraries. By learning how to cleverly use the language in every situation in which it might prove useful was the subject of a paper written in the Center for Connected Learning and Computer-Based Modeling at the Northwestern University of Evanston, in Illinois. [5]

III. Methodological approach

A. Model classification

By analyzing an artificial society resembling ours, we had to think of it in terms of a functional traffic network system and how its variables behave between themselves. Therefore, our model can be considered as:

- Dynamic - because it evolves over time.
- Stochastic - as it considers random variables (such as accident probability and car speed).
- Continuous in space - because the changes are so smoothly integrated in the model that one can't perceive the difference in the system when the agents move themselves.
- Discrete in time - functioning with a concept called ticks per second, clearly being visible any changes to the system.

In the remaining sections of this paper, we will refer to the people who work behind the simulation as observers instead of users.

B. Simulation environment

Because we wanted to simulate a carpool suited environment, we selected our simulation site to be an area which could easily be populated with cars and people. We wanted to insert real life situations, in which the roads are easily distinguishable from the other pavements, the traffic flows in two directions along two lanes, traffic lights control intersections and people have limits when it comes to the time waiting for a response regarding their carpooling intentions. Because the system is dynamic, time must also be modeled.

Another aspect is how the drivers behave consciously and are not just randomly, meaningless generated. This means that in a simulated world, every driver has an origin (the house), a destination (the work) and the selected path (the road) in which they systematically traverse in order to meet their deeds. We will discuss in later sections more about our implementation and how we reused a number of existing models and libraries to create our own personal model.

C. Input variables

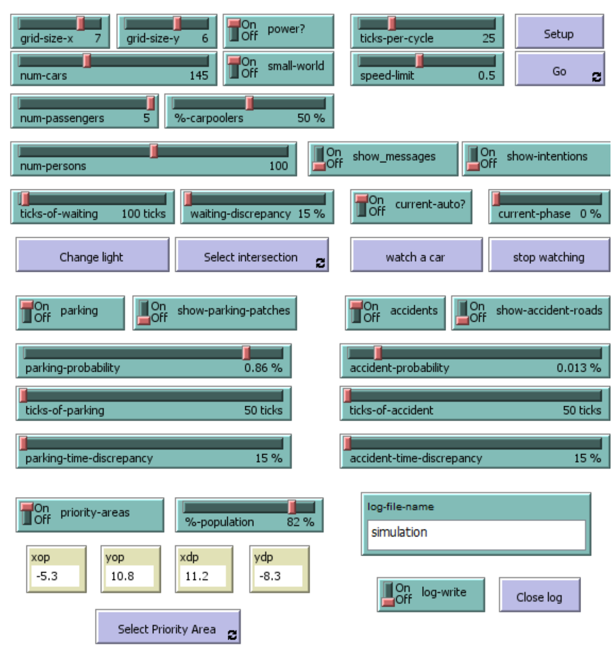

Fig. 1. Variables selected to control the simulation through the user input.

As already stated, our simulation has the purpose of showing how the environment related to a road can influence the traffic network that rides on it, such as the number of cars, the population density, the accident probability and the lights on the intersections. All these will eventually influence the outputs that are used to prove our various research hypothesis. The results will be shown in a later section of this paper. Among the existing input variables, we can state the following points:

- The grid size inputs will define the size of the world in which we will model. As will be seen shortly, the model behaves in a Manhattan-like traffic system, in which the roads are connected in a matrix system of intersections controlled by traffic lights. The values can vary between 1 and 9.
- The cars have a speed limit that emulate how they move in the roads.
- There's variables that change the density of the population in the world. This population (and therefore, the cars they own) will equally influence the demand and the offer for carpoolers (this is, as more people exist, the higher the percentage of people offering their car for carpooling as well as people wanting to use a carpool service).
- Cars are subject to a small, albeit significant and recurrent probability of accidents, and those accidents heavily influence the traffic in the road they occurred.
- People might decide, with a lower than average probability, to park the cars instead of having them continuously doing the same path over and over. This results in less cars in the system, influencing the flow of carpooling and resulting in less people being served by the carpool system.
- Other variables influence how the traffic lights behave, the existence of priority areas and the overall speed of the simulation (by changing how the ticks, whose definition we will present in a later section, behave along the system).

Most variables are discrete and changeable, resulting in a very dynamic system where the observer has total control over what's happening.

D. Output variables

After inserting the variables, the simulation shall run. When the observer decides to stop, the simulation ends. Afterwards, the observer can access a generated log where it can perceive the most relevant information about the simulation. This comes in a .log file and includes the time passed between the beginning and the end of the simulation, the progressive information about how the cars behaved between themselves and the citizens and other information deemed relevant to the problem.

E. Performance measures

The observer has also access to a myriad of plots and numbered information where it can evaluate various aspects about the simulation as time passes. These slots are constantly updating themselves, ending when the simulation also ends.

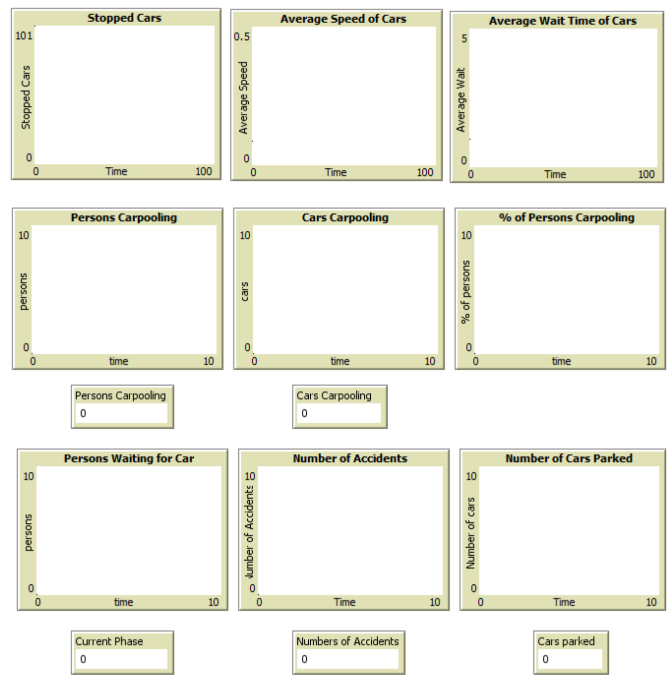

Fig. 2. Example of plots accessible to the user

Among these, we can check:

- The number of stopped cars.
- The average speed of cars.
- The average waiting time of cars when they are stopped.
- The number of persons carpooling, the number of cars offering their services, as well as the percentage of people offering to carpool (in relation with the total amount of people).
- The number of accidents in relation with time and the percentage of cars that are influenced by the accident (in relation with the total amount of cars).
- The number of parked cars as well as the percentage of cars which are parked (in relation with the total amount of cars).

F. States of the system

The system can be in one of two states at each time. If it is stopped, this means the model is being prepared with the various variables by the observer, and the outputs (both the .log file as well as the plots) are empty. After clicking in the appropriate button, the observer can watch while the simulated world, with the inputs chosen by him, are properly functioning. Meanwhile, the outputs are all being generated as time passes, and the user can freely choose when to stop the simulation and reorganize the input variables.

IV. Implementation and results

A. Multi-agent systems

A multi-agent system is a system which comprises many existing agents at once, each of them interacting between each other following a certain asset of rules. An agent can be considered as a software-based computer system with its properties being well defined and known to the observer. By approaching this type of methodology, we were able to implement our system and simulate through it a road traffic network, by considering each element in the system as an agent (the cars, the people, the traffic lights and every patch representing either a road or a building) and having them behavior being constantly influenced by the remaining agents in the system. We accomplish so by using the functionalities and capabilities of a multi-agent programmable modeling environment called NetLogo.

B. NetLogo

We used NetLogo as a multi-agent modeling environment to create and simulate our carpooling city model. NetLogo is well known to create the desired conditions to simulate social and natural phenomena where the people behind the model being created are free to set their rules and instructions that all the agents must follow.

NetLogo is a particularly easy to use environment where one can effortlessly define the interface to be used by the observer. The programmer has fast access to the code being written, having the generated models for rapid access to be tested so that existing questions can be answered.

There's lots of libraries available and an extensive collection of existing models which can be further extended. Documentation is also vast and tutorials on point is one of its main advantages. There's also many extensions, two of them, the BDI Agents extension and the FIPA ACL extension, being integrated in our project. This integration will be explained in a later subsection of this paper.

NetLogo can be considered as a world made of agents that behave according to the rules defined by the modeler. These agents can be either patches (agents which are immovable but interact with the other agents) or turtles (agents which are mobile, move on top of the patches, and can also interact with the other agents). There is also a third agent which is capable of seeing everything that is currently going on in the system, the observer, also having access to the variables being considered by the model.

C. Traffic Grid as a previously created model

We worked upon an existing model offered by the model library existing in the NetLogo software. This model, developed by Gershenson in 2004, behaves according to the following rules:

- The cars have the goal of going from their current generated position to its house.
- The cars flow in a straight line with only one direction.
- The lights change themselves automatically and control the intersections which divide the city.
- The cars drive at a maximum speed (defined in an input variables) and stop when they find a car in front of them or when they reach an intersection which has a red light on it.
- Each agent road patch can be empty or have a car on it.
- There's outputs related with the different scenarios, which will analyze the existing variables such as the speed or the car waiting time.

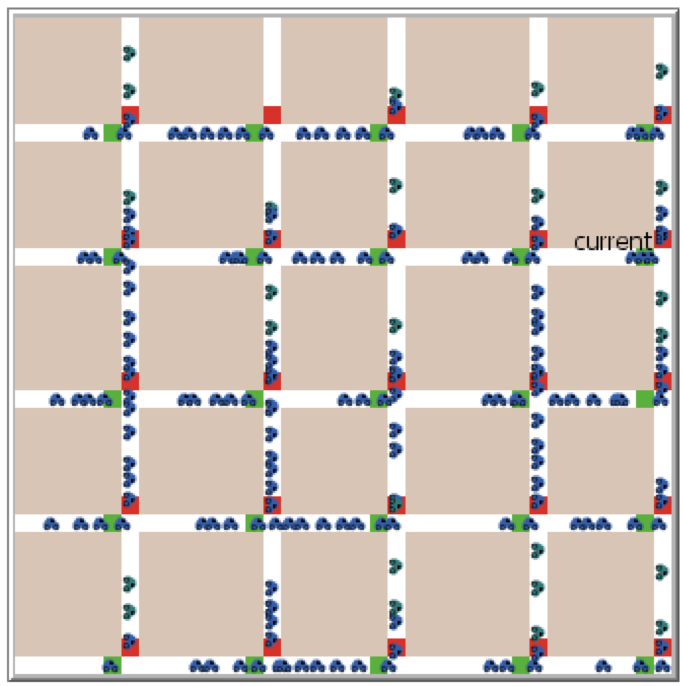

Fig. 3. The basic, unchanged traffic grid which we used as a starting point.

D. Implemented improvements

We decided to further update the existing traffic grid model because we considered that it didn't correspond to a realistic and updated vision of a city capable of responding to the carpool problem we previously stated. We envisioned a larger city, with two lanes instead of one, where the people take a higher stake in the decisions of the model and where traffic accidents and parking cars can also be considered for simulation. Along these, we wanted to make full use of a BDI architecture, making use of an already existing extension to further expand our model. The main choices we implemented were:

- Two lanes in one road, instead of only one (creating the possibility of bidirectional traffic).
- Traffic lights at each intersection.
- Cars moving in different directions (and not only straightforward).
- Population density, being defined by the observer by dragging in the interface the areas in which the higher percentage of the population will be.
- Traffic accidents, where the observer can define the probability that, as each tick passes by and the system state changes, a moving car turtle will have an accident of some sort and heavily condition the traffic behind him.
- Parking, where the observer can choose a parking probability in the same way as an accident probability. This sort of input will generate a random behavior, which is known to cause impact in the society (and therefore, in the carpooling service).
- People wanting to carpool, being themselves a very different element of the system besides cars, but being in constant interaction with them. These people are randomly generated by the observer, which not only control their total number but how much time must pass for them to give up their desires for carpooling as their method of transportation. People is crucial in determining how a society with those parameters are prepared to suit a carpooling environment (as will be seen in the results discussion). This happens because they will decide whether or not they are happy with the system (happiness means they found someone which offers to give them a carpool).
- If people get answered by their carpooling calls, they get a happy yellow smiley face. If however they don't get a positive answer after a certain amount of time, they get a sad blue face. Regardless of the outcome, they disappear and give space for new generated people.
- Cars, as well as people, can not be generated in the same patch.
- The shortest path algorithm is used to inform the agents about the best way to travel from their origin to their destination.

E. Integration of outside extensions

As previously mentioned, we used two extensions existing in the NetLogo library to include a BDI architecture in our system, integrated with a FIPA extension as well.

A BDI architecture, short for Beliefs-Desires-Intentions architecture, is a behavioral agent architecture which, as the name suggests, has the purpose of developing intelligent agents. Characterized by the implementation of an agent's beliefs, desires and intentions, it is known to actually use these concepts to solve various problems in agent programming. The extension works by integrating beliefs in the agents (the actual number of passengers in the car, the maximum number of passengers in the car and the location of the work location, to name a few), desires (to not have an accident, for example) and intentions (go from home to work, from work to home or park a car). There's also an integrated controller for the observer which allows him to watch each agent's intentions in real time.

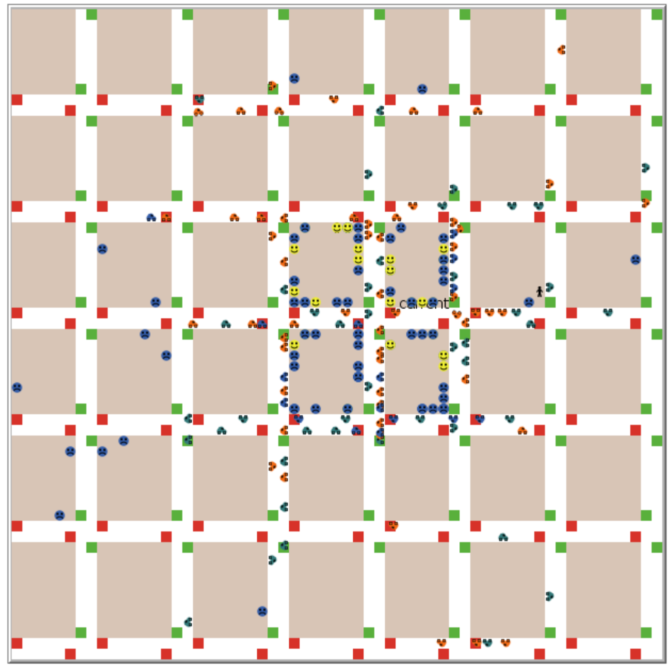

Fig. 4. Our modified traffic grid, with focus on the aforementioned changes

On the other hand, FIPA, which stands for Foundation for Intelligent Physical Agents is a body for developing and setting computer software standards for heterogeneous and interacting agents and agent-based systems. With the FIPA ACL Support extension, we created the possibility for two agents to efficiently communicate between each other, sending messages following the protocol in question.

F. Assumptions

Throughout the development of this project, we accepted several assumptions as being real and mirroring our society, yet being unchangeable by our system, impossible to simulate and analyze and completely irrelevant to the final result. Some of these are, for example, infinite money, infinite motor fuel, traffic lights being always respected, non-existing cars overtaking other cars, cars always making the same route, among others which could be inserted into the system but which would not prove useful to the problem we were trying to tackle.

G. Matching algorithm

After the simultaneous existing of persons and cars communicating between themselves in the system, we must decide on how two agents should match and how they are capable of responding to their intentions. This way, we could state that:

- A car can either offer itself for carpooling or not (this probability is defined by the observer). We only care about those who are carpoolers, the others existing only to influence the traffic.
- A car who offers their services for carpooling has a starting point, an ending point and a path by which it should travel along the simulation.
- An agent representing a user, generated in a patch different from a road, has the intention of going to a patch of a road.
- The implemented algorithm matches both a user and a car agent which passes by the patch of road the user wants to go to.
- After the match, the user is properly informed and should release a happy yellow smiley face, adding it to the statistics.
- If after a variable amount of time the user don't find a matching car, it releases a sad blue smiley, also adding it to the statistics.

V. Result analysis and discussion

A. Different results for different values

In order to test our hypothesis, we created different scenarios to see how the progressive changes in the variables would affect the final results.

To test how the system plots fluctuate in similar conditions with small changes in its environment, we maintained some variables unaltered, such as the grid size, traffic light existence, speed limit, maximum number of passengers allowed in a car and the percentage of cars who does carpooling. The values which we admitted were: grid size of a 7 per 7 matrix, percentage of carpoolers as 60% of the total existing cars and 5 seats existing in every car (which means a total of 4 available seats for carpooling excluding the driver). We will also assume that, in the presence of a more densely populated area, the total percentage of population existing in that area is of 80%. Besides that, we will also assume for this analysis that, in case of a car parking or a traffic accident, the average number of waiting ticks is 50 (meaning the system considers the aforementioned situations for an average number of 50 time units).

1) First case scenario: **Variables** - 50 cars, 50 persons, priority areas off, 0.6% of parking probability and 0.005% of accident probability. **Results** - After 10 seconds of simulation, there was a maximum number of 2 persons in 2 cars carpooling. There was a maximum number of 9 parked cars and no accidents.

2) Second case scenario: **Variables** - 50 cars, 50 persons, priority areas of 4 blocks, 0.6% of parking probability and 0.005% of accident probability. **Results** - After 10 seconds of simulation, there was a maximum number of 23 persons in 12 cars carpooling. There was a maximum number of 8 parked cars and no accidents.

3) Third case scenario: **Variables** - 50 cars, 50 persons, priority areas of 4 blocks, 0.9% of parking probability and 0.05% of accident probability. **Results** - After 10 seconds of simulation, there was a maximum number of 14 persons in 8 cars carpooling. There was a maximum number of 12 parked cars and 2 accidents.

4) Fourth case scenario: **Variables** - 50 cars, 50 persons, priority areas of 8 blocks, 0.6% of parking probability and 0.005% of accident probability. **Results** - After 10 seconds of simulation, there was a maximum number of 9 persons in 5 cars carpooling. There was a maximum number of 9 parked cars and no accidents.

5) Fifth case scenario: **Variables** - 50 cars, 50 persons, priority areas of 4 blocks, no parking probability, no accident probability. **Results** - After 10 seconds of simulation, there was a maximum number of 22 persons in 12 cars carpooling.

6) Sixth case scenario: **Variables** - 100 cars, 100 persons, priority areas of 4 blocks, 0.6% of parking probability and 0.005% of accident probability. **Results** - After 10 seconds of simulation, there was a maximum number of 57 persons in 31 cars carpooling. There was a maximum number of 18 parked cars and 1 accident.

B. Scenarios analysis

As expected, the system behaves accordingly to the social behavior one would expect. If no accidents occur and if all the cars are in constant motion (meaning no existence of parking) such as the one seen in the fourth case, then the number of people being served by a carpooler is greater than what would happen otherwise.

If we concentrate on how the number of people and cars available in the model influence the percentage of people being served by a carpooler, we can compare the second and the sixth cases. By just doubling the number of agents in the system, we observed how the percentage of people increased from 46% to 57%

Another fact we could observe was how priority areas influence the amount of people being served by the carpooling system (for the same percentage of people in a densely populated area). The sparser the area, the less served users, such as seen between the second and the fourth case scenarios (a difference between 46% and 24%).

VI. Conclusions

By considering the analyzed data through the six tests we conducted (described in the previous section), we were able to check how different situations, when related with each other, although minimal at first, can influence the outcome of many variables in the system of which we wouldn't even be counting. With all the mess that a big crowd of people and cars provoke observed from above, with people constantly and almost instantaneously being generated, searching for carpoolers or giving up their hopes of finding a suitable service, we could not perceive how a simple difference in population density influences the percentage of people being served by the carpooling system by almost two times. This simulation projects greatly help to understand how social and other type of phenomena work.

The main conclusion is that it is perfectly possible to integrate a carpooling system in a society under the right circumstances, once there's a suitable number of people to offer their services and there's a good matching algorithm between all the parties involved. Accidents and parking can influence the best possible outcomes, but usually, the more people, the merrier.

Future work we might lean ourselves on in the future is an approximation of this type of model to other cities, with various different road network systems distinguishable from our traffic grid. Therefore, it would also be interesting to implement multiple concentration areas in order to create more case scenarios by which the resulted simulation may provide more accurate results for the problem.

ACKNOWLEDGMENT

We would like to give the following people a special "Thank you!" regarding our project:

- Zafeiris Kokkinogenis - for invaluable help throughout the development of this project.
- Rosaldo Rossetti - for suggesting the theme and what approach we should follow (mainly regarding the implementation of BDI agents).
- CUFRA - for the refulgent Francesinha they served us in the day we wrote this acknowledgment section, greatly inspiring us.

References

[1] Proceedings 25th European Conference on Modelling and Simulation ECMS Tadeusz Burczynski, Joanna Kolodziej, Aleksander Byrski, Marco Carvalho (Editors)

[2] Marcelo Armendariz, Juan Burguillo, Ana Peleteiro, Gerald Arnould, Djamel Khadraoui, Carpooling: a Multi Agent simulation in netlogo. Dep. of Telematic Engineering CRP Henri Tudor, University of Vigo, 2012

[3] Ana Peleteiro*, Juan C. Burguillo and Marcelo Armendriz, Modelling and simulating a dynamic carpooling system for improving citizens mobility. Department of Telematic Engineering, University of Vigo, 2012

[4] A.M. Peleteiro-Ramallo, J.C. Burguillo-Rial, P.S. Rodrguez-Hernndez, E. Costa-Montenegro, Sin-City: a pedagogical testbed for checking multi-agent learning techniques, Department of Telematic Engineering, University of Vigo, 2010

[5] S. Tissue, U. Wilensky, NetLogo: Design and implementation of a multi-agent modeling environment, Center for Connected Learning and Computer-Based Modeling Northwestern University, Evanston, Illinois, c. 2004

[6] FIPA - Wikipedia, Wikipedia - The Free Encyclopedia, https://en.wikipedia.org/wiki/FIPA, accessed in June, 2017.

AUTHOR BIOGRAPHIES

António Pedro Fraga is a Portuguese BSc + MSc student in Informatics and Computing Engineering at the Faculty of Engineering at the University of Porto, currently studying on his fourth year out of five. His main interests are artificial intelligence, mobile development and physics.
E-mail: antoniopedrofraga@gmail.com
Website: pedrofraga.me "The greatest joy of a software developer is to realize that something works" - self.

Luís Oliveira is a Portuguese BSc + MSc student in Informatics and Computing Engineering at the Faculty of Engineering at the University of Porto, currently studying on his fourth year out of five. His main interests are artificial intelligence, software engineering and sports.
E-mail: luisoliveira8c@gmail.com
"It does not matter how slowly you go as long as you do not stop." - Confucius.

Pedro Martins is a Portuguese BSc + MSc student in Informatics and Computing Engineering at the Faculty of Engineering at the University of Porto, currently studying on his final year. He recently studied abroad for one semester at Budapest University of Technology and Economics. His main interests are artificial intelligence, data mining, databases, mobile development and watching movies. E-mail: pedrovlmartins@gmail.com "If the problem makes you want to give up, then make sure you never give up" - self.