A BDI-agent-based traffic simulation for the observation of Braess' Paradox

António D. Casimiro, Diogo D. S. Amaral, and Rui P. P. Cardoso
Artificial Intelligence and Computer Science Lab
Department of Computer Engineering
School of Engineering, University of Porto
Rua Dr Roberto Frias, s/n, 4200-465, Porto, Portugal
{antonio.casimiro, diogo.amaral, rui.peixoto}@fe.up.pt

***Abstract*—This work has followed the agent-based modelling and simulation (ABMS) paradigm, namely a BDI-agent-based approach to modelling drivers' behaviour in a road network, producing a traffic simulation. A phenomenon known as Braess' Paradox has been successfully observed in the simulation. The results could potentially lead to solutions to bottleneck issues, such as modifications in the network topology or the use of incentives in order to influence and possibly correct drivers' behaviour. Another goal of this research has been to find out whether the use of BDI agents and reinforcement learning approaches is appropriate and realistic enough to model real-world drivers.**

***Keywords*—BDI, simulation, car, agent, behaviour, network, road, driver, traffic, vehicle, Braess' Paradox, SUMO, TraCI, reinforcement learning, q-learning, softmax**

I. Introduction

In large-scale road networks, it is of the utmost importance to analyse bottlenecks, as well as the behaviours which drivers may exhibit in the presence of events such as traffic jams. The purpose of this research has been to simulate, in a controlled environment, the emergent behaviour of drivers in a traffic system in order to enable the optimisation of traffic assignment. A specific problem called **Braess' Paradox**[4] has been selected as the subject of this research.

This project is intended to suggest solutions to bottleneck issues, such as modifications to the network topology or the use of incentives in order to influence and possibly correct drivers' behaviour. Another key goal of this research has been to find out whether the use of BDI agents is appropriate and realistic for the modelling of real-world drivers.

The structure of this paper is as follows. Section II shall define the problem and its domain, presenting the main concepts. Section III shall provide some of the background knowledge which is required for the complete understanding of this paper. Section IV shall describe the steps which have been followed in order to produce the simulation which is the final result of this research. Section IV-A shall describe the system model constructed for this simulation. Section IV-B shall mention verification and validation. Section IV-C shall describe the implementation of both the model and simulation. Section IV-D shall describe the experiments which have been carried out with the simulation. Section IV-E is devoted to result analysis. Section V mentions work which is related to the research herein exposed. Section VI summarises the main conclusions extracted from this research.

II. Problem definition

This section provides a formal definition of the problem at hands. It shall walk the reader through the most significant concepts and present the traffic assignment problem as an optimisation problem.

A **driver** departs from a certain node in a road network and has the intention to arrive at a specific destination. This entity, which has been modelled as a cognitive agent, has learning and decision-making abilities, holds some knowledge about the network, which may be only partial, some goals which it will attempt to fulfil, and plans which it will try to execute in order to reach its goals.

A **road** is as a line segment where drivers are allowed to navigate, and it is characterised by two nodes, a certain number of lanes, each serving a specific direction, and a speed limit. A **network** is a set of roads. Each node in the network can be reached from any other node, which means that the network is connected.

A **traffic assignment zone** is an area to which a subset of the drivers intend to travel. In order to arrive at its destination, each driver must select a certain set of roads, whilst trying to finish its journey as quickly as possible.

For each agent, the objective function is the *minimisation* of the total travel time. Thus, the goal is to find a route A, which is a set of roads connecting the start and destination nodes ($A \subseteq E$, where E is the set of all roads in the network), such that the total travel time, $f' = \sum_{e \in A} t_e$, where t_e is the time elapsed while traversing road e, is minimum.

The simulation should not assume complete knowledge of the network. If a route is found to be the one with the least travel time from the source to the destination, this conclusion should not be final, since traffic assignment is dynamic and the travel time for any route could therefore change between two consecutive iterations of the simulation.

III. Preliminaries

This section provides some background knowledge which is required in order to fully understand the rest of the paper.

A. Graph and Network

The road network has been modelled as a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, such that $\forall(a, b) \in E, a \in V \wedge b \in V$. Each vertex in the graph corresponds to a node in the network and each edge corresponds to a road.

B. Deliberative Agents

Deliberative agents are a type of software agents used in multi-agent systems and simulations. They possess a symbolic model of the world and are capable of making decisions as a result of a reasoning process[15].

BDI agents are a kind of deliberative agents, characterised by a set of beliefs, desires, and intentions. The beliefs represent knowledge which the agent has acquired about the environment which surrounds it. The desires are the goals which the agent aims to achieve or maintain. The intentions are the plans which the agent will attempt to execute in order to achieve its goals. The belief, goal, and plan bases may be updated by the occurrence of external events. This type of architecture separates the selection of new plans from the execution of currently active plans and attempts to emulate the human cognition process by taking into account concepts, such as intention and belief, which are part of humans' nature[9]. Since the purpose for the agents in this simulation is to model real-world drivers, the BDI architecture has seemed appropriate.

C. The Q-learning Algorithm for Reinforcement Learning

Reinforcement learning is a type of machine learning for agents which attempts to influence their actions by maximising a cumulative reward. The models used for reinforcement learning consider a set of states, a set of actions, and a set of rules which determine an immediate reward for performing a certain action in a given state, which results in a state transition[8].

Q-learning is a reinforcement learning algorithm. It learns a utility function which assigns a value, called **q-value**, to each pair state-action, representing how good it is to perform an action in a given state. Greater q-values mean better choices, at least to the agent's best knowledge.

The core of the algorithm is a simple update of all q-values at the end of each iteration t, as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot (r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

In the expression above, α is the learning rate ($0 < \alpha \leq 1$), r_{t+1} is the reward observed after performing the action a_t in state s_t. The term $\max_a Q(s_{t+1}, a)$ is the estimate of an optimal future value and γ is the discount factor ($0 \leq \gamma \leq 1$), which sets the importance of future values to the current value[13].

There are many approaches to selecting an action in a certain state, given the q-values of all actions in the said state. A greedy approach, for instance, is based on selecting the action with the highest q-value for the given state. Since it is desirable for the agents to keep a certain degree of exploration, the approach followed in this research has been the random selection of an action with a certain probability. This probability of selecting an action, a_t, in a given state, s_t, is given by the following Softmax formula, where A is the set of all possible actions in state s_t, and $Q(s_t, b)$ is the q-value of performing the action b in state s_t:

$$P(s_t, a_t) = \frac{e^{\frac{Q(s_t, a_t)}{\tau}}}{\sum_{b \in A} e^{\frac{Q(s_t, b)}{\tau}}}$$

The parameter τ is called temperature. It should start with a high value and be decreased as the number of iterations of the simulation increases.

The BDI agents which have been used to model real-world drivers use Q-learning as a reinforcement learning technique for selecting the best route from the source to the destination.

D. Braess' Paradox

Braess' paradox is a phenomenon sometimes observed in road networks. In this situation, an apparent improvement to a congested road network could cause a redistribution of traffic which results in longer travel times. It was discovered in 1968 by mathematician Dietrich Braess, who proposed that, in systems in which each driver is making a self-interested decision as to which route is the fastest, the resulting Nash equilibrium may not be equivalent to the optimal flow through the network. This idea could help explain why adding a shortcut to a road network could actually increase the average travel time[4].

The authors of this paper have attempted to create a scenario in which this paradox could potentially be observed.

IV. Methodological Approach

This section goes through the methodological approach to the problem at hands which has been followed throughout this research. Firstly, a definition of the model shall be presented, which has been the basis for implementing the simulation. The process of verification and validation of this model shall also be mentioned. A short description of how the simulation has been implemented shall be provided. At last, the experiments which have been conducted with the simulation shall be explained and its results shall be presented and analysed.

A. Model Definition

1) Modelling Metaphor: This work has followed the agent-based modelling and simulation (ABMS) paradigm, namely a BDI-agent-based approach to modelling drivers' behaviour. As mentioned earlier, BDI agents have been selected to model real-world drivers' behaviour. The agents have as many states as the number of possible routes from the source to the destination nodes plus one, which is the initial state. In the initial state, the agents shall select one of the possible routes which could take each of them from its source to its desired destination. After this first choice, the state of each agent corresponds to the route which they have initially selected. At each intersection, the agent has the option of continuing to follow the current route or to change to another route, among those which also pass by that intersection.

The architecture of these agents may be described in terms of beliefs, desires, and intentions as follows.

The agents may hold beliefs on the following:

- Available routes from the source to the destination nodes.
- The q-value of performing an action in a given state.
- The current state (either the initial state or the current route).
- The set of actions which have been performed in each state during the current iteration.
- The total travel time after the agent has finished a trip.

The agents' goals, also called desires, are:

- Travelling from a source node to a destination node.
- Minimising the total travel time.

The agents' plans, also called intentions, are:

- Starting a journey, choosing the initial route.
- Choosing whether to stay in the current route or to change to another one when an intersection is approached.
- Updating the q-values of each pair state-action.

The last plan listed above is concerned with the Q-learning algorithm, explained in section III-C. The last action performed by each agent has a reward which is equal to the inverse of the total travel time, which is measured within the simulation and divided by 1000. All the other actions have a reward of 0.

2) Entities and relationships: A simulation is characterised by a set of entities, which have attributes, and a set of relationships between these entities. This section presents these entities and relationships.

The entities of the simulation are the vehicles, which are not distinguished from their drivers, as well as each road in the network. A vehicle is characterised by its class (passenger, trailer, coach, etc.), size, acceleration, deceleration, maximum speed, and the origin and destination nodes for its trip. A road is described by the two nodes which define it, its speed limit, and the number of lanes, as well as the direction of each lane.

The relationships between the entities are as follows. A vehicle is located in a road network. For each vehicle, the route which it is currently following must be valid, which means that the destination node must be reachable from the origin node in the network.

3) Decision variables: Decision variables, or controllable input variables, represent the set of aspects of the simulation which may be realistically controlled in order to produce different results. These, together with uncontrollable input variables, which are described in more detail in section IV-A4, are used to obtain the output variables of the simulation, which are detailed in section IV-A5.

In this case, the decision variables are concerned with the topology of the network, namely which roads exist, the number of lanes in each road, the speed limit for each road, and, of special importance for the study of Braess' Paradox, whether or not to add a faster road which might be used as a shortcut for the vehicles.

4) Uncontrollable input variables: Uncontrollable input variables represent aspects of the simulation which may not be controlled. These are defined by external processes, which could be modelled by probabilistic distributions.

The authors have identified the following uncontrollable input variables:

- The number of cars which are spawned in the network.
- The origin and destination nodes for each car.
- All the attributes of a vehicle, which have been listed in section IV-A2.

5) Output variables: The simulation produces output variables which are used as performance measures. These variables are the following:

- The total travel time for each agent.
- The average travel time in the network.

These variables are the basis for measuring the performance of the system.

6) System states: As mentioned before, each agent has its own internal state, which is either the initial state or the route it is currently following.

The system as a whole may go through several states. The state in which the vehicles controlled by the agents travel freely and may increase their speed up to the speed limit of the roads they are currently traversing is called **free-flow state**. Since the driver agents implement reinforcement learning, the system is expected to reach states in which no agent may increase the speed at which it is travelling without forcing at least one other agent to reduce theirs. This is known as a Nash equilibrium. These states are called **steady states**. A steady state may be a **normal state** or a **congestion state**. In order to distinguish between these two sorts of steady states, a threshold on the average speed of the vehicles might be defined. If the average speed of the vehicles is greater or equal than that threshold, the steady state should be considered a normal state. Otherwise, it should be considered a congestion state.

B. Verification and Validation

Verification of both the model and the simulation has occurred in the form of tests during their implementation. These tests have been concerned with the assessment of the correctness of the model, such as whether the agents' belief set is correctly updated upon certain events.

The validation process should involve the simulation of an actual city traffic system whose behaviour can be observed. Data should be collected in order for the traffic demand generated in the real-world network to be correctly modelled and simulated. The observed outputs of both the real system and the simulation would then be compared, so that conclusions regarding how well the latter reproduces the former could be stated. This activity has not been carried out by the authors, since the required resources have not been available.

C. Implementation of the Model and Simulation

The implementation of the model has been based on a combination of different tools.

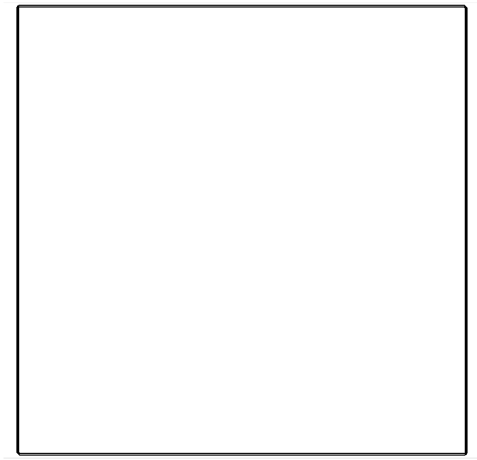

Fig. 1. Original road network.

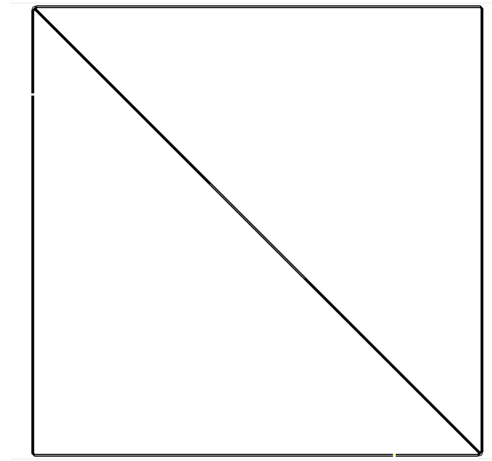

Fig. 2. Modified road network with an extra road.

The main application, which has been written in `Java`, is responsible for the aspects regarding the simulation, such as its setup or the creation and initialisation of agents.

`AgentSpeak`, an agent-oriented programming language, has been used for the definition of the BDI agents which represent the vehicles. `AgentSpeak` is a language which is implemented by several tools. The `Jason` tool has been used in this work, since it provides an interpreter for the `AgentSpeak` language and the implementation of a reasoning engine.

`SUMO` has been the simulation tool selected for this project, since it is exclusively suited for the implementation of road traffic simulations. It implements all the details related to traffic flow, such as traffic rules, enabling users to be concerned only with the higher-level aspects of a traffic simulation. It includes a graphical user interface under which an animation of the simulation may be observed. It also provides some utilities, namely `NETEDIT`, a tool which enables a quick and easy construction of a road network in a format which can be understood by the simulation engine. `SUMO` exposes an API, called `TraCI`, which allows third-party applications to launch and control simulations. A `Java` implementation of this API, called `TraCI4J`, has been used in this work, allowing each `Jason` agent to control a vehicle in the `SUMO` simulation.

Figures 1 and 2 show the two simple road networks which have been constructed with the `NETEDIT` tool. The original road network has been modified with an extra road, so that the system could be tested for a potential instance of Braess' paradox. In the simulation, all agents depart from the bottom left corner and their destination is the top right corner.

D. Experimentation

Experiments have been carried out on the simulation with two main goals. The first has been to observe an instance of Braess' Paradox. The second has been to verify whether the use of Q-learning as a reinforcement learning technique is suited for leading the agents into minimising their total travel time. Experiments have been carried out with both road networks presented in the previous section.

TABLE I
AVERAGE TRAVEL TIME IN SIMULATION SECONDS

Learning Rate	Discount Factor	Average Travel Time (original road network)	Average Travel Time (modified road network)
0.25	0.1	5913.2764	7568.993
0.25	0.25	5912.6596	7521.4264
0.25	0.5	5913.678	7510.4988
0.25	0.75	5913.0546	7479.6914
0.5	0.1	5913.515	7526.3932
0.5	0.25	5912.9588	7719.8068
0.5	0.5	5912.8976	7494.82
0.5	0.75	5913.552	7642.313
0.75	0.1	5913.6798	7621.4988
0.75	0.25	5912.7446	7497.5506
0.75	0.5	5913.7396	7537.3944
0.75	0.75	5913.4378	7609.7346

Several combinations of the learning rate and the discount factor, the parameters for the Q-learning algorithm, have been tested as an attempt to find out which would result in the least average travel time, considering all the vehicles.

The results are presented in the following section.

E. Result Analysis

Table I shows the average travel time, in simulation seconds, obtained for different values of the learning rate and discount factor, after 5 iterations of the simulation.

These results seem to suggest that an instance of Braess' Paradox has been observed. The average travel time for the vehicles in the original road network is less than when a new road, which could be interpreted as a shortcut, is added to the network. The redistribution of **the same number** of vehicles in the modified network causes the average travel time to increase, since the agents which control the vehicles are making self-interested decisions as to which route is the quickest. Moreover, there is not a significant difference between the results obtained with different values for the

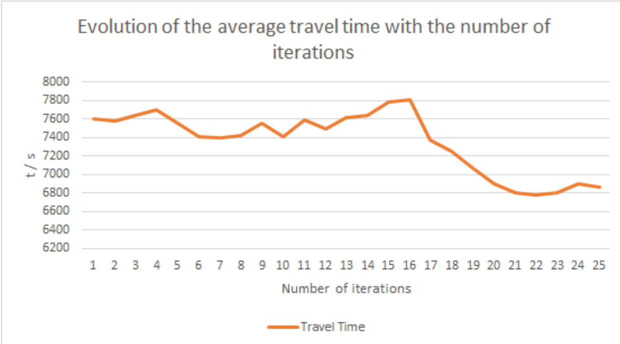

Fig. 3. Evolution of the average travel time with the number of iterations, in simulation seconds.

learning rate and the discount factor after only 5 iterations of the simulation.

The chart of figure 3 shows how the average travel time for vehicles travelling in the modified network evolves with the number of iterations of the simulation and, consequently, the Q-learning algorithm. These measures have been performed for 25 iterations, with a learning rate of 0.5 and a discount factor of 0.25.

The average travel time tends to decrease after around 16 iterations and it seems to converge to a stable value after around 20 iterations. Therefore, the Q-learning algorithm seems to be a sufficiently effective machine learning technique for the agents to learn the routes which minimise their total travel time.

V. Related Work

The literature includes many reports of cognitive agents (among which, BDI agents) being used to model mental and social characteristics of humans. They could potentially implement a reasoning process similar to ours, and, therefore, enable the development of more realistic social models and simulations. Abstract languages such as `AgentSpeak` have been proposed to model cognitive agents due to their elegance and solid theoretical foundations. Some authors have noted the adequacy of using agent-oriented techniques for modelling traffic and transportation systems, given the highly dynamic and distributed nature of these systems. These models are important for the development of simulations which provide valuable insight about the factors capable of influencing traffic movement and the behaviour of drivers, which could enable a better management of road networks.

VI. Conclusions and Future Work

This paper proposes an agent-based model and simulation for the analysis of possible solutions to bottleneck issues which may arise from the traffic assignment problem. Braess' Paradox has been considered and observed in simulations during the course of this research. Taking into account the nature of an agent-based model and simulation, as well as the need for modelling real-world drivers, BDI agents have been used, given their ability to emulate human behaviour. The agents have used Q-learning as a reinforcement learning technique, enabling them to select the best routes from the source to the destination nodes.

Several experiments have been carried out on the simulation so as to draw conclusions concerning Braess' Paradox and the usage of Q-learning as a reinforcement learning algorithm. The experiments have shown that adding a shortcut to the original network actually increases the average travel time, an observation which matches the definition of Braess' Paradox. This enables the authors to state that, if each vehicle follows a non-cooperative strategy in order to select a route, the resulting flow through the road network may not be optimal. This conclusion is important for the design of autonomous vehicles, in which cooperative strategies seem to be the most appropriate. It has also been observed that the average travel time decreases after some iterations of the simulation, which has led the authors to the conclusion that Q-learning is an appropriate reinforcement learning technique for this scenario.

One possible future course of investigation is to consider larger road networks, whose analysis may be of great importance for acquiring a better insight regarding traffic bottlenecks in large cities. Simulations regarding the possibility of agents cooperating, instead of making self-interested decisions, could be very interesting for the study of autonomous vehicles. The authors also find it appropriate to seek to improve the agents' learning abilities.

References

[1] T. A. Arentze and H. J. P. Timmermans. *Albatross: a learning based transportation oriented simulation system*. Eindhoven, The Netherlands: EIRASS, 2000.

[2] R. Arnott, A. de Palma, and R. Lindsey. *Does providing information to drivers reduce traffic congestion?* Transportation Research, 1991.

[3] A.L.C. Bazzan, J. Wahle, and F. Klügl. *Agents in traffic modelling: from reactive to social behaviour*. Berlin: Advances in Artificial Intelligence (KI-99), LNAI 1701, 1999.

[4] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. "On a Paradox of Traffic Planning." In: *Transportation Science* 39.4 (Mar. 4, 2010), pp. 446–450. URL: http://dblp.uni-trier.de/db/journals/transci/transci39.html#BraessNW05.

[5] *Application of multi-agent systems in traffic and transportation*. IEEE Proceedings on Software Engineering, 144 (1). 1997, pp. 51–60.

[6] *DRACULA: dynamic route assignment combining user learning and microsimulation*. Vol. E. Proceedings of the 23rd European Transport Forum. 1995, pp. 143–152.

[7] R. Machado and R.H. Bordini. *Running AgentSpeak(L) agents on SIM_AGENT*.

[8] Martijn van Otterlo and Marco Wiering. "Reinforcement Learning and Markov Decision Processes". In: *Reinforcement Learning: State-of-the-Art*. Ed. by Marco Wiering and Martijn van Otterlo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–42. ISBN: 978-

3-642-27645-3. DOI: 10.1007/978-3-642-27645-3_1. URL: http://dx.doi.org/10.1007/978-3-642-27645-3_1.

[9] Anand S. Rao and Michael P. Georgeff. "BDI Agents: From Theory to Practice". In: *Proc. 1st Int'l Conf. on MultiAgent Systems*. Ed. by Victor Lesser and Les Gasser. San Francisco, CA: MIT Press, 1995, pp. 312–319. ISBN: 0-262-62102-9. URL: https://www.aaai.org/Papers/ICMAS/1995/ICMAS95-042.pdf.

[10] Anand S. Rao and Michael P. Georgeff. "Modeling Rational Agents within a BDI-Architecture". In: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*. Ed. by James Allen, Richard Fikes, and Erik Sandewall. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1991, pp. 473–484. URL: http://jmvidal.cse.sc.edu/library/rao91a.pdf.

[11] R.J.F. Rossetti and R. Liu. *Activity-Based Analysis of Travel Demand Using Cognitive Agents*. Ed. by Elsevier. Oxford, UK: H. Timmermans (Ed.) Progress in Activity-Based Analysis, 2005.

[12] R.J.F. Rossetti et al. *Using BDI agents to improve driver modelling in a commuter scenario*. Oct. 2002.

[13] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.

[14] D.P. Watling. *Urban traffic network models and dynamic driver information systems*. CKBS-SIG Workshop on Cooperating Knowledge Based Systems, 1994.

[15] Michael Wooldridge and Paolo Ciancarini. "Agent-oriented Software Engineering: The State of the Art". In: *First International Workshop, AOSE 2000 on Agent-oriented Software Engineering*. Limerick, Ireland: Springer-Verlag New York, Inc., 2001, pp. 1–28. ISBN: 3-540-41594-7. URL: http://dl.acm.org/citation.cfm?id=370834.370836.