



Human-agent Interaction based on Game Theory: Case of a road traffic supervision task

Martial Razakatiana, Christophe Kolski, René Mandiau, Thomas Mahatody

► To cite this version:

Martial Razakatiana, Christophe Kolski, René Mandiau, Thomas Mahatody. Human-agent Interaction based on Game Theory: Case of a road traffic supervision task. 13th International Conference on Human System Interaction (HSI 2020), Institute of Electrical and Electronics Engineers, Jun 2020, Tokyo, Japan. pp.88-93, 10.1109/HSI49210.2020.9142687 . hal-03331828

HAL Id: hal-03331828

<https://uphf.hal.science/hal-03331828>

Submitted on 14 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Human-agent Interaction based on Game Theory: Case of a road traffic supervision task

Martial Razakatiana
LIMAD-EDMI
Université de Fianarantsoa.
BP 1264 Fianarantsoa,
Madagascar rzkmartial@gmail.com

Christophe Kolski
LAMIH-UMR CNRS 8201
Univ. Polytechnique Hauts-de-
France,
Le Mont Houy,
59313 Valenciennes cedex 9,
France christophe.kolski@uphf.fr

René Mandiau
LAMIH-UMR CNRS 8201
Univ. Polytechnique Hauts-de-
France,
Le Mont Houy,
59313 Valenciennes cedex 9,
France
rene.mandiau@uphf.fr

Thomas Mahatody
LIMAD-EDMI
Université de Fianarantsoa.
BP 1264 Fianarantsoa,
Madagascar
tsmahatody@gmail.com

Abstract—This work contributes to the field of human-system interaction modeling through an artificial intelligence approach. It focuses on the cooperative realization of a complex task. For this purpose, we propose a human-agent interaction model based on game theory to describe the decision making between the human operator and the assistant agent. The proposed model is based on searching Nash equilibria for a repeated two-player game in which each player has a choice between two actions. In particular, the assistant agent knows how to calculate the equilibrium that depends on information coming from the context (human operator and work environment). This approach allows us to consider a context aware human-machine system. Then, the assistant agent knows how to optimize its intervention with regard to the human operator assisted by this agent during a complex task. For the validation of our model, we highlight the efficiency of the assistant agent using this principle by considering a road traffic simulator using Netlogo. An analysis of the simulation results is provided to illustrate the effectiveness of our approach.

Keywords—*intelligent agent, multiagent system, game theory, road traffic supervision, simulation.*

I. INTRODUCTION

Human-computer interaction (HCI) is becoming increasingly important in the design and implementation of new industrial interactive devices and assistance systems [1]. Since the emergence of Artificial Intelligence (AI) at the end of the seventies in the world of interactive systems (cf. for example [2]), the need for efficiency and user-friendliness with interactive systems has been increasing. This cross-fertilization between HCI and AI leads to co-called intelligent, personalized or context-aware interactive systems. The new user interfaces constitute devices, aiming at the user being able to interact with the system with ease, assisted by an agent [3], which is intelligent in the sense of AI. Thus, beyond the visual aspects, human-user interaction models are implemented. This implies an important dimension of research on human-machine interaction: the underlying software mechanisms and the principle of adaptation to ensure intelligent interaction. Several software mechanisms based on AI models have been used through more and more case studies in intelligent interactions [4]: inference engine, fuzzy systems, Markov models, and so on, whether in a centralized or distributed framework (in particular, multi-agent systems (MAS) approaches). Intelligent agents already have their place in several application domains: e-commerce, Intelligent Transport Systems (ITS), simulation, and so on [5]. If we focus on road traffic simulation, several studies are based on MAS approaches. For instance, in [6] the authors proposed a multi-agent approach using the ant colony principle to manage the traffic flow. In addition, several approaches use intelligent agents on traffic signal controls and commands at each intersection as in [7], [8] and [9]. However, none of such approaches takes explicitly into account information from human operator in charge of traffic control in a specific sector; that can lead to poorly coordinated activities between human beings and intelligent agents. In this research, we are interested in distributed interaction models for the implementation of human-agent interaction systems. In particular, we focus on the intelligent agent behavior towards a human being, in the context of a complex task with critical situation.

After this introduction, section II provides a state of art on intelligent agents in road traffic simulation. Section III proposes the principle of interaction between humans and intelligent agents, using game theory. Section IV presents the framework of the simulation involving a human being and an intelligent agent in traffic supervision. The results obtained are described in Section V. Finally, the last section provides a conclusion and research perspectives.

II. STATE OF ART ON INTELLIGENT AGENT IN ROAD TRAFFIC SIMULATION

Several definitions about agents are available in the literature [10] [11] [12] [13]. An agent is generally defined as an entity capable of reacting in a given environment using its sensors [10]. By extension, an intelligent agent is a particular agent that can act without human intervention [11]. This work focuses on their use in the case of road traffic simulation.

Congestion is a difficult problem to solve in the case of urban road traffic supervision. Several studies have focused on the use of intelligent agents to solve traffic problems. Traffic light management was the main topic on the use of intelligent agents [7], while using a technique to predict future traffic conditions and vehicle movements [8] [14]. To reduce the extent of congestion at road intersections, Yeo *et al.* [15] propose an improvement in traffic by trying to capture traffic information around the intersection. In addition, a multi-agent approach has been proposed to solve these traffic problems, as in [9], [16] and [17]. In [18], the authors propose Co-Fields, a coordination model for multiagent systems; it is used to guide the cars toward their destinations avoiding the most crowded areas.

Other solutions based on intelligent agents can be mentioned, for example: the use of Internet of Things (IoT) for Intelligent Transport Systems (ITS) [19], the use of virtual agents able to interact with tangible objects in traffic simulation on RFID (Radio Frequency IDentification) tabletops [20]. In [14], intelligent agents, each dedicated to traffic lights, have a reinforcement learning mechanism. In addition, a solution to the problem of intersection management is proposed by [16] and [21]; these authors use a payoff matrix that allows each vehicle to decide whether to stop or continue (go, stop). In order to improve the road traffic situation, Wang *et al.* propose a virtual vehicle coordination approach based on the theory of group consensus to sense environment information efficiently [22]. In [23], a model of a network of intersections is proposed in order to obtain deadlock-free traffic with a specified route. Finally, a recent study reviews the state-of-the-art around the application of bioinspired methods in the field of ITS [24].

However, none of the above-mentioned approaches explicitly takes into account the human operator to accomplish the traffic road supervision. On this subject, a human operator can be characterized by several aspects or attributes depending on the context modeling [25]. For example, in the user profile, one can consider the level of knowledge in relation to the functioning of the system with which the user interacts [26]. The level of instantaneous workload can also be taken into account, for example according to five levels as in [27]. It is also possible to take into account user preferences, goal(s), level of expertise, etc. The following proposal aims to take into account explicitly the human operator's actions in the context of human-agent interaction.

III. PROPOSAL OF A MODEL BASED ON GAME THEORY

In this section, a multiagent interaction model, in which several human beings and software agents may be located, is proposed. However, in this paper, we focus the study on a system composed of one human being and one software agent. After a brief introduction on game theory, a formalization of the system is presented.

A. Definition of a two-player game

A two-player game is defined as a situation in which two participants can each make their own decision; one's payoff depends not only on his or her action; it also depends on the other's action. A game is characterized by the following elements:

- 1) *The players*: These are the participants.
- 2) *Actions for each player*: These are the choices available for each player at each move of the game.
- 3) *The rules of the game*: they determine how the game is played and the rules to be respected: each player may play simultaneously or successively.
- 4) *The type of game*: cooperative or non-cooperative. It is said to be cooperative in the case where participants can communicate with each other in order to be able to decide together on the strategies to adopt in order to win; it is non-cooperative in the opposite case.

B. Interaction model based on game theory

Game-based coordination allows each participant to choose the appropriate actions in response to the actions of other players (in our case, only one other player). In this type of game, we assume that

the agent is rational. The equilibrium of the game is a set of optimal strategies for each player. The search for this equilibrium thus allows each agent to decide on the optimal action to perform using the game matrix. The most commonly used is Nash equilibria [28], [29]. In this case, Nash equilibria refers to the strategy in which neither player has an interest in unilaterally changing his or her strategy.

C. Presentation of a game

A game can be represented as a payoff matrix whose elements are the players' payoffs (Table 1). In Table 1, the participants of the game are called Player 1 and Player 2. Then, $act11$ and $act12$ are the possible actions for Player 1; $act21$ and $act22$ are the possible actions for Player 2 [21] [28]. Thus, the payoffs (defined by utilities) of each player are expressed as follows: $i \in \{1,2\}$:

- $U_i(act11, act21)$ refers to the payoff of player i when Player 1 plays $act11$ and the other player plays $act21$
- $U_i(act11, act22)$ refers to the payoff of player i when Player 1 plays $act11$ and the other player plays $act22$
- $U_i(act12, act21)$ refers to the payoff of player i when Player 1 plays $act12$ and the other player plays $act21$
- $U_i(act12, act22)$ refers to the payoff of player i when Player 1 plays $act12$ and the other player plays $act22$.

TABLE 1. PAYOFF MATRIX

		Player 2	
		$act21$	$act22$
Player 1	$act11$	$U_1(act11, act21)$ $U_2(act11, act21)$	$U_1(act11, act22)$ $U_2(act11, act22)$
	$act12$	$U_1(act12, act21)$ $U_2(act12, act21)$	$U_1(act12, act22)$ $U_2(act12, act22)$

Given a system modeled by a game, the participants should find the best possible result by trying to maximize their payoffs. One method for optimizing the solution of a game is to look for Nash equilibria [29], for the benefits of all participants. To determine an equilibrium, we choose the algorithm described in [30] [31] (its explanation will not be detailed here).

IV. TWO-PLAYER GAME FORMALIZATION

In this section, we propose a model of intelligent interaction between an agent and a human user. The proposed interaction model is based on the principle of game theory in the case of multi-agent resolution of a given problem. The players are: a human being and an intelligent agent assisting him or her in the completion of the task. For both players, several possible actions can be considered as part of the interaction. However, to simplify the game, we will summarize it in only two actions: the cooperation that we later note by (C) and the noncooperation noted by (D). Our system will therefore be modeled by a game with two players and two possible actions for each player [31].

a) Description of the players' actions

For the human being, (C) means a request for help or an expectation of advice or relevant information that may influence his or her decision. (D) means that he or she is autonomous: in this case, the human being simply acts without asking for the help of the assistant (software agent).

For the software agent: (C) means that it acts directly or makes a proposal to the human being about what he or she can do. (D) means that the agent does nothing (it does not respond) by leaving the human being to fend for himself or herself, without assistance (in consequence the human being can learn or continue to learn by doing, which is similar to the concept of learning by doing [32]).

b) Construction of the payoff matrix in two-action game

For the calculation of the matrix, we propose the following formulas, in which:

- wl refers to the user's workload level (Five levels, according with [27]: *No activity* (level 1), *Relaxed* (level 2), *Comfortable* (level 3), *High* (level 4), *Excessive* (level 5));

- *crt* refers to the criticality level of the situation (Five levels, according with [33]: *Insensible* (level 1), *Harmless* (level 2), *Unpleasant* (level 3), *Dangerous* (level 4), *Uncontrollable* (level 5));
- *usl* refers to the user's experience level (Five levels, according with [26]: *Novice* (level 1), *Advanced Beginner* (level 2), *Competent* (level 3), *Proficient* (level 4), *Expert* (level 5)).

In our two-game based model, we have chosen the human payoff values according to the following adaptation principles:

- If the user does not have enough experience on the task (low experience level: less than level 3), it is preferable for the software agent to act, especially if the workload level and the criticality level are high.
- If the user has a high level of experience in the field: (values between 3 and 5) in which he or she is involved, he or she does not need assistance, especially if his or her workload level and the criticality level are low.
- The human being's payoff that requires cooperation is proportional to the level of his or her workload and to the criticality level.
- If the user is autonomous (work without assistance by the agent), he or she reduces his or her payoff according to the criticality level of the situation, especially if the user does not have enough experience, he or she takes risks not to ask the agent for help (i.e. for assistance).

For the human being, we therefore propose the human utility function noted $U_H(xh, xa)$, with xa the response of the software agent to the human user's action xh .

- $U_H(C, C) = wl + 2 \times crt - 2 \times usl$
- $U_H(D, C) = usl - (wl + crt)$
- $U_H(C, D) = usl - (1 + crt)$
- $U_H(D, D) = usl - crt$

For the software agent, the utility function is noted $U_A(xh, xa)$, with xh the user's action and xa the software agent's response:

1) If the user requests help, the assistant's payoff, if any, is proportional to the workload and criticality level.

2) If the user has a high level of experience, it can be considered that he or she does not need assistance, especially if the workload and criticality levels are low. This allows the assistant not to waste time (and to let the human learn by doing, in the sense of [32]).

3) If the human being requests cooperation and the assistant does not respond, the assistant may consider that the human being does not need help, except in the case of high workload and criticality levels.

4) If the user is autonomous, he/she reduces his or her payoff according to the criticality level of the situation,

$$\begin{aligned} U_A(C, C) &= wl + crt - usl \\ U_A(D, C) &= usl - wl \\ U_A(C, D) &= -wl \\ U_A(D, D) &= usl - crt \end{aligned}$$

Thus, the cooperative game involving the human operator and the assistant agent is modeled by a repeated game of N moves, where N is the number of iterations considered in the case study. The average payoff of human operator depends on the decision history of both players:

$G_H = \frac{1}{N} \sum_{i=1}^N U_H(x_i, \bar{x}_i)$ where: x_i the action chosen in the i -th move by the human being and \bar{x}_i denote the choice of assistant agent.

For the assistant agent, the average payoff is given by the following formula: $G_A = \frac{1}{N} \sum_{i=1}^N U_A(\bar{x}_i, x_i)$ where: x_i designates the action chosen in the i^{th} round by the assistant and \bar{x}_i for the human operator. U_H and U_A are the two utility functions proposed above.

In the following section, we consider a system composed of a software agent and a human operator in the context of traffic supervision simulation.

V. CASE STUDY USING NETLOGO

A. Presentation of Netlogo platform

Netlogo is a multi-agent programmable modeling environment. The choice of this platform in our work is justified given its simplicity in terms of agent creation and implementation. In addition, it supports multi-agent collaborative interaction via its client-server architecture across the network. The simulation is based on the work of Wilensky and Stroup [34] according to the concept of participatory simulation. Human beings take on the role of clients that can connect via a hubnet server. Software agents can take the initiative to act at a traffic junction or send messages to human operators.

B. Description of the traffic simulator

a) The server

The main simulator provides an overview and control of the road traffic. According to the simulator user interface (server), it is possible to create software agents for road traffic management. To do this, we use the create-new-assistant button: a wizard is created randomly at any intersection position. To be able to choose the intersection for the new assistant, we can use the “choose-inter-ass” button: this allows choosing the intersection of the newly created assistant agent with the mouse (the assistant is created).

The “suppr-ass” button allows you to delete an assistant from all assistants. On the server (Fig. 1), it is possible to visualize the values of the traffic information: The average speed of vehicles, the number of intersections, the number of human beings involved in traffic management, the number of software agents involved, the number of accidents (vehicles in collision), the average level of human workloads, the controls allowing to modify some parameters (speed limiter, acceleration...).



Fig. 1. User interface of the simulator (hubnet server)

On the right-hand side of the server user interface (Fig. 1), one can also see the evolution of the Nash equilibria, which changes dynamically according to the game payoff matrix. This matrix will reflect the evolution of parameters such as the user's workload and experience levels (which will have consequences on the values in the matrix). On the far right of interface, one can see in real time the evolution of the maximum number of stopped vehicles waiting for the entry of an intersection. The “go” button is used to launch the simulation of the traffic.

Thus, the criticality level is calculated from a set of information from the traffic simulator: for example, the density of vehicles, the number of stopped vehicles. This level can take one of ten possible values, divided into five levels as expressed above: *Insensible*, *Harmless*, *Unpleasant*, *Dangerous*, *Uncontrollable*.

b) The client.

Several clients (i.e. user interfaces) can be connected at the same time for traffic management. The client allows the human being to intervene on the simulator. So, the client can modify the speed limit via a slider, choose an intersection and change the colors of the latter's lights (Fig. 2). The client also allows modifying the workload level of the human being, as well as the level of his or her profile.

c) Software agent

The software agent is an assistant agent created to help the human being in the context of traffic supervision. This agent intervenes following the human decision, by exchanging the colors of the lights at one or more intersections. For this purpose, the following algorithm is implemented in the simulator:

- 1) Update of the game payoff matrix according to the proposed and/or estimated information.

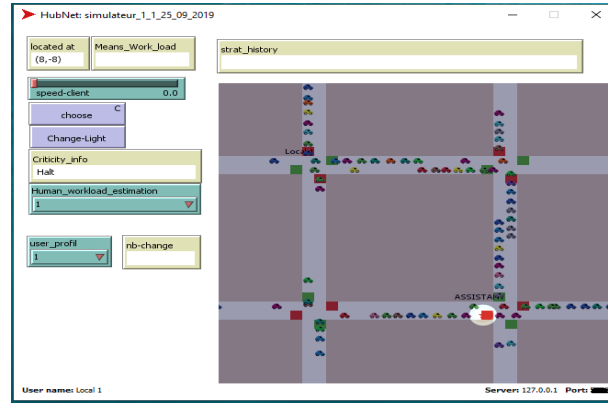


Fig. 2. User interface of the simulator (*hubnet* client)

2) Finding an equilibrium.

3) Based on this equilibrium, the agent chooses an action adapted to the current application (for example, it evaluates the traffic flow, and changes the colour of the lights for a given intersection).

4) Repeat the analysis after a certain number of cycles.

C. Results of the simulation and interpretation

The system consists of an intelligent agent and a human being connected as a hubnet client, with a number of vehicles equal to 84. This number (84) was chosen for the following reason: in the simulator, as many vehicles as desired can be created, but it is important not to exceed the available spaces (roads). In addition, to have a criticality level threshold, the number of vehicles must be large. In our simulation, we assume an environment with four intersections (it is possible to have many more; but we think that this choice is sufficient to validate our idea). That leads to generate about 84 vehicles while keeping an environment that can easily evolve (if the number of vehicles is too large, it would be necessary to define a space occupation management and an anticipation mechanism for agents that would exceed the problem of this paper, see for example [35]).

Then, traffic behavior does not depend on the number of iterations. It only depends on the actions of the human being and whether or not the assistant agent intervenes, according to the Nash equilibria; the assistant agent calculates these equilibria at each iteration. Of course, the human being does not calculate this equilibrium. The assistant agent calculates the Nash equilibria and then makes the decision to intervene or not. If the agent judges that the intervention is not necessary, it may take care of other tasks, which conditions the result during a current iteration. We believe that the choice of 50 for the number of iterations is sufficient to be able to vary the parameters of the matrix, and show the interest of the approach. However, we can of course go beyond 50 depending on the duration of observation. Thus, the average payoff of the assistant and that of the human being depend on the number of iterations.

The results provided below are obtained without any human intervention. Fig. 3 thus shows the evolution of the number of interventions of the agent in relation to the evolution of the state of the human being during iterations:

- **work_load_level:** refers to the level of human workload (this level can be modified on the client interface).
- **user_experience_level:** level of experience of the human being.
- **Number_of_coop:** number of interventions of the software agent.

Thus, the curve visible in Fig. 4 shows the evolution of the number of vehicles stopped according to the instantaneous equilibria conditioned by the context information considered for 50 iterations. As explained above, the number of iterations is sufficient in this context. The evolution of the number of interventions of the agent depends on the workload level and the user's experience level:

- Between iterations 11 and 21, Fig. 3 shows that the software agent does not intervene: this is justified because during this period, the level of experience is high while the workload level is low.
- There are not many more interventions between iterations 21 to 30 (for the same reasons).
- From 31st iteration, the number of interventions increases. This is due to an increase in the level of human workload compared to the user's level of experience.

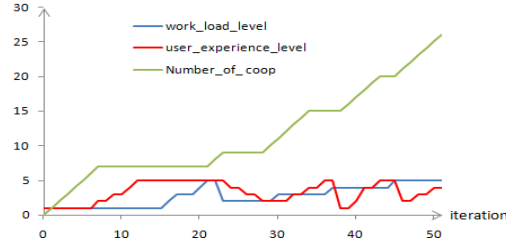


Fig. 3. Workload level, User experience level and Numbers of interventions of the software agent.

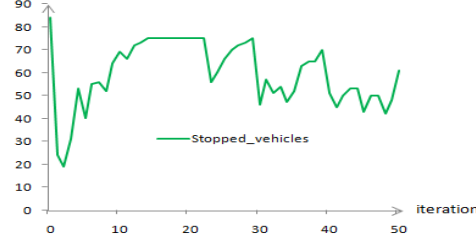


Fig. 4. Number of stopped vehicles

Thus, the curve in Fig. 5 shows the evolution of the average vehicle speed for the 50 iterations of observations.

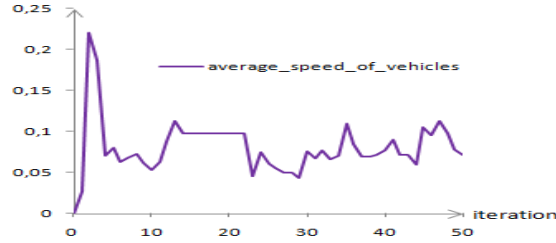


Fig. 5. Average speed of the vehicles

D. Results concerning the Nash equilibrium

During the 50 cycles, we can see (Fig. 3) the variation of Nash equilibria according to the evolution of the context information that determines the payoff matrix of the game: the level of experience of the human being and his or her workload level. For the number of iterations, we can have any number of iterations; but for the sake of clarity of the evolution of the number of interventions, the average speed, the average waiting time of the vehicles, we have considered on Figures 3 to 6 about fifty iterations.

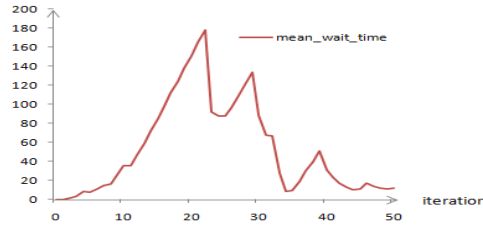


Fig. 6. Average waiting time of vehicles

As visible in Fig. 3, the evolution of the Nash equilibrium is translated by the evolution of the number of interventions of the assistant. This number increases in the case of intervention (CC) for the current iteration. Otherwise, the number of interventions does not change; but in the case (DD), the assistant leaves the human operator to deal with other activities.

Scoring in the simulation results: Nash equilibria in the case of two players are given in the form of [CC], [CD], [DD], [DC]. For each observation cycle, the task of the software agent is to identify first the crossroads that have the maximum number of vehicles waiting for the entrance. Then the agent changes the lights: the green lights turn red and vice versa:

[CC]: means that the human being is waiting for help and that the software agent is acting at one or more traffic junctions.

[CC] and [CD]: mean that the human being may or may not be autonomous; but the assistant always acts as long as there is cooperation in the Nash equilibrium calculated at each iteration.

[DD]: means that the human being is autonomous and that the assistant does nothing. In this case, the human being must act because the assistant will not do anything (for example, the agent can solve other traffic problems).

E. Interpretation

At any time, the software agent calculates the game payoff matrix. Then it determines the equilibrium and from it, the agent can examine the action of the human being and then it reacts to this action in relation to the Nash equilibria obtained. Fig. 4 shows that the increase in the number of assistant interventions leads to a decrease in the number of stopped vehicles during the simulation. This result explains the fact that if the human being needs the intervention of the assistant, the latter intervenes by considering the Nash equilibria. If the human being should be autonomous, for example when the criticality level and workload level are low and the level of experience is high, the agent does not intervene in view of the equilibrium result. In this case, if the human being does not intervene, there will be an overall slowdown of the crossroads. Thus, the evolution of the average speed curve (Fig. 5) and the average waiting time at lights (Fig. 6) illustrate the result of the work carried out by the software agent when necessary in terms of traffic flow.

Despite the decrease in the average traffic speed for example (in cases where the human being has a high level of expertise and a minimal workload), the assistant agent who does not intervene has the possibility to take special care of other complex tasks. So, in the sense of the payoff of the assistant, we can say that even if it does not intervene, its payoff can be translated by its availability to other tasks.

These results are promising, despite the relatively simple behavior of the software agent in view of the complexity of the application in which it is involved. Indeed, these results show the effectiveness of the intervention (or non-intervention) of the software agent while considering the information related to the context.

VI. CONCLUSION

This work has presented an interaction model, considering a software agent, based on game theory. The case study on road traffic supervision shows the interest and simplicity of implementing the software agent using the game matrix. The agent calculates the equilibrium at any time and decides the appropriate action based on Nash equilibria. The results thus show the effectiveness of the software agent in traffic management even in the absence of human intervention. The motivation for the use of game theory is to be able to consider an agent that can help the operator by taking care of other complex tasks at the same time. The agent must not waste unnecessary time in its interventions: this is the basic principle used for the proposal of utility functions. The game considered here is a repeated and cooperative two-player game between the human operator and the assistant agent. Our research perspectives focus on the generalization of the approach, by implementing several intelligent agents as assistants to several human beings for the supervision of road traffic. Another perspective will be to embed each assistant agent with learning capabilities, in relation to the history of interactions.

REFERENCES

- [1] J.A. Jacko, "Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications," CRC press, 2012.
- [2] E.A. Edmonds, "Adaptative Man-Computer Interfaces." In : M.J. Coombs and J.L. Alty (eds.), Computing skills and the user interface, London, Academic press, pp. 389-426, 1981.
- [3] R. Mandiau, C. Kolski, B. Chaib-Draa, and P. Millot, "A new approach for the cooperation between human(s) and assistance system(s): a system based on intentional states," Proceedings World Congress on Expert Systems, Orlando, Florida, Dec. 16-19, 1991.
- [4] C. Kolski, G. Boy, G. Mélançon, M. Ochs, and J. Vanderdonckt, "Cross-fertilisation between human-computer interaction and artificial intelligence," In P. Marquis, O. Papini, H. Prade (Eds.), A guided tour of artificial intelligence research, Vol. 3, Springer, pp. 1117-1141, 2020.
- [5] E. Grislin-Le Strugeon, E. Adam, C. Kolski, "Agents intelligents en interaction homme-machine dans les systèmes d'information," In C. Kolski C. (Ed.), Environnements évolués et évaluation de l'IHM. IHM pour les SI, vol. 2, Hermès, Paris, pp. 209-248, 2001.
- [6] J. J. Kponyo, Y. Kuang, and Z. Li, "Real time statut collection and dynamic vehicular traffic control using ant colony optimization," ICCP 2012 IEEE, 2012, pp. 69-72.
- [7] D.A. Roozemon, J.L.H. Rogier, "Agent controlled traffic lights," ESIT 2000, 14-15 September 2000, Aachen, Germany.
- [8] Y. Liu, L. Liu, and W.-P. Chen, "Intelligent Traffic Light Control Using Distributed Multi-agent Q Learning," IEEE 20th Int. Conf. on Intelligent Transportation Systems, Yokohama, 2017, pp. 1-8, 2017.
- [9] V. Hirankitti, J. Krohkaew, and C.J. Hogger, "A Multi-Agent Approach for Intelligent Traffic-Light Control," IAENG, Proceedings of the World Congress on Engineering 2007 Vol IWCE 2007, July 2 - 4, London, 2007.
- [10] R.J. Oskpoue, H.N. Varzeghani, and Z. Samadyar, "Intelligent agent: a comprehensive survey," International Journal of Electronics Communication and Computer Engineering. Volume 5, Issue 4, 2014 pp. 790-798.

- [11] J.P. Müller, "Architecture and application of intelligent agent: a survey," *The Knowledge Engineering Review*, Vol. 13:4, 1998, pp. 353-380.
- [12] D.A. Roozmond, "Using intelligent agents for urban traffic control systems," *Proceedings of International Conference in Artificial Intelligence in Transportation System and Science*, 1999, pp. 69-79.
- [13] A.C. Alvarado, E. Vazquez Santacruz, and M. Gamboa Zuniga, "Construction of a basic intelligent agent," *Intelligent Systems Conference (IntelliSys)*, 2017, London, pp. 341-345.
- [14] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. Tonguz "Intelligent traffic signal control using reinforcement learning with partial detection," arXiv:1807.01628v2, 2019.
- [15] B.C. Yeo, W.S. Lim and H.S. Lim, "Impact of vehicle detection accuracy to the performance of intelligent traffic control," *Australian Journal of Basic and Applied Sciences*, 10(8), 2016, pp. 119-126.
- [16] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espie, "A behavioral multi-agent model for road traffic simulation," *Eng. Applications of Artificial Intelligence*, 21, 2008, pp. 1443– 1454.
- [17] A. Champion, J.-M. Auberlet, R. Mandiau, S. Espié, and C. Kolski, "Simulation comportementale du trafic routier en intersection : Un mécanisme de résolution de conflit," *Recherche, Transports, Sécurité* 100, 2008, pp.185-194.
- [18] M. Camurri, M. Mamei, and F. Zambonelli, "Urban Traffic Control with Co-Fields," *Environments for Multi-Agent Systems III*, "LNAI 4389, Springer, 2007, pp. 239-253.
- [19] H.O. Al-Sakran, "Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology," *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 2, 2015, pp. 37-43.
- [20] Y. Lebrun, E. Adam, R. Mandiau, and C. Kolski, "A model for managing interactions between tangible and virtual agents on an RFID interactive tabletop: Case study in traffic simulation," *Journal of Computer and System Sciences* 81, 2015, pp. 585– 598.
- [21] R. Mandiau, A. Champion, J.-M. Auberlet, S. Espié, and C. Kolski, "Behaviour based on decision matrices for a coordination between agents in a urban traffic simulation," *Applied Intelligence*, 28, pp. 121–138, 2008.
- [22] S. Wang, Q. Zhao, N. Zhang, T. Lei and F. Yang, "Virtual Vehicle Coordination for Vehicles as Ambient Sensing Platforms," *IEEE Access*, vol. 6, pp. 11940-11952, 2018.
- [23] F. Perronnet, J. Buisson, A. Lombard, A. Abbas-Turki, M. Ahmane and A. El Moudni, "Deadlock Prevention of Self-Driving Vehicles in a Network of Intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4219-4233, Nov. 2019.
- [24] J. Del Ser, E. Osaba, J. J. Sanchez-Medina, I. Fister and I. Fister, "Bioinspired Computational Intelligence and Transportation Systems: A Long Road Ahead," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 466-495, Feb. 2020.
- [25] A.K. Dey, "Understanding and using context," *Personal and ubiquitous computing*, 5 (1), 2001, pp. 4-7.
- [26] S.D. Hall-Ellis, and D.S. Grealy, "The Dreyfus Model of Skill Acquisition: A Career Development Framework for Succession Planning and Management in Academic Libraries," *College & Research Libraries*, 74 (6), 2013, pp. 587-603.
- [27] B. Kirwan, A. Evans, L. Donohoe, A. Kilner, T. Lamoureux, T. Atkinson, and H. MacKendrick, "Human factor in ATM system design life cycle," *FAA/Eurocontrol ATM R&D Seminar*, 16-20 June, 1997, Paris, France, pp. 1-21.
- [28] M.N. Szilagyi, "Investigation of N-Person Games by Agent-Based Modeling," *Complex Systems*, 21 (3), 2012, pp. 201-243.
- [29] J. Nash, "Equilibrium points in n-person games", *Proceedings of the National Academy of Sciences*, 36(1), 1950, pp. 48-49.
- [30] H. Hamila, E. Grislin-Le Strugeon, R. Mandiau, A. Mouaddib, "Strategic dominance and dynamic programming for multi-agent planning, application to the multi-robot box-pushing problem," *ICAART 2012, 4th International Conference on Agents and Artificial Intelligence*, Vilamoura, Algarve, Portugal, 2012, pp. 91-97.
- [31] Y. Shoham, and K. Leyton-Brown, "Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations," Cambridge University Press, 2008.
- [32] R. DuFour, R. DuFour, R. Eaker, and T.W. Many; and Mike Mattos, "Learning by Doing: A Handbook for Professional Learning Communities at Work," Solution Tree Press, 2016.
- [33] C. Purucker, F. Rüger, N. Schneider, A. Neukum, and B. Färber, "Comparing the Perception of Critical Longitudinal Distances between Dynamic Driving Simulation, Test Track and Vehicle in the Loop," *Proc. AHFE 2014, Kraków, Poland 19-23 July 2014*, pp. 2089-2098.
- [34] U. Wilensky, and W. Stroup, "NetLogo HubNet Gridlock AlternateHubNet," Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 2002.
- [35] A. Doniec, R. Mandiau, S. Piechowiak, S. Espié, "Controlling no-normative behaviors by anticipation for autonomous agents," *Web Intelligence and Agent Systems*, 6 (1), 2008, pp. 29-42.